ENABLING CONGESTION AVOIDANCE AND REDUCTION IN THE MICHIGAN-OHIO TRANSPORTATION NETWORK TO IMPROVE SUPPLY CHAIN EFFICIENCY: Freight ATIS

PROGRESS REPORT

May 30, 2007

Principