# A Computationally Efficient and Temporally Scalable Dynamic Traffic Simulation and Assignment System – Dynamic Urban Systems in Transportation (DynusT)

Yi-Chang Chiu, PhD., Eric Nava, MSCE
Department of Civil Engineering and Engineering Mechanics
University of Arizona, Tucson, AZ 85721
Tel: (520) 626-8462, Fax: (520) 626-2550, E-mail: chiu@email.arizona.edu

## 1. INTRODUCTION

The dynamic traffic simulation and assignment model DynusT (Dynamic Urban Systems in Transportation) is a model system that is designed and implemented to perform simulation-based dynamic traffic assignment (DTA) analysis. Due to its unique algorithmic structure and software implementation, it is capable of performing DTA on regional-level networks with long simulation period. This makes DynusT particularly well-suited for regional level modeling such as regional transportation planning, corridor studies, integration with activity-based models and mass evacuation modeling. The purpose of this paper is to present an overview of the theoretical and algorithmic innovation in DynusT.

As shown in Figure 1, DynusT consists of iterative interactions between its two main modules – traffic simulation and traffic assignment. Vehicles are created and loaded into the network based on their respective origins and follow a specific route based on their intended destinations. The large-scale simulation of network-wide traffic is accomplished through the mesoscopic simulation approach that omits inter-vehicle car-following details while maintaining realistic macroscopic traffic properties (i.e. speed, density and flow). More specifically, the traffic simulation is based on the Anisotropic Mesoscopic Simulation (AMS) model that simulates the movement of individual vehicles according to the concept that a vehicle's speed adjustment is influenced by the traffic conditions in front of the vehicle. In other words, at each simulation interval, a vehicle's speed is determined by the speed-density curve, and the density is

defined as the number of vehicles per mile per lane with a limited distance - defined as the speed-influencing region (SIR) – downstream of the vehicle (Chiu et al. 2010).

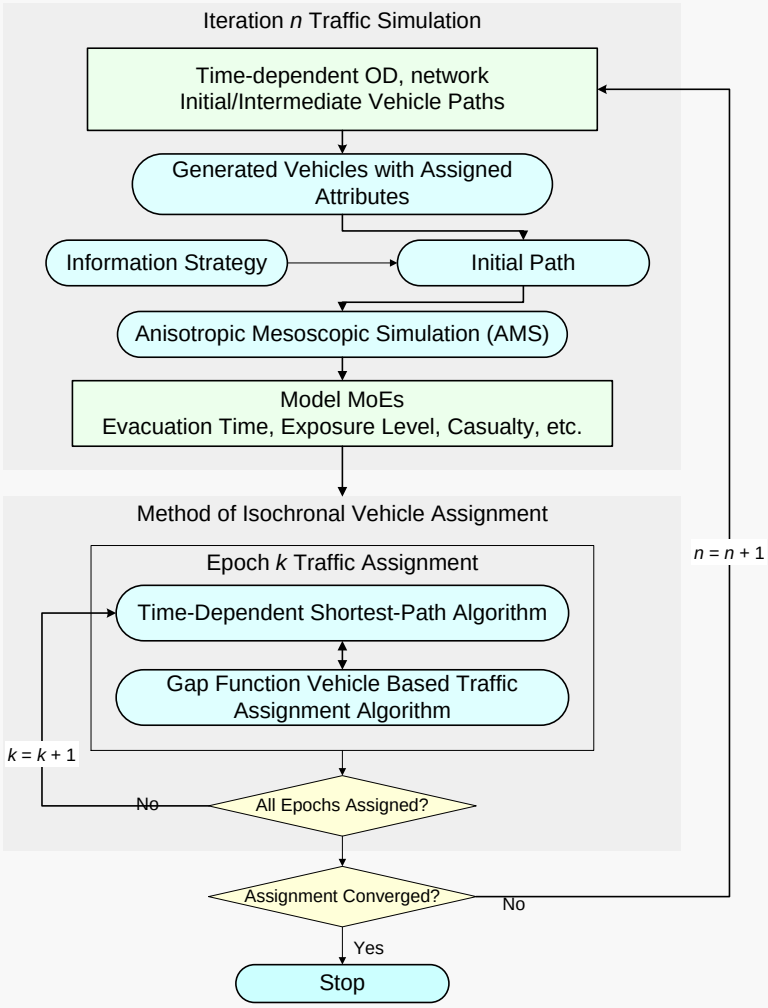



Figure 1: Traffic Simulation, Assignment and Link Volume Estimation Framework in DynusT

After simulation, necessary measures of effectiveness (MoEs) are fed into the traffic assignment module. The traffic assignment module consists of two algorithmic components: a time-dependent shortest-path (TDSP) algorithm and time-dependent traffic assignment. The TDSP algorithm determines the time-dependent shortest path for each departure time, while the traffic assignment component assigns a portion of the vehicles departing at the same time between the same OD pair to the time-dependent least-travel time path following the "route swapping" type of traffic assignment procedure.

In DynusT, the assignment algorithm maintains the balance of computational efficiency and solution algorithm quality. Innovations in computational efficiency allow DynusT to perform 24-hour assignment, which is critical for estimating daily traffic patterns for the purpose of this study. The computational features include: (1) Reuse vehicle ID to commit computer memory only for those vehicles that exist in the network during simulation, thus, memory usage is not cumulative to the total number of generated vehicles; (2) Assign vehicles with time-dependent shortest paths (TDSP) that are solved based on an *epoch*, which is the time period over which network statistics are collected for solving for the TDSP. An *epoch* was defined to be about 1-2 hours in length. This is to ensure that the memory usage for the TDSP is limited by the length of the *epoch* regardless of the length of the total evacuation simulation period.

Once the assignment of the current iteration is finished, all vehicles are loaded and moved along their paths in the simulation module again to evaluate if the TDUE condition is satisfied. If so, the algorithmic procedure is terminated; otherwise, the next iteration continues.

The rest of this paper is structured as follows. Section 2 presents the mesoscopic simulation model – anisotropic mesoscopic simulation (AMS) enables the traffic simulation component of DynusT. Section 3 discusses the gap function vehicle-based (GFV) traffic assignment algorithm that serves as the foundation for the Method of Isochronal Vehicle Assignment (MIVA) scheme as described in Section 4. Section 5 summarizes the model capability of DynusT in the areas of demand-supply interaction, information provision, pricing and mass evacuation. The supporting capabilities for DynusT such as OD calibration is discussed in Section 6.

## 2. MESOSCOPIC TRAFFIC SIMULATION - ANISOTROPIC MESOSCOPIC SIMULATION (AMS) MODEL

The AMS model is developed based on two intuitive concepts and traffic characteristics: (1) at any time, a vehicle's prevailing speed is influenced only by the vehicles in front of it, including those that are in the same or adjacent lanes; (2) the influence of traffic downstream upon a vehicle decreases with increased distance. These two characteristics define the "anisotropic" property of the traffic flow and provide the guiding principle for AMS model design. Based on such concepts, we define that for any vehicle *i*, only those leading vehicles s present in vehicle *i*'s immediate downstream and within a certain distance are considered to influence vehicle *i*'s

speed response. This is a similar concept to a stimulus-response type of car-following model, with the distinction that in AMS, the stimulus of a vehicle's speed response is represented in a macroscopic manner instead of using inter-vehicle distance or speed as in microscopic models.

For modeling purposes, the *Speed Influencing Region* for vehicle *i* ($SIR_i$) is defined as vehicle *i*'s immediate downstream roadway section in which the stimulus is significant enough to influence vehicle *i*'s speed response. This concept is further depicted in Figure 1, in which a multi-lane homogeneous roadway segment is considered. The *Speed Influencing Region* (*SIR*) for vehicle *i* is defined as the area (including the lane in which vehicles reside and all the adjacent lanes) in front of vehicle *i*, where the traffic condition (represented by the density) affects vehicle *i*'s speed response. At each simulation clock tick, vehicle *i*'s speed is influenced by the density in *SIR*. The upstream traffic and downstream traffic outside the *SIR* does not influence vehicle *i*. The $SIR_i$ length can be assumed to be either equal for all vehicles or variable according to different flow conditions. The $SIR_i$ length is assumed to be an average value *l* across all vehicles in this paper. The traffic density in $SIR_i$, denoted as $k_i$, is calculated as the number of vehicles present in $SIR_i$ divided by the total lane-miles of the $SIR_i$. As such, the unit of $k_i$ becomes the number of vehicles per mile per lane.

At the beginning of a simulation interval *t*, for each vehicle *i*, the prevailing speed of vehicle *i* during the simulation interval *t* is determined by Equation (1), where $\wp : k \to v$ is a non-increasing speed-density relationship function with two boundary conditions: $\wp(0) = v_f$ and $\wp\left(k_{queue}\right) = 0$. The queue density $k_{queue}$ is defined as the "bumper-to bumper" density observed in a long, standing-still queue, which is generally greater than the jam density reported in the literature.

The algorithmic steps of an AMS model during simulation are as follows: at each clock tick *t* (the beginning of a simulation interval), each vehicle's speed $v_i^t$ is evaluated based on its *SIR* density, which is obtained from the previous clock tick $k_i^{t-1}$ through the *v*-*k* relationship function $\wp\left(k_i^{t-1}\right)$. The *SIR* density is calculated based on Equations (2) or (3), depending on whether or not the *SIR* spans over the freeway segment with a different capacity. If the *SIR* spans a homogeneous highway section, Equation (2) applies; otherwise, Equation (3) is used. Vehicle

*i*'s traveling distance at the end of the current simulation interval is obtained by taking the prevailing speed $v_i^t$ times the duration of the simulation interval $\Delta$.

$$v_i^t = \wp\left(k_i^{t-1}\right) \tag{1}$$

$$k_i^{t-1} = \min\left[k_{queue}, \frac{N_i^{t-1}}{nl}\right] \tag{2}$$

$$k_i^{t-1} = \min\left[k_{queue}, \frac{N_i^{t-1}}{mx_i^{t-1} + n(l - x_i^{t-1})}\right] \tag{3}$$

where,

*i*: Subscript denoting a vehicle. The index *i* decreases with vehicles traveling in the same direction on the same link

*t*: Superscript denoting a simulation interval

*l*: *SIR* length

$v_i^t$: Prevailing speed of vehicle *i* during simulation interval *t*

$x_i^{t-1}$: Distance between Vehicle *i* and lane-drop (open) at clock tick *t-1*

$k_i^{t-1}$: Density of the *SIR* for vehicle *i*

$N_i^{t-1}$: Number of vehicles present in *SIR* , excluding vehicle *i*

$v_f$: Free-flow speed in the speed-density relationship

$\wp : k \to v$: Non-increasing speed-density function specifying the *v-k* relationship, where $\wp(0) = v_f$ and $\wp(k_{queue}) = 0$

$k_{queue}$: Queue density, $\wp(k_{queue}) = 0$

During the AMS simulation, each vehicle maintains its own prevailing speed and *SIR* at the beginning of a simulation interval. Individual vehicles' traveling distances are therefore likely to differ, even though they are on the same link. This feature is different from certain previous models (Jayakrishnan, Mahmassani and Hu, 1994; Balakrishna, Koutsopoulos and Ben-

Akiva, 2005), in which all moving vehicles on the same link travel at the same speed. This characterizes the AMS model as a vehicle-based mesoscopic model having a greater degree of resemblance with car-following-based microscopic models. The major difference between AMS and car following models is that in AMS, a vehicle's speed adjustment at each simulation time interval is governed by the *SIR* density $k_i^t$, which is a macroscopic measure of all the vehicles present in the *SIR* region, instead of an inter-vehicle measure between the target and the leading vehicle(s).

Since the *SIR* moves with each vehicle during simulation, it can be anticipated that in the AMS model, the vehicle advancing mechanism is generally independent of the representation of network structures (i.e. size/length of cell/segment/link) under the uninterrupted flow condition. Each vehicle makes speed adjustment decisions solely based on its *SIR* density; the AMS simulation results generally remain stable regardless of how link lengths are defined unless the link is shorter than a certain threshold that violates that required by a general time-based simulation.

AMS handles queue formation/discharge in a natural and straightforward manner. When $k_{queue}$ is reached, $v = \wp(k_{queue}) = 0$ ; vehicles speed up when the *SIR* density decreases. This mechanism allows for clear representations of substantial or transient queue formation or discharge. When a free-moving vehicle approaches the end of a queue, its speed gradually approaches the same speed of the queue tail as its *SIR* density approaches the *SIR* density of the leading vehicles. Depending on how the overtaking condition is met, this vehicle may trail at the end of the queue without "jumping over" leading vehicles, or it may stop ahead of the leading vehicle.

Equation (1) was further extended to simulate traffic flow in uninterrupted flow facilities under various configurations, such as homogeneous highways, non-homogeneous highways m and temporary blockage, by specifically considering different *SIR* density $k_i^t$ calculations corresponding to those conditions. As shown Equation (2), in the case of the homogeneous highway, $k_i^t$ is calculated as the number of vehicles presenting in the *SIR* divided by the total lane-miles of the *SIR* (i.e. the *SIR* length times the number of lanes). When lane drops or lane additions occur within the *SIR*, the total lane-mile of *SIR* is the sum of lane-miles of separate

sections, as shown in Equation (3). The lane drop (Figure 1(b-1)) or point bottleneck (Figure 1(b-2)) (from *m* to *n* lanes, $n < m$) occurs downstream from vehicle *i* The total lane-miles in the *SIR* is calculated as $mx+n(l-x)$, and the resulting $k_i^t$ is the smaller of $k_{queue}$ and $\frac{N_i^{t-1}}{mx+n(l-x)}$, which is the number of vehicles in the *SIR* at the beginning of the time interval *t-1* divided by the total lane-miles $mx+n(l-x)$ in the *SIR*.

In the case of a lane drop or a point bottleneck ($n < m$), the *SIR* density of a vehicle gradually increases (and hence speed reduces) as it approaches the bottleneck. When $n = 0$, a complete blockage occurs; this can be applied to either the point blockage or red-light signal indication. On the other hand, in the case of discharging from a bottleneck, as a vehicle approaching the open-up of the bottleneck, the density reduces and speed increases gradually.

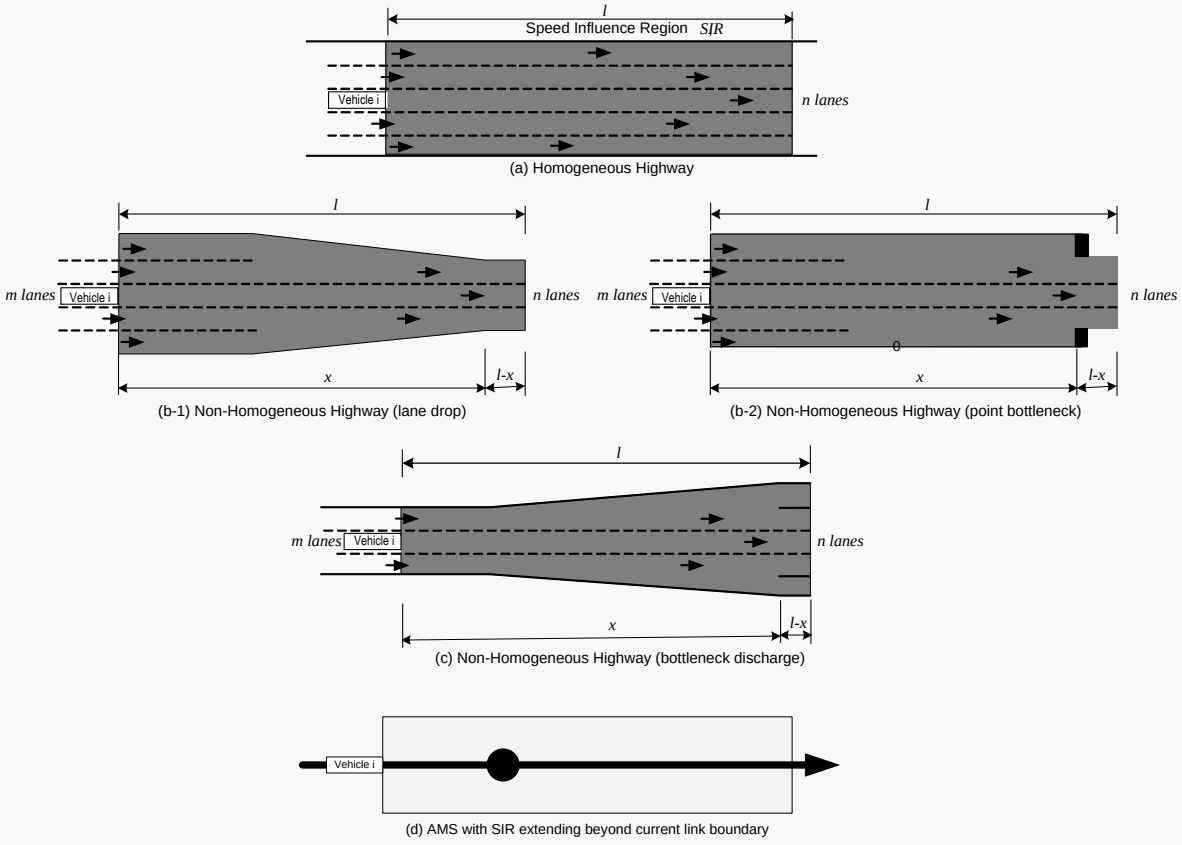



Figure 2 AMS model concept

## 3. GAP FUNCTION VEHICLE BASED TRAFFIC ASSIGNMENT

Contrasting the above reviewed methods, the proposed gap function vehicle-based (GFV) algorithm adopts the gradient projection concept, where path flow updates are comprised of both gradient and step size, and takes advantage of vehicle-based simulation allowing reassigned selected individual vehicles with better paths to improve their travel times. The gradient projection approach is common in constrained nonlinear programming and has been applied in many classical static and dynamic traffic assignment works (Dafermos and Sparrow 1969; Florian and Nguyen 1974; Sheffi 1985; Smith 1993). The "route-swapping" heuristics developed in more recent years could be found to be related to the same concept, except that in the route-swapping heuristics the step size is usually a pre-determined "swapping rate" and the direction is linearly proportional to the travel time difference to the shortest path travel time (Smith and Wisten 1995; Huang and Lam 2002; Szeto and Lo 2006). In a more recent study the swapping rate was proposed to be the ratio of the difference of the individual path travel time to the shortest path travel time (Lu 2007; Lu, Mahmassani et al. 2008).

In the proposed GFV method in the present study, the step size is in relation to the Relative Gap (RG) value calculated for all the paths between each origin-destination-departure time $(i, j, \tau)$ triplet at iteration *l*. Similar in concept to the prior studies, the GFV method leads to a smaller step size with a smaller RG value. The gradient determines the search direction, which means the paths to be updated with more flows or less vehicles. For each $(i, j, \tau)$ triplet, paths are sorted according to the average travel time, and vehicles traveling on each path are also sorted according to decreasing experienced travel time. Note that the assignment interval is normally much longer than the simulation interval. Therefore, vehicles departing within the same assignment interval, while subject to the same path set, would experience different experienced travel times due to their different depart times during the simulation interval.

Furthermore, at each iteration, once the assignment is completed, the path set $K^{l+1}(i, j, \tau)\ \forall$ all $(i, j, \tau)$ are released from memory. At the beginning of each iteration, the path set is re-constructed by scanning through each vehicle and assigned path. This may cause slight increase in computation, but it eliminates the need to retain the path set $K^{l+1}(i, j, \tau)\ \forall$ all $(i, j, \tau)$ when such information are not used (e.g. in simulation); thus, reducing the peak memory needed during the entire simulation-assignment procedure. It should also be noted that this strategy is

likely to deplete high-travel-time paths of vehicles. As a result, such paths will not appear in the next iteration path set.

Different from methods that explicitly solve for the optimal flow distribution among paths within $K^l(i,j,\tau)$ (Chang and Ziliaskopoulos 2007; Lu, Mahmassani et al. 2008), the determination of step size and gradient determination are based on simple joint approximation of descent direction and step size (proof not provided herein). This strategy is adopted after a careful consideration of the trade-offs between the solution quality and computational tractability. The methods that explicitly solve for the optimal distribution given $K^l(i,j,\tau)$ require significantly added computation time that may make the algorithm computationally intractable unless a parallel computing scheme is utilized. As shown in the numerical examples in the results section, the proposed GFV method exhibits satisfactory convergence quality with computational time.

A schematic representation of the proposed solution algorithm framework developed as part of Dynamic Urban Systems for Transportation (DynusT) illustrated in Figure 3. The iterative simulation assignment is initialized with the primary inputs of network loading: demand patterns, time-dependent origin-destination (OD) matrices and initial path assignments. As the Anisotropic Mesoscopic Simulation model (AMS) simulates vehicles within the network, evaluation of time-varying link densities, link flows, travel times and speeds are made (Chiu, 2008). After the initial network loading, the interplay between GFV and simulation continues until the convergence criterion is met. The convergence criterion is the gap function value, which is further discussed as follows.

The GFV procedure starts by sorting vehicles for all $K^l(i,j,\tau)$ based on vehicles' experienced travel time. Note that vehicles are loaded into the network on links; therefore, the origin node *i* for a vehicle refers to the downstream node of the link on which the vehicle is loaded.

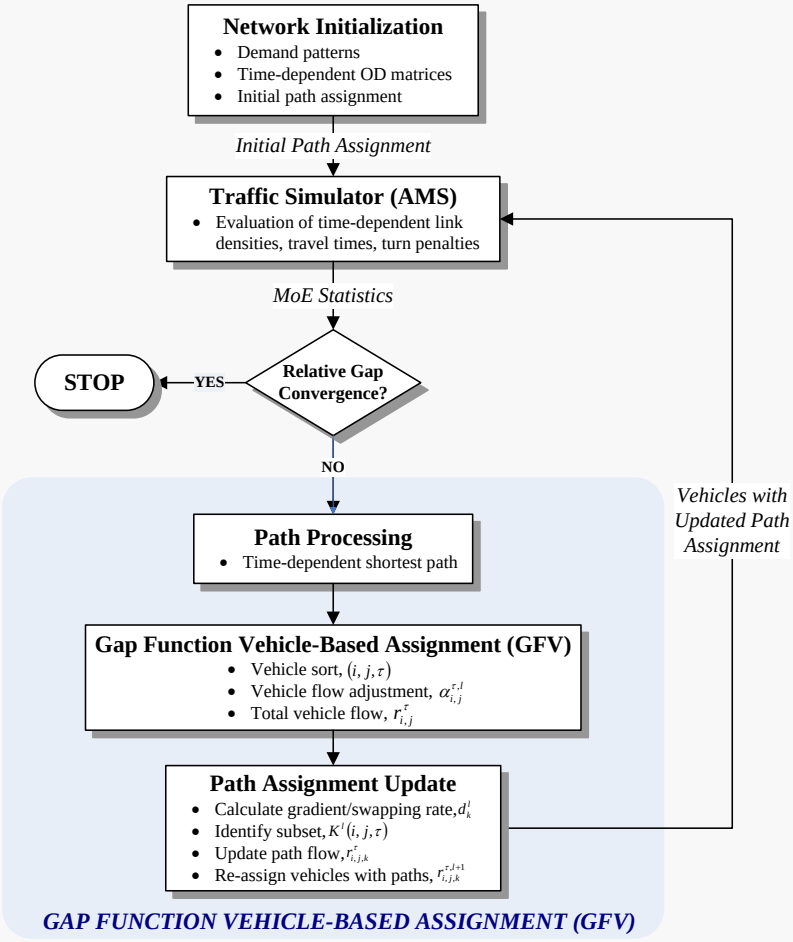



Figure 3 GFV method in conjunction with DynusT

In the GFV algorithm, the step size is in relation to the Relative Gap ($RG$) value calculated $\forall k \in K^l(i,j,\tau)$. Further, the relative gap $RG_k$ for path $k$ defined in Equation 3.1 indicates that $q_v$ is the experienced travel time of vehicle $v$ (from the downstream node of origin link $i$ to destination node $j$) and $u_{i,j}^{\tau,l}$ is the calculated time-dependent shortest path travel time for $(i,j,\tau)$ at iteration $l$ solved by the TDSP algorithm. $RG_k$ measures the travel time deviation of path $k$ to the shortest path.

$$RG_k = \frac{\sum_{v \in V^l(i,j,\tau,k)} q_v - r_{i,j,k}^{\tau,l} \cdot u_{i,j}^{\tau,l}}{r_{i,j,k}^{\tau,l} \cdot u_{i,j}^{\tau,l}}, \forall k \in K^l(i,j,\tau) \quad 3.1$$

The stopping criterion follows Equation 3.2.

$$\overline{RG} \leq RG^0 \quad 3.2$$

where $RG^0$ is the user-specified threshold and $\overline{RG}$ follows the Equation 3.3.

$$\overline{RG} = \frac{\sum_{i,j,\tau,k} \sum_{v \in V^l(i,j,\tau,k)} q_v - \sum_{i,j,\tau} \left( r_{i,j}^{\tau} \cdot u_{i,j}^{\tau,l} \right)}{\sum_{i,j,\tau} \left( r_{i,j}^{\tau} \cdot u_{i,j}^{\tau,l} \right)} \quad 3.3$$

Next, the time-dependent shortest path is solved. At each iteration $l$, the flows for each $k \in K^l(i,j,\tau)$ to be shifted at this iteration is the step size $\alpha_{i,j}^{\tau,l}$ times the total flow $r_{i,j}^{\tau}$ between the $(i,j,\tau)$ triplet. $\alpha_{i,j}^{\tau,l}$ is defined as the minimal of two candidate step sizes as shown in Equation 3.4. $\alpha'$ is the $RG$-based step size, calculated based on Equation 3.5; $\alpha^0$ is the maximum step size. The step size is determined by:

$$\alpha_{i,j}^{\tau,l} = \min\{\alpha^0, \alpha'\} \quad 3.4$$

as the $RG$-based step size is calculated as average RG value for all path $k \in K^l(i,j,\tau)$:

$$\alpha' = \left\{ \frac{\sum_k RG_k}{|K^l(i,j,\tau)|} \right\} \quad 3.5$$

$|K^l(i,j,\tau)|$ is the cardinality of the set of non-zero flow paths between criterion $(i,j,\tau)$ for iteration $l$. Paths in $K^l(i,j,\tau)$ are ordered with decreasing travel time. Note $K^l(i,j,\tau)$ includes the TDSP solved at the current iteration.

Based on Equations 3.1 through 3.5, one should expect that as the algorithm starts from an all-or-nothing (AON) assignment, the $|K^l(i,j,\tau)|$ is small and $RG_k$ may be large; therefore, the step's size will initially be capped at $\alpha^0$. As iterations increase, $|K^l(i,j,\tau)|$ also increases while $RG_k$ decreases. As mentioned before, the step size will first be capped by $\alpha^0$ at initial iterations, and as iterations increase the step size will be then handled by $\alpha'$ as the size of path sets $K^l(i,j,\tau)$ would eventually stabilize. Although not proven analytically, numerical experiments shown later in this section show that $\alpha'$ generally continues to decrease as the Dynamic User Equilibrium (DUE) solution is iteratively improved.

If the assignment starts from an initial simulation in which vehicles are being loaded following DUE paths solved from a prior DUE run, the step size in the initial iteration is likely to be capped by $\alpha'$. This reduces excess fluctuation of the initial assignment away from the DUE condition since the initial solution is already close to the DUE condition. Only those $K^l(i,j,\tau)$

exhibiting large gaps are to be shifted with a larger flow amount. This observation implies a relatively stable and quick convergence for a "warm-start" assignment. Empirical evidences will be provided in later sections.

Next, descent direction $d_k^l$ updates the direction of flow adjustment at iteration $l$ using what are considered increasing-flow and decreasing-flow path subsets. $K^l(i,j,\tau)^+$ defines the path subset that will have the increased flow after assignment while $K^l(i,j,\tau)^-$ is the path subset with decreased flow, where:

$$K^l(i,j,\tau)^+ \cup K^l(i,j,\tau)^- = K^l(i,j,\tau) \quad 3.6$$

After determining the step size, the amount of the total shifted flow is $\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}$, by definition. It is important to note that in $K^l(i,j,\tau)$, paths are always ranked in the order of increasing path average travel time $\bar{q}_{i,j,k}^{\tau,l}$, which is:

$$\bar{q}_{i,j,k}^{\tau,l} = \frac{\sum_{v \in V^l(i,j,\tau,k)} q_v}{|V^l(i,j,\tau,k)|}, \forall k \in K^l(i,j,\tau) \quad 3.7$$

and

$$\bar{q}_{i,j,1}^{\tau,l} \leq \bar{q}_{i,j,2}^{\tau,l} \leq \cdots \leq \bar{q}_{i,j,|K^l(i,j,\tau)|}^{\tau,l} \quad 3.8$$

The decreasing-flow path set $K^l(i,j,\tau)^-$, as defined in Equation 3.9 is determined by scanning paths $k = 1, 2, \ldots, \hat{k}$ until the condition defined in Equation 3.9 is met. $\hat{k}$ is the cutoff path in which vehicles in any path $k \in K^l(i,j,\tau)$ whose $\bar{q}_{i,j,k}^{\tau,l} \geq \bar{q}_{i,j,\hat{k}}^{\tau,l}$ will be considered for reassignment.

Subsequently, the decreasing-flow path set is defined as:

$$K^l(i,j,\tau)^- = \left\{ k = \hat{k}, \ldots, |K^l(i,j,\tau)| \middle| \sum_{k=1}^{\hat{k}-1} r_{i,j,k}^{\tau,l} < \alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \leq \sum_{k=1}^{\hat{k}} r_{i,j,k}^{\tau,l} \right\} \quad 3.9$$

Equation 3.9 only determines the decreasing-flow path set; however, being that the GFV algorithm is vehicle-based, the step size $\alpha_{i,j}^{\tau,l}$ specifies a certain number of vehicles would be reassigned to the increasing-flow path set. For that reason, all vehicles on path $k \in K^l(i,j,\tau)^- \backslash \hat{k}$ will be assigned to the increasing-flow path sets, but only part of those on the cut-off path $\hat{k}$

will be assigned. $\hat{v}$ is defined as the cutoff vehicle on path $\hat{k}$ such that any vehicle beneath $\hat{v}$ is not considered for reassignment.

$$\hat{v} = \sum_{k=1}^{|K^l(i,j,\tau)^-|} r_{i,j,k}^{\tau,l} - \alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \tag{3.10}$$

Those vehicles belonging to $k \in K^l(i,j,\tau)^-$ will be reassigned with one of the paths in the set $K^l(i,j,\tau)^+$, which may also include the latest solved TDSP. The re-distribution scheme as depicted in Equation 3.11

$$d_k^l = \begin{cases} \dfrac{(RG_{\hat{k}} - RG_k)^\theta}{\sum_{k \in K(i,j,\tau)^+} (RG_{\hat{k}} - RG_k)^\theta} & ,\forall\, k \in K^l(i,j,\tau)^+ \\ -\dfrac{r_{i,j,k}^{\tau,l}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}} & ,\forall\, k \in K^l(i,j,\tau)^- \backslash \hat{k} \\ -\dfrac{\hat{v}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}} & ,\hat{k} \in K^l(i,j,\tau)^- \end{cases} \tag{3.11}$$

where $RG_{\hat{k}}$ is the relative gap value for cutoff path $\hat{k}$.

For those paths $k \in K^l(i,j,\tau)^+$, the number of vehicles is increased by the amount of:

$$\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot \frac{(RG_{\hat{k}} - RG_k)^\theta}{\sum_{k \in K(i,j,\tau)^+} (RG_{\hat{k}} - RG_k)^\theta} \tag{3.12}$$

For a vehicle to be assigned with a path $k \in K^l(i,j,\tau)^+$, the probability can be expressed as

$$P(k) = \frac{(RG_{\hat{k}} - RG_k)^\theta}{\sum_{k \in K(i,j,\tau)^+} (RG_{\hat{k}} - RG_k)^\theta} \tag{3.13}$$

For those paths $k \in K^l(i,j,\tau)^- \backslash \hat{k}$ the change in the number of vehicles is:

$$-\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot \frac{r_{i,j,k}^{\tau,l}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}} \tag{3.14}$$

For $k = \hat{k}$, the decrease amount is:

$$-\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot \frac{\hat{v}}{\alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau}} \quad 3.15$$

Note that $\theta$ is a scaling factor, which can be kept as a pre-defined constant or can be an increasing function of iteration number. The larger $\theta$ is, the more aggressive the assigned flow to the first best paths will be.

Lastly, all flow determined to be reassigned will be applied to all paths $k \in K^l(i,j,\tau)^+$ following a nonlinear proportional scheme:

$$r_{i,j,k}^{\tau,l+1} = r_{i,j,k}^{\tau,l} + \alpha_{i,j}^{\tau,l} \cdot r_{i,j}^{\tau} \cdot d_k^l \ , \forall k \in K^l(i,j,\tau) \quad 3.16$$

### 3.1. GFV Computational Performance

To demonstrate consistency and stability of the GFV algorithm, 3 scenarios were executed and compared against the MSA algorithm:

**Case 1: Cold Start (CS) –** Start from non-DUE initial conditions

**Case 2: Warm Start (WS)–** Start from DUE as the initial condition

**Case 3: Work Zone with Warm Start (WWS) –** Start from DUE as the initial condition with change in network capacity

The CS case starts with the OD demand trip table with vehicles loaded at different departure times being assigned with a "instantaneous" shortest path that is calculated based on a "snapshot" of current network link travel times and node penalties at time of path calculation.. In proceeding iterations, the time-dependent shortest path algorithm along with the GFV algorithm will continue to improve the relative gap toward the DUE condition.

The initial iteration of the WS case begins from the simulation in which the same vehicle population is loaded with the DUE path obtained from the CS scenario. This is equivalent to taking the final vehicle and path roster at convergence to start the initial simulation. This would produce an initial simulation results similar to the DUE condition of the CS case[1]. The purpose

[1] The simulation will not be identical to the prior DUE simulation due to randomness introduced into simulation.

of this test case is to examine whether the path assignment algorithm is robust enough recognize how close the initial simulation is to the DUE condition

The WWS case is similar to WS as it begins from DUE vehicle and path roster from the CS case, but a work zone is introduced into the network to bring a disturbance to the DUE condition. This case is aimed at demonstrating the consistency and stability of the algorithm in that vehicles affected by the work zone should be adjusting their path toward re-equilibration and the "non-affected" vehicles would retain their original DUE path with minimal changes.

The Fort Worth network, as shown in Figure 4, represents a portion of the Fort Worth I-35W corridor traveling from north to south. The network contains 13 zones with 180 nodes and 445 links as the departure time interval was set at 2 minutes. The network contains traffic control signals along arterial link intersections which introduce both intersection delays and congestion. A total of 70,921 vehicles were generated as the simulation ran for 300 minutes at 100 iterations.

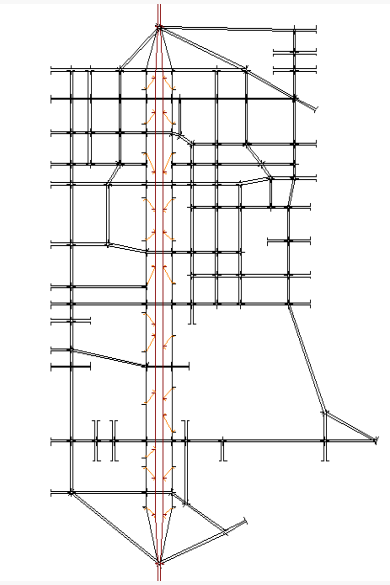


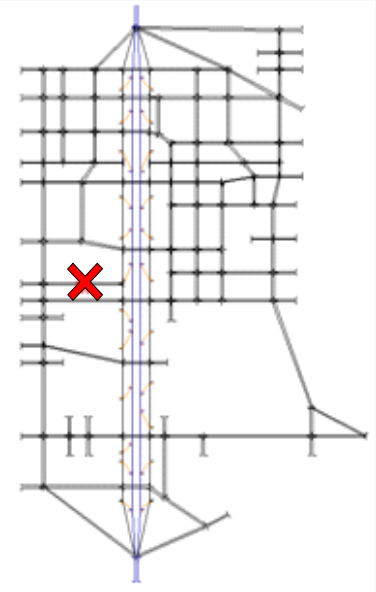
Figure 4 Fort Worth Network

The relative gap (RG), which was defined in Equation 3.3, and average network travel time provide the metric of convergence. The reported RG values include the maximum and final RG value for each test network. For the Fort Worth network case, the initial starting travel time were approximately similar for GFV and MSA, yet the GFV RG value slides consistently toward a stable level near 6%, whereas MSA is observed with a large GP value spike at initial iterations.

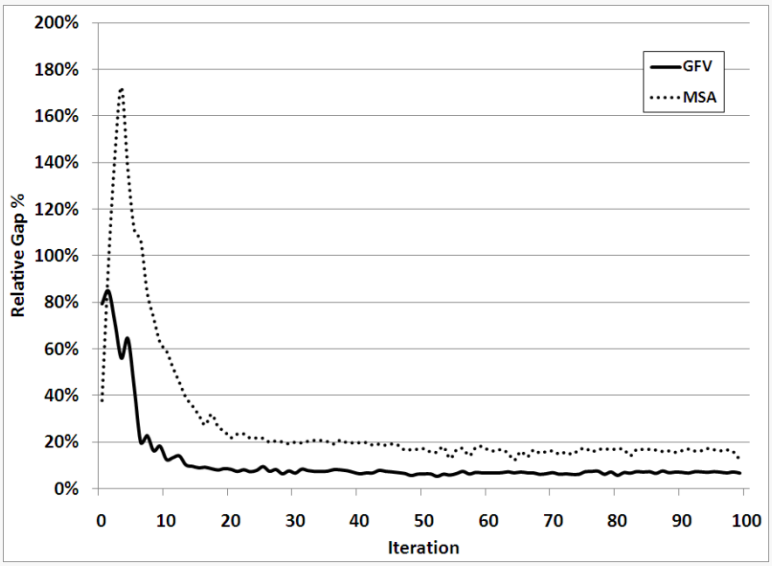
This is anticipated as MSA's fixed step size is likely to "over-correct" the assignment in the initial iterations. This is consistent with several prior studies.

The WS case is aimed at demonstrating the algorithm's design ability to recognize the initial solution quality, thus providing quick convergence back to a consistent and stable solution that was previously found in CS case. As can be seen in Figures 1.5c and 1.5d, a small RG value jump is observed for GFV and after 15 iterations, the RG values returns to the same level as in the CS case.

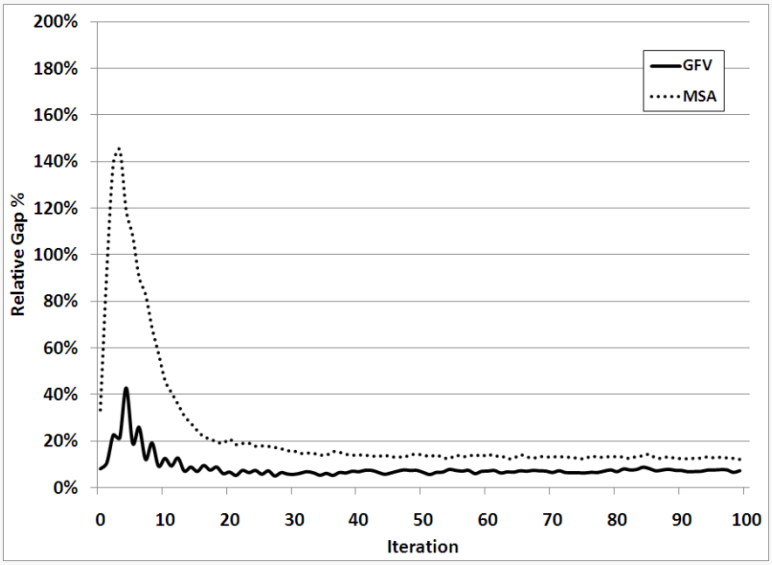
Even in WWS case where the work zone is introduced, The majority of traffic is traveling along the freeway and the work zone is located on the west side of the network (see Figure 5b), therefore, the path assignments along this freeway corridor should not be much affected, and total network convergence was not greatly affected and converged quickly. The large jump in both RG value and average network travel time value in MSA's initial iterations for all 3 cases demonstrate the instability of the fixed step size and descent direction. In particular, for cases WS and WWS the initial solution path distributions are blindly redistributed, thus taking more iteration to correct the flow distribution back to the DUE state.



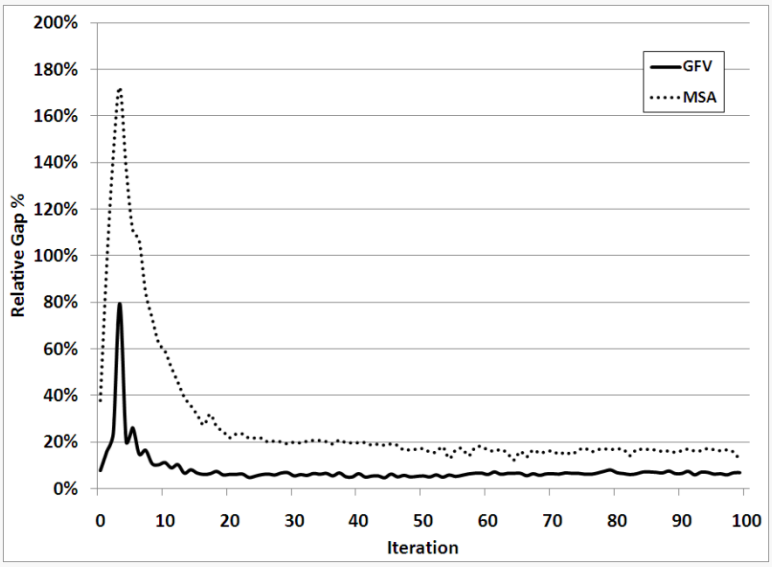
(a) Fort Worth Location of Workzone




(b) CS Case RG Value




(c) WS Case RG Value




(d) WWS Case RG Value

Figure 5 Fort Worth Network Convergence Results

## 4. METHOD OF ISOCHRONAL VEHICLE ASSIGNMENT

The MIVA approach is one of two critical developments enabling the daily simulation and assignment. The technical details about MIVA discussed hereafter, however, is limited to only the rationale and schematic design of the approach in the interest of paper length. More details can be found at (Chiu and Nava 2010). The proposed MIVA is aimed at decoupling the simulation (analysis) time period from those defined for TDSP and assignment algorithms. This technique reduces the memory usage and computational time for the TDSP and assignment to a much smaller level without degrading the assignment solution quality regardless of the length of the simulation (analysis) time period. The proposed MIVA scheme, as shown in Figure 6, consolidates the different time periods within a simulation-based DTA procedure to facilitate the following discussion. The time periods include the iterative execution of simulation, TDSP algorithm, and assignment procedures. Specifically, the simulation procedure evaluates the current assignment solutions and generates averaged time-varying link travel time/cost or intersection delay data necessary for the subsequent TDSP and assignment.

The simulation is a time-discretization process in which the network traffic states are updated at each *Simulation Interval*. During the simulation, several simulation intervals are

consolidated into an aggregation interval, over which the link travel times and intersection turning delays are averaged. This interval is the time unit used by the TDSP and the assignment procedure. All the time-dependent link travel times/costs or intersection delays/penalties are stored in a single data structure indexed by the aggregation intervals. Most importantly, the entire simulation (analysis) period further consolidated into several isochronal *Epochs*. Each Epoch is also associated with a *Projection Period* as shown in Figure 6. An Epoch is defined as the time period within which the generated vehicles are assigned with the path set solved by the TDSP algorithm of which the temporal domain is defined by the length of the Projection Period. The start time of each Projection Period is synchronized with the start time of its associated Epoch, but the end time is defined as the arrival time of the last arriving vehicle for those vehicles departing within the associated Epoch.

The entire assignment procedure is comprised of the sequential[2] execution the traffic assignment for each Epoch. At each Epoch, vehicles generated in this Epoch are first pre-processed and sorted. This step pertains to the algorithmic implementation of the Gap-Function Vehicle (GFV) assignment procedure utilized in this research (Chiu and Bustillos 2009). Next, the memory for the TDSP algorithm is allocated and initialized with link and node travel time/cost/delay values over the length of the Projection Period, followed by executing the TDSP algorithm. In the subsequent assignment step, underperforming vehicles are assigned with updated paths according to the GFV algorithm. Once the operation for the current Epoch is finished, the computer memory for TDSP and assignment algorithms are de-allocated. This step repeats until all the Epochs are processed. After finishing the last Epoch assignment, all the vehicles are fed into the simulation module to be simulated again. The whole iterative process continues until the convergence criterion is met or until the max number of iteration is reached.

[2] In fact, this step can be easily implemented as a multi-threaded procedure for faster computation.

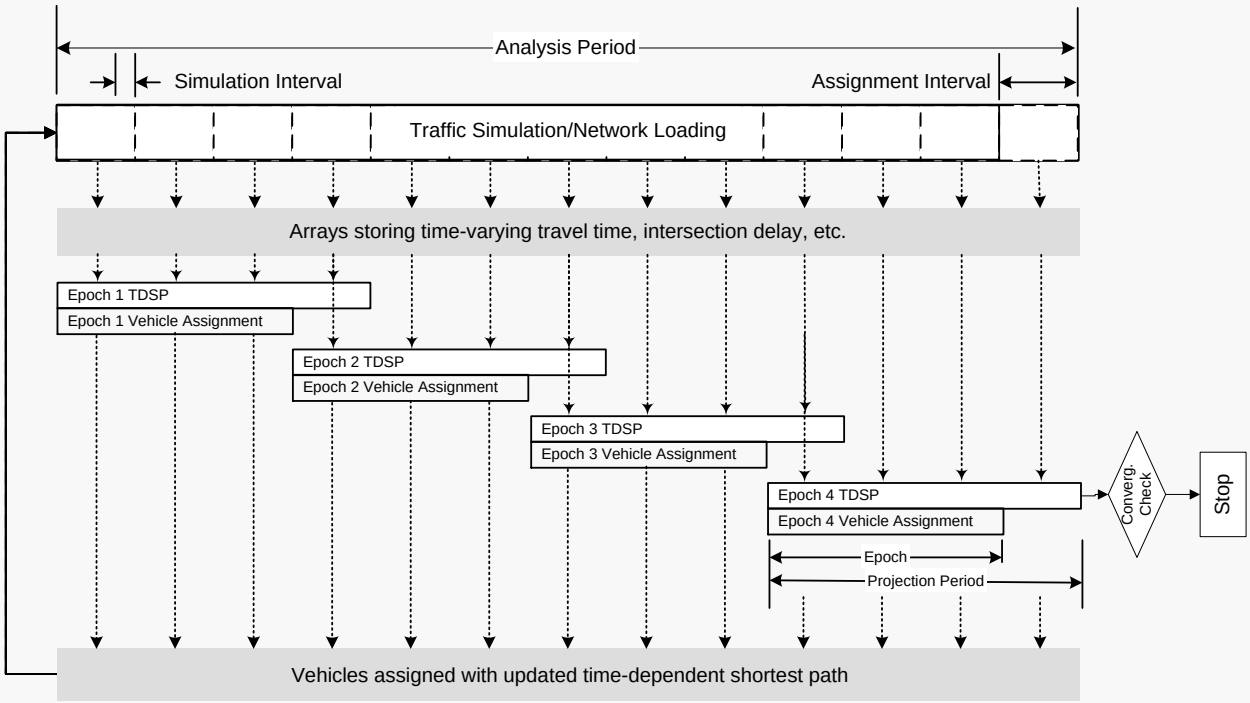



Figure 6 Algorithmic scheme of MIVA

In MIVA, a certain amount of computation overhead is associated with each Epoch. Such an overhead are required per GFV or MIVA computation. Memory allocation and de-allocation for TDSP and vehicle and path set arrays are typical fixed-amount overhead occurring within each epoch regardless of the length. As the number of epochs increases, the total time incurred by these overhead would linearly increase. Although a shorter Epoch will also lead to faster TDSP computation (note that the complexity of TDSP is $O(n^3T)$), there comes a point where the overhead outweighs the TDSP time savings as the number of Epochs increase.

The relation of overhead and TDSP cannot be analytically derived as doing so requires knowing the exact computation time for each algorithmic step operations; but these are highly hardware and operating system dependent (e.g. 32-bit versus 64-bit). The actual TDSP algorithm computation time is also affected by the variability of link and node performance (e.g. computing based on widely variable time-dependent link performance will take longer than on free-flow conditions). Consequently, the time-optimal number of epoch may not be predicted *a priori*. A model user would not know the optimal setting unless he/she tries several different setting, but this is obviously undesirable.

A potential strategy is to enable an on-the-fly self-tuning mechanism that is able to automatically test and determines the time-optimal epoch setting during the solution iterations. The goal of designing an on-line self-tuning (ST) MIVA mechanism is to introduce an adaptive and robust capability that can gradually search for a time-optimal epoch setting from a set of permissible number of epochs. This permissible epoch set is determined primarily based on the memory resource limitation. A user conducts an initial "bootstrapping" test run starting with few numbers of epochs to decide the minimal number of epochs that the memory can handle (e.g. without paging the virtual memory, namely, the hard drive) as this sets the lower bound for the permissible epoch set. The upper bound can be determined at the model user's discretion, but all permissible number need conform a few basic algebraic relations, that is the analysis period needs to be the multiples of epoch length and both the analysis period and the epoch length needs to be the multiples of the aggregation/assignment intervals. This ensures consistent time relations among these intervals as MIVA operates on time-discretized manner.

An online self-tuning (ST) mechanism is wielded into the MIVA computational scheme in which the ST operation starts from a set of permissible Epoch values. Over the iterations, the ST mechanism will eliminate undesirable values and eventually retain a desirable epoch setting. Essentially, the ST-MIVA algorithm evaluates an Epoch value setting for one iteration. The computation time is recorded for the duration of that iteration's execution of the assignment module. Through iterations, different Epoch value settings will have been evaluated, thereby choosing the Epoch value setting with the least execution time will be used during the remaining iterations of the SBDTA algorithm. Performing a "line search" is certainly efficient. The means by which Epoch value settings will be evaluated and chosen is a "bisection" search method. Keep in mind, any Epoch value setting will save computational memory usage, so choosing a reasonable setting rather than spending resources searching for an optimal value may be more suitable as the marginal memory savings may not be worth the searching price.

### 4.1. MIVA Computational Performance

The following network was used to test the capability of executing the SBDTA algorithm for a 24 hour simulation. The test network was anticipated to have a long computational time, so by using the ST-MIVA algorithm, it would save much time to not have to determine an optimal Epoch setting value through multiple tests. The DRCOG (Denver Regional Council of

Governments) network consists of 2,832 zones, 10,095 links and 23,147 links. The aggregation interval is set to be 15 minutes. Approximately 6.8 million vehicles were generation within the 24-hour analysis period and a total of 10 iterations are specified.

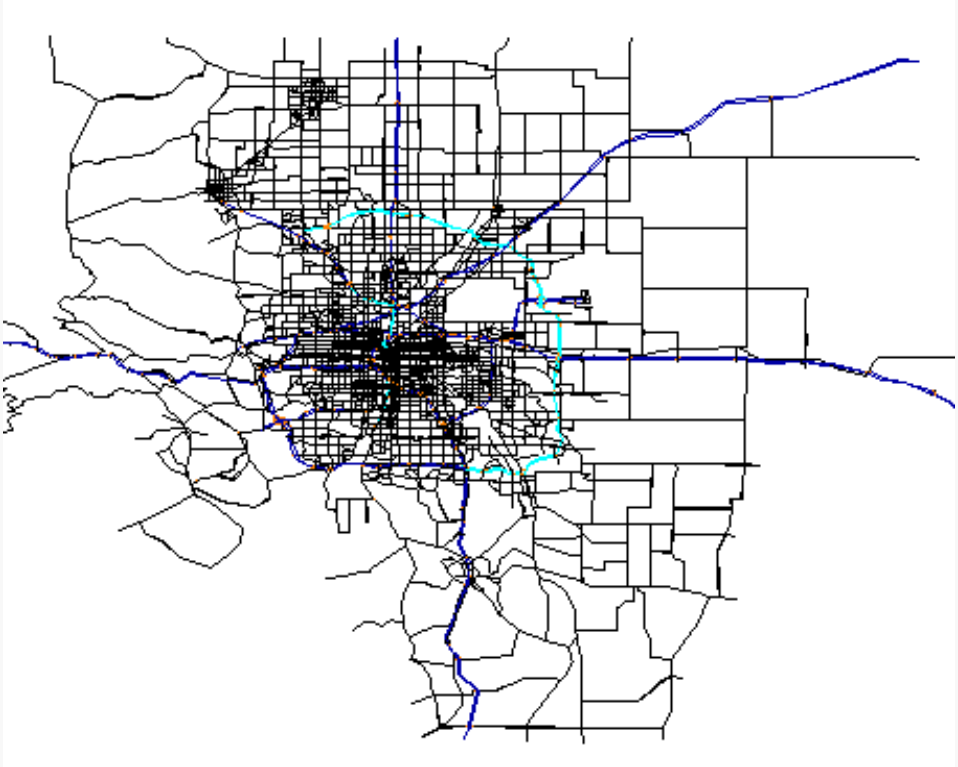


Figure 7: DRCOG Network

The DRCOG network ran on a machine with four 2.4 GHz Quad-Core AMD Opteron 2378 processors with 32 GB of RAM.

The DRCOG network using the MIVA scheme demonstrated a memory usage reduction of 79.2% when comparing with the 5.4 GB usage for the ST-MIVA and 25.9 GB for the full-scale case. The memory saving results re-affirms the superior performance of ST-MIVA from both computation time and memory standpoints.

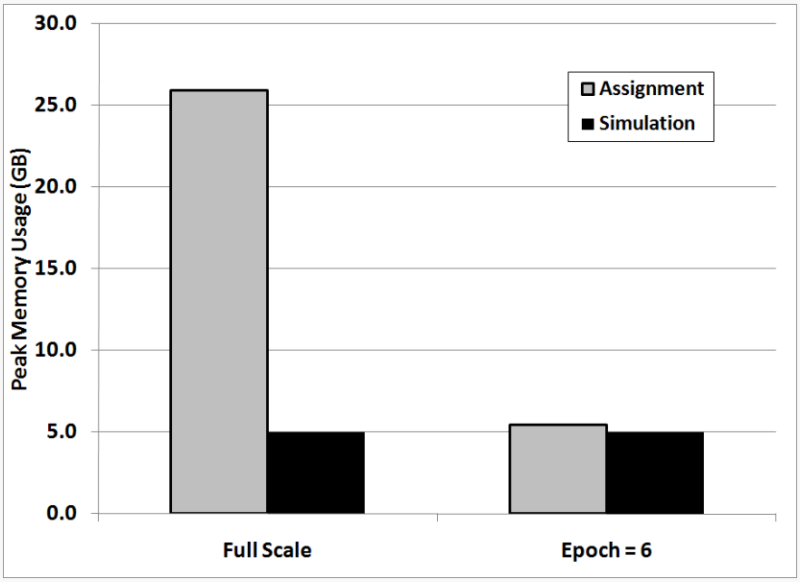



Figure 8 Peak Memory Usage for El Paso and DRCOG comparing the ST-MIVA Algorithm and Full-Scale Case

There are a total of six permissible epoch settings for the DRCOG network. For the first 2 extreme epoch settings, four epochs appear to outperform 24 epochs. Eight epochs are selected in the 3rd iteration and finally the 4-epoch setting is found to be the optimal. The search pattern clearly demonstrates the robustness and quick convergence to a desirable Epoch setting value. The marginal savings in execution time from the first iteration to the final convergence iteration is significantly demonstrated by saving approximately 1 hour. Comparing to the worse setting, then the execution time savings is about 3.8 hours, representing a 27% improvement. The final Epoch setting value was 8 (180 minute Epoch lengths).

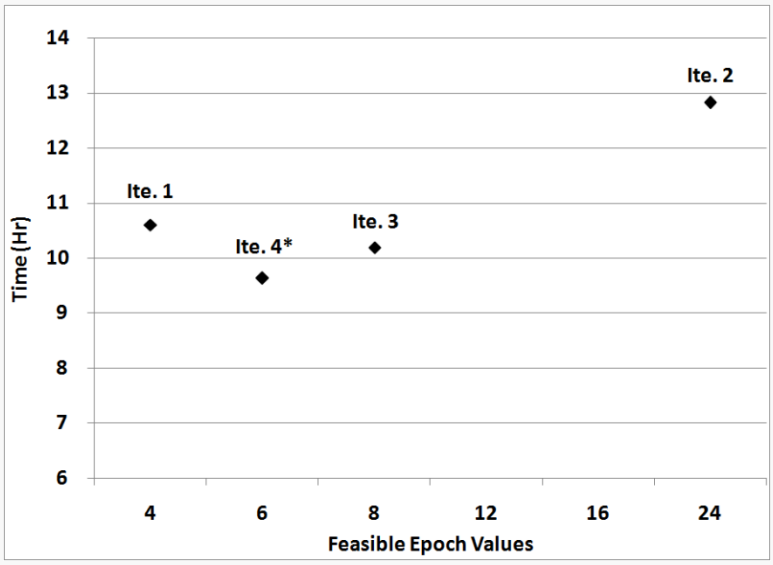



Figure 9 Self-Tuning Algorithm Result of Final Epoch Value Setting

This testing provides a critical piece of evidence that ST-MIVA will lead to an epoch setting that outperforms the full-scale scenario from both memory and computation time standpoints. As previously concluded, memory usage always reduce with increased number of epochs. Six epochs for the DRCOG network also outperform the full-scale case in both memory and computation time.

Also evident from the DRCOG network is the execution time savings when utilizing the ST-MIVA technique. Keep in mind of the grandness of the DRCOG network and running the network for a full 24 hour assignment. The total amount of execution time for the ST-MIVA algorithm was 118.91 hours. However, compared to the full scale assignment time of 175.29 hours, a 32.1% computation time improvement is obtained.

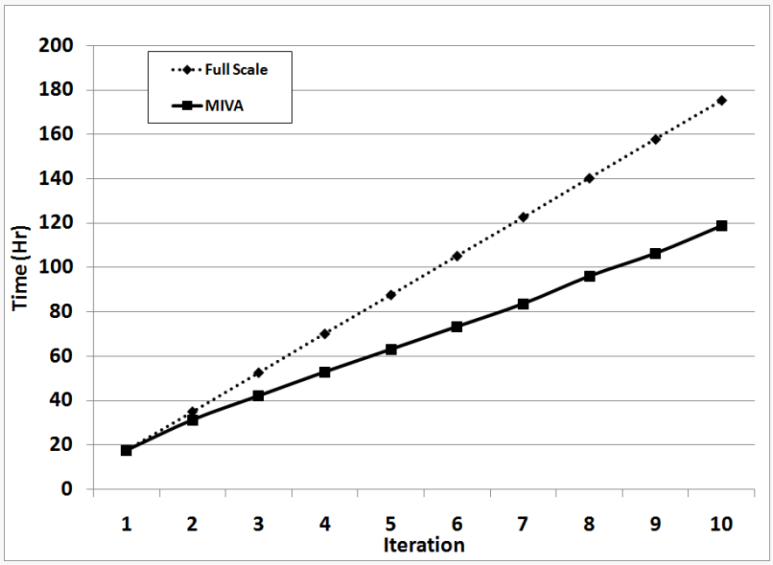



Figure 10 Execution Time Pattern Savings with ST-MIVA Algorithm

Figure 11

# 5. MODELING CAPABILITIES

## 5.1. General Information Provision

### *5.1.1. Pre-trip information*

Pre-trip best path information in modeling is equivalent to the driver knowing in advance that there is road work or a closure before leaving, and so avoids the congestion by choosing an alternate route and/or departure time. DynusT simulates this scenario by assigning a pre-trip vehicle the quickest path at the time that it is generated.

### *5.1.2. Enroute information*

Two types of information are considered for this class of vehicles:

1. Radio type of information in which the incident or disaster location is presented to drivers at the pre-defined frequency. One can define a fraction of drivers to receive such information. Upon receiving the information, the driver will select a route to their destination based on their prior knowledge about the network condition as well as their speculation of the possible congestion around the incident area.

2. GPS navigation devices that presents new route based on updated travel time retrieved from the base station. The driver decides on whether the new route is chosen based on the the boundedly rational behavior. The switching criteria are “Indifference Band” and “Threshold Bound”. A driver considers switching routes whenever the en-route travel information is updated at each predefined interval.

Also note that in the event of disasters, in which all the roadways connecting to the a driver's destination are blocked, both the above en-route information will trigger the driver's decision in that (1) if the driver leaves the original prior to the occurrence of the event, he will return home as soon as he/she receive the closure information; (2) if a driver departs after the disaster occurrence, the trip will be canceled. Note that this rule applies to only non-evacuees. Those evacuate from the hot-zone will continue to their intended safe locations.

### 5.2. Pricing

To be completed.

### 5.3. Mass Evacuation

The advantage of using DynusT traffic simulation model is the ability to capture complex and non-linear dynamical interactions between various entities during an evacuation event. These entities can be generally characterized as “demand” and “supply” constituting an evacuation modeling context. The total number of evacuees, the intended evacuation destinations, departure pattern, and the chosen routes determines “desires” to request the transportation service. On the “supply” side the network topology, capacity, configurations and traffic controls limits the amount of “available service” can be used by the evacuees.

It is quite usual that during an evacuation event, “demand” is much greater than “supply” over an extended time period, resulting in observed server and prolonged traffic congestion. The severity and time extent of such congestion cannot be estimated by simple calculation such as “total flow divided by capacity” due to a widely known traffic flow phenomenon in which the traffic throughput is much less than the nominal capacity under congestion. More so, evacuees change their evacuation decisions prior to or during evacuation in response to various traffic management strategies, such as evacuation information or contra-flow lanes, etc. Once the

decisions are modified the traffic "demand" on various evacuation routes would change, so does the congestion resulted from the new demand-supply interaction.

The system-level outcome of such an interaction, as depicted in Figure 12 can be properly captured through traffic simulation model equipped with pertinent simulation modeling components. The interaction of the evacution demand and supply interaction is represented in terms of a set of outcome or measure of effectiveness (MoE) by which an emergency management agency can derive further management strategies to improve the efficiency of the supply side.

The "information" component is an important element in evacuation model whose impact needs to be properly captured in modeling. Evacuation related information provided to evacuees by various public or private entities through various information dissemination channels such as commercial radio stations or highway advisory radios (HAR), TV programs, website, or dynamic message signs (DMS) often affect evacuees' evacuation decision prior to or during evacuation. Properly estimating the outcomes and impact of information is the key for a success evacuation modeling exercise.

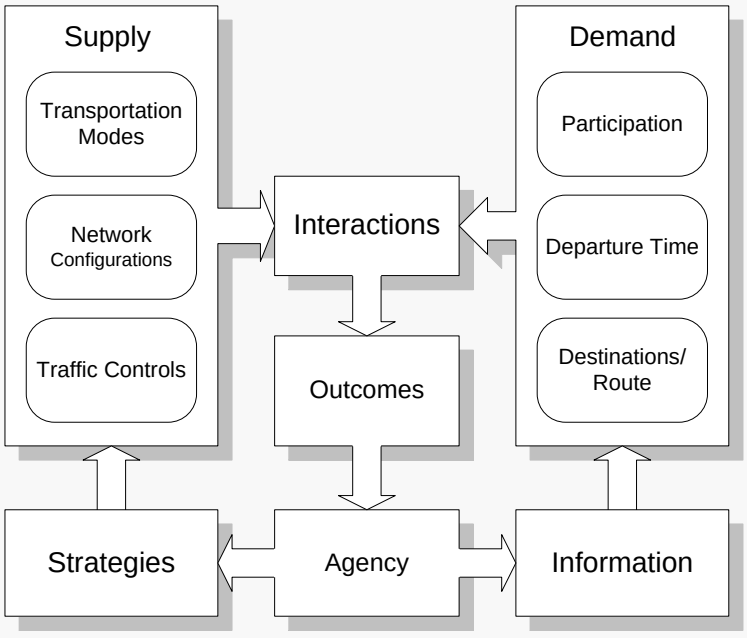



Figure 12 Demand-supply interaction for evacuation event modeled by DynusT (Mirchandani et al. 2009; Zheng et al. 2010)

### *5.3.1. Descriptive and Prescriptive Capabilities*

The overall DynusT evacuation modeling capability can be considered from either "descriptive" and/or "prescriptive" perspective. DynusT performs both the descriptive and prescriptive modeling and evaluation of various evacuation demand-supply conditions.

**Descriptive capability**

The descriptive capability means the use of the model to evaluate the outcome of the demand-supply conditions, evacuation management and information strategies specified by the user. As depicted in Figure 13, the descriptive capability is suitable for modeling "what-if" scenarios in which the user wishes to understand the outcomes by controlling how demand and supply conditions are specified.

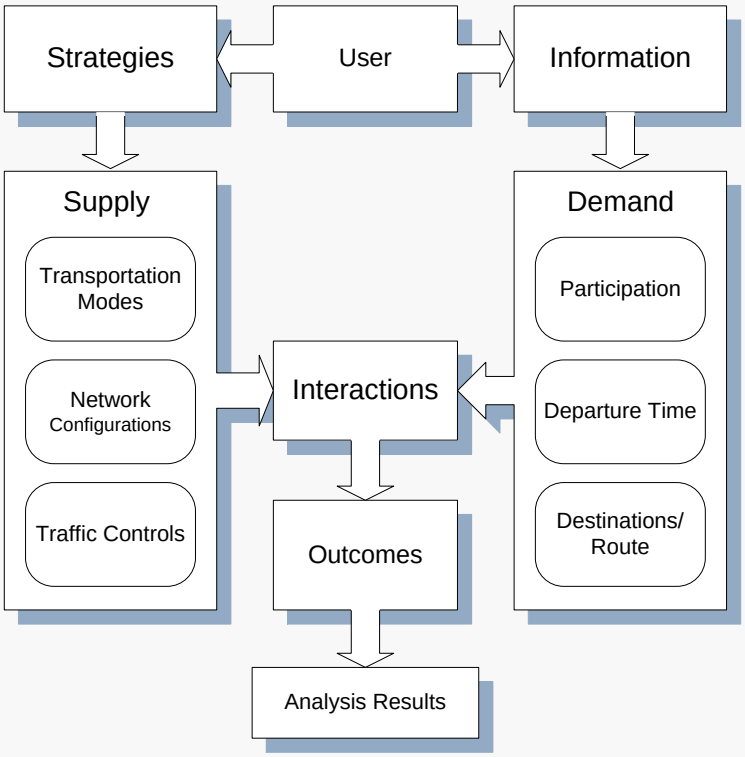



Figure 13 Descriptive evacuation modeling framework (Mirchandani et al. 2009; Zheng et al. 2010)

**Prescriptive capability**

The main difference between the prescriptive and the descriptive capability is that a user could specify (or adopt) an objective at which the optimal strategy would be sought for that

satisfy the set objective. The search of the optimal strategy is automatically performed by the model through the simulation of the demand-supply interactions as well as a solution algorithm as depicted in Figure 14.

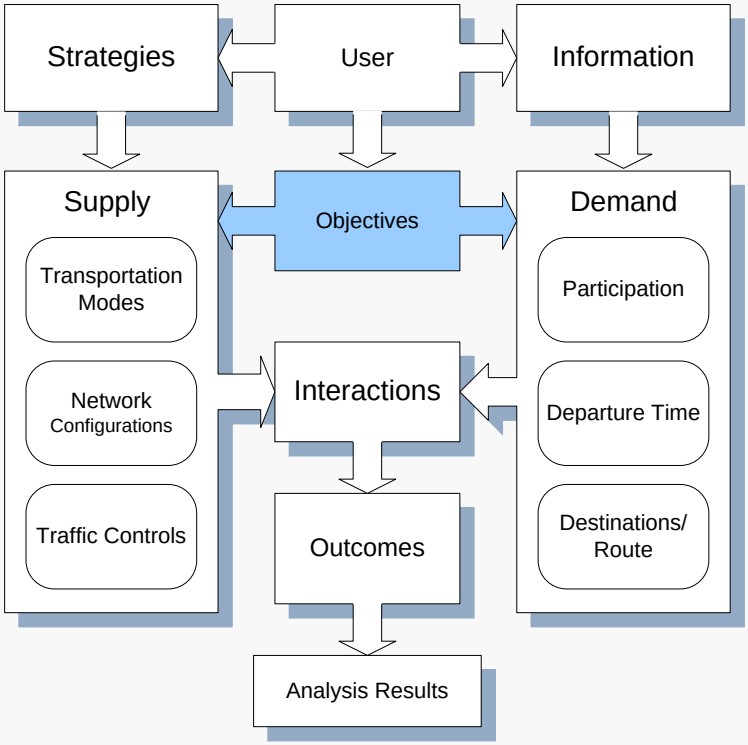



Figure 14 Prescriptive evacuation modeling framework (Mirchandani et al. 2009)

More specifically, in DynusT, the prescriptive capability is aimed at simultaneously solving for the optimal evacuation destinations, routes and flow splits from evacuation zones to safe zones. This scheme allows all regular traffic analysis zones (TAZ) to be identified as either evacuation, intermediate or safe zones for the evacuation purpose. As shown in the figure below, while the evacuation and intermediate zone topology remains unchanged, all boundary nodes in all safe zones (could be one of multiple) will be designated as physical evacuation destinations considered as gateway nodes and which are connected to a hypothetical sink node. This hypothetical sink node serves as the single super virtual destination for all evacuation flows. The evacuation trip generation will be estimated based on the new topology and the flow assignment problem of interest is transformed into a many-to-all network flow problem.

Corresponding trip demand information needs to reflect the evacuation zone to safe zone evacuation direction. Multiple safe zones defined in the original topology need to be aggregated

into one single zone for the OD demand matrix estimation purpose. This feature makes the trip distribution process simple and accurate since all the flow outbound of evacuation zones will just need to be pointed to this aggregated safe zone.

Next, the destination nodes are specified within the safe zones. Vehicles are considered safe upon arrival to these destination nodes. Destination nodes are those which are located at the perimeters of the boundary adjacent to evacuation zones or intermediate zones. By connecting these nodes to the virtual sink nodes, the evacuation flows will traverse only through these nodes to the virtual sink nodes. At this point, the modeling is completed and the optimal destination, routes and flow decision will be determined based on the modified topologies and associated OD trip information.

In the context of evacuation, DynusT allows the user to specify the background and evacuation demand separately. The background demand represents the traffic that is not affected by the disaster and still maintains certain regular activities. The evacuation demand represents the movement of traffic from the disaster impact area (or hot zone) to the safe zones. The background demand can be reasonably estimated based on the existing OD table (the user may need to apply certain scaling factors or conduct certain adjustments if necessary). The evacuation OD, in the context of a prescriptive model, can be estimated and specified by the DynusT user. Since both the background and evacuation demand have distinct spatial and temporal patterns, they need to be specified separately.

The evacuation demand is specified through the Super Zone layer. A super zone is defined as an aggregated zone that contains one or more original zones. Normally, in a mass evacuation, the origin of the evacuation may cover a large area (Chiu and Mirchandani 2008).

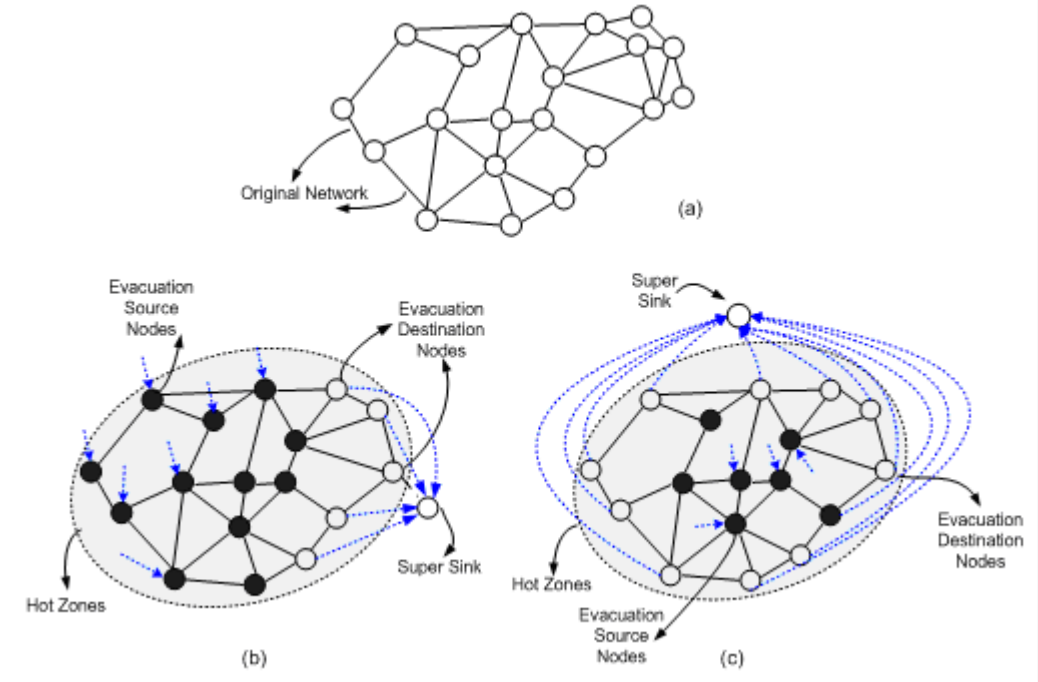



Figure 15 Single-destination for prescriptive modeling (Chiu et al. 2007)

### *5.3.2. Information Provision during Evacuations*

Information affects evacuees' departure and route choice decisions. As discussed above, departure curves can be developed by the model user based on the estimated time-varying departure patterns for evacuees. Although pre-trip information, with route and departure time advisories, is assumed to be made available through various news media, not all evacuees may access this information or follow the advisories. A user-specified market penetration rate indicates the percent of total evacuees who choose recommended routes and departure time; the rest still use their perceived optimal routes and departure times.

Once en route, information can further affect the re-route choices. The user of the platform can specify how frequently evacuation information is broadcast to en-route evacuees. A "belief" weight, accounting for road blockages, capacity reductions, past experiences and newly computed predicted travel times, can be assigned to a designated evacuation route to represent evacuees' confidence in the specified evacuation route, or specified re-routing, based on the given information.

Furthermore, DMS messages can also be modeled in the evacuation event. The model can specify a diversion rate consistent with the intensity of the sign language; "congestion ahead"

will result in minor diversion, whereas "major incident freeway closure, all vehicles exit" will result in complete diversion.

**Modeling of Flooding Information and Travel Decisions**

During the flooding evacuation, flooding information will be disseminated to the traveling public through various channels such as radio. Such information will influence a tripmaker's various traveling decisions. These decisions need to be incorporated into the simulation model in order to properly depict the traffic conditions resulting from these decisions. To incorporate this consideration, some additional decision rules were developed. First, it was assumed that flooding information was broadcast to the public at user-specified intervals. An evacuee departing from his evacuation zone would first head toward his intended destination. In the event that a driver learns that his route to the destination is blocked by the flood, then it is assumed that he will divert to one of the specified shelters.

For a non-evacuee facing the same situation, two conditions were modeled: (1) if the route is blocked after departure, the driver is assumed to return to his origin once he receives the radio information about the route blockage; and (2) if the driver's intended route is blocked before departure, he cancels his trip. Figure 16 illustrates the above modeling concepts.

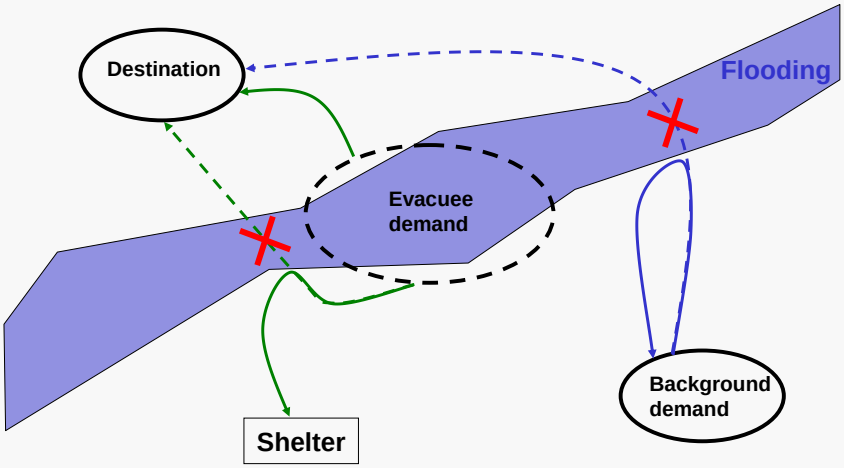
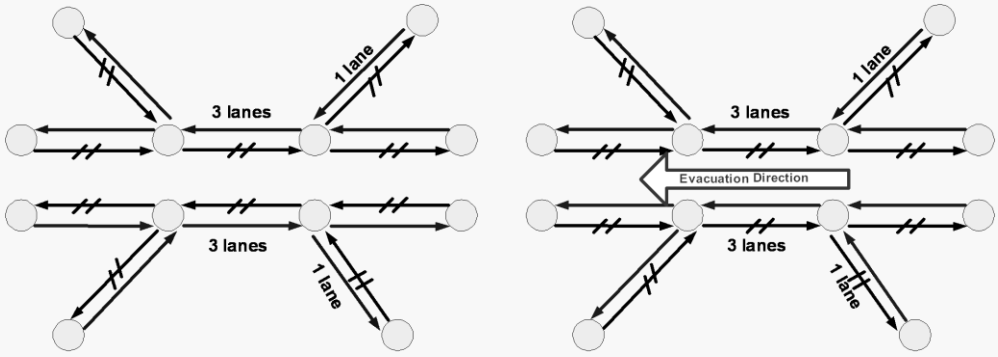



**Figure 16** Concept for Detouring due to Information Provided (Mirchandani et al. 2009; Noh et al. 2009)

**Contra-flow/Reversed Lane Modeling**

A unique modeling feature utilized in the platform is the activation and decommissioning of contra-flow lanes at designated time instances during simulation, which is accomplished by including in each directional freeway mainline or ramp lane, or each directional arterial lane, a coupling, counter-directional dummy link as depicted in Figure 7. Under normal traffic conditions, the dummy links are imposed with zero capacity, so that no evacuees would choose the route containing them. In the contra-flow case, the network structure remains, but some dummy links in the evacuation direction are provided with non-zero capacity, while their corresponding coupled links in the opposite direction are imposed with zero capacity. Required capacity for emergency vehicles in both directions is reassigned to areas needing such access. Furthermore, designated evacuation routes are assigned an "evacuation route attractiveness factor" to make these routes more likely to be used by the evacuating vehicles. The attractiveness factor needs to be calibrated using actual link counts so that the simulated counts are comparable to the actual traffic counts. This contra-flow feature is used in the developed simulation model.




(a) (b)

**Figure 17 Modeling Techniques for Contra-flow Lanes: (a) Normal Conditions, (b) Contra-flow Scenario (Chiu et al. 2008; Mirchandani et al. 2009)**

# 6. ONE-NORM LARGE-SCALE OD AND SPEED PROFILE CALIBRATION

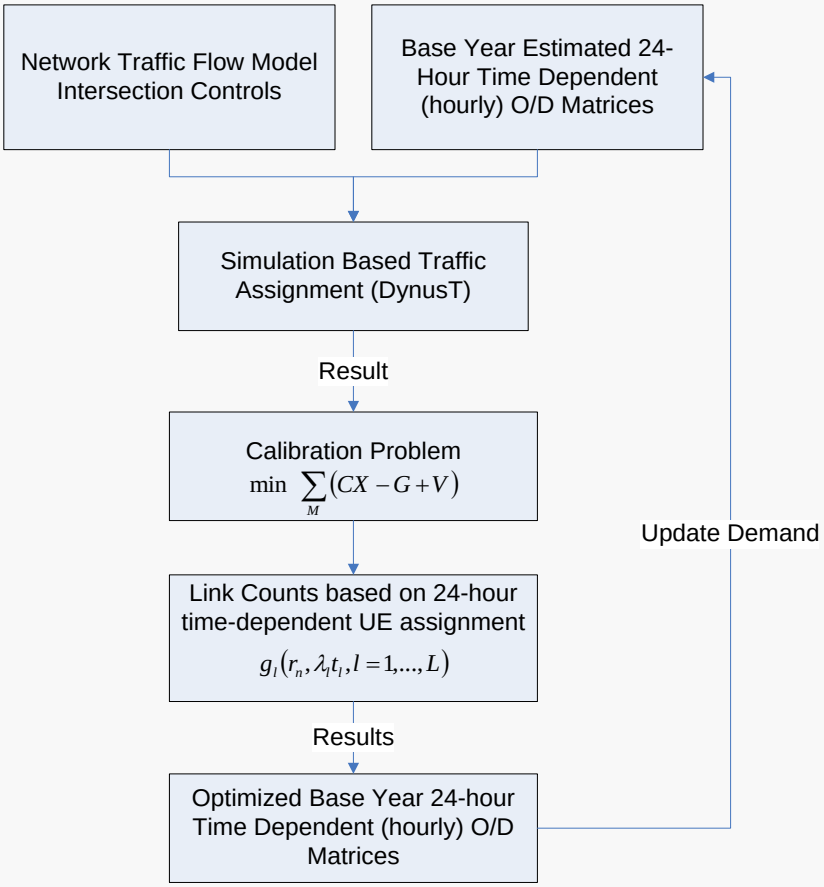



Figure 18

### *6.1.1. Formulation*

To be added

### *6.1.2. Calibration Results*

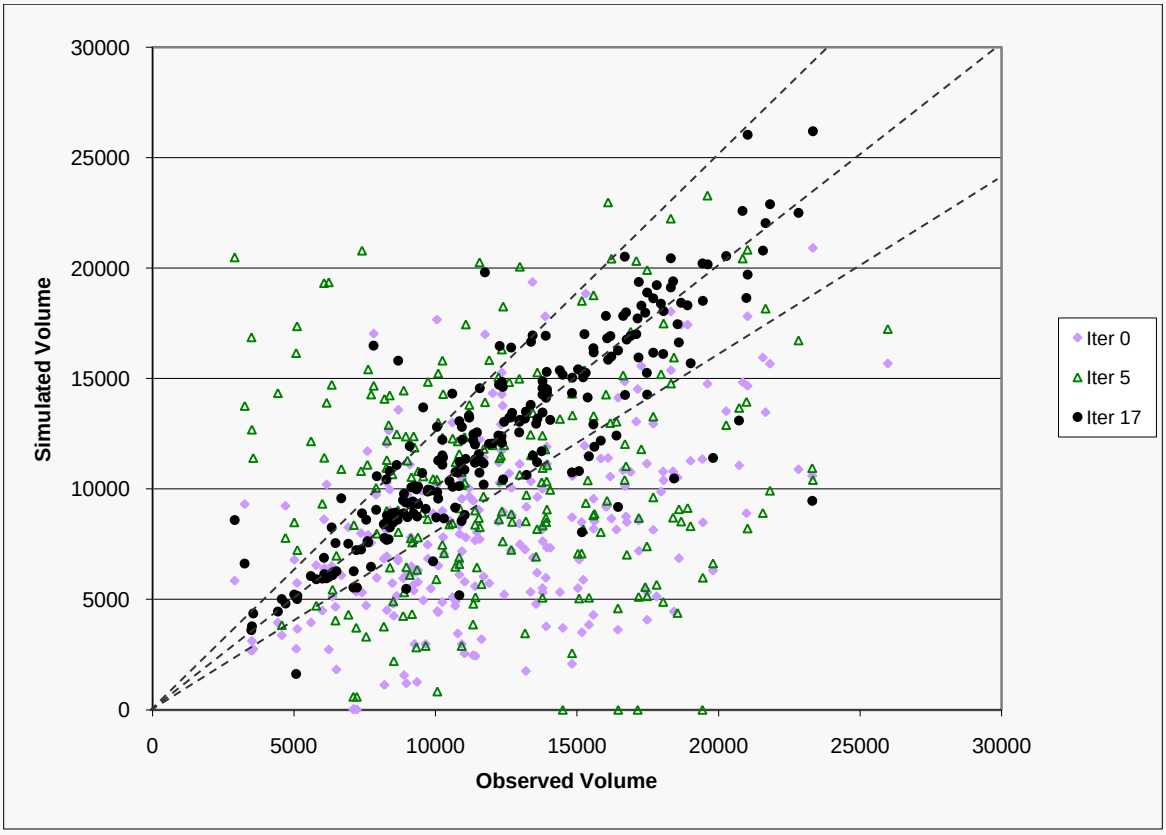
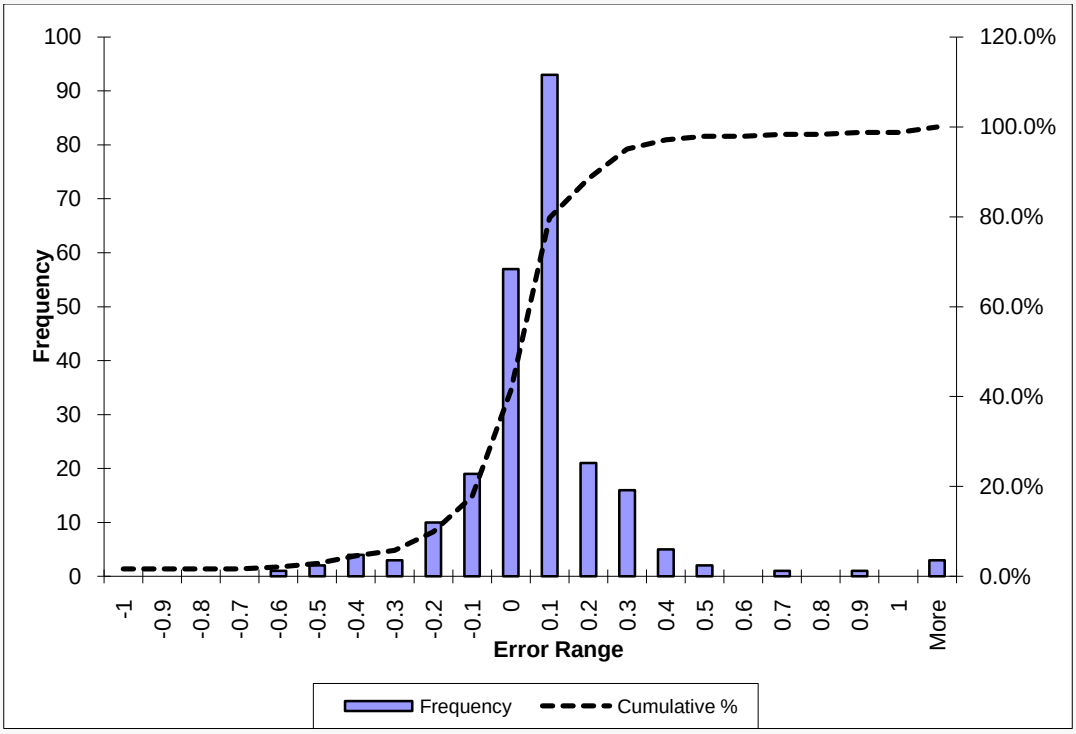



Figure 19



# 7. REFERENCE


Chiu, Y.-C., and Bustillos, B. "A Gap Function Vehicle-Based Solution Procedure for Consistent and Robust Simulation-Based Dynamic Traffic Assignment." *The 88th Annual Meeting of Transportation Research Board*, Washington, D.C.

Chiu, Y.-C., and Mirchandani, P. B. (2008). "Online Behavior-Robust Feedback Information Routing Strategy for Mass Evacuation." *IEEE Transaction on Intelligent Transportation Systems*, 9(2), 264-274.

Chiu, Y.-C., and Nava, E. (2010). "On the Development of Computationally Efficient and Solution-Consistent Solution Approach for Large-Scale Dynamic Traffic Assignment - Part 2: Method of Isochronal Vehicle Assignment." *Networks and Spatial Economics*, Under review.

Chiu, Y.-C., Zheng, H., Villalobos, J. A., and Gautam, B. (2007). "Modeling No-Notice Mass Evacuation Using a Dynamic Traffic Flow Optimization Model." *IIE Transactions*, 39(1), 83-94.

Chiu, Y.-C., Zheng, H., Villalobos, J. A., Henk, R., and Peacock, W. (2008). "Evaluating Regional Contra-Flow and Phased Evacuation Strategies for Central Texas Using a Large-Scale Dynamic Traffic Simulation and Assignment Approach." *Journal of Homeland Security and Emergency Management*, 5(1).

Chiu, Y.-C., Zhou, L., and Song, H. (2010). "Development and calibration of the Anisotropic Mesoscopic Simulation model for uninterrupted flow facilities." *Transportation Research Part B: Methodological*, 44(1), 152-174.

Mirchandani, P. B., Chiu, Y.-C., Hickman, M., Zheng, H., and Noh, H. (2009). "Platform for Evaluating Emergency Evacuation Strategies DRAFT Final Report 634." ATLAS Center, University of Arizona, Tucson.

Noh, H., Chiu, Y.-C., Zheng, H., Hickman, M., and Mirchandani, P. (2009). "An Approach to Modeling Demand and Supply for a Short-Notice Evacuation." *Transportation Research Record*, Forthcoming.

Zheng, H., Chiu, Y.-C., Mirchandani, P., and Hickman, M. "Modeling of Evacuation and Background Traffic for an Optimal Zone-Based Vehicle Evacuation Strategy." *the 89th Annual Meeting of Transportation Research Board*, Washington, D.C.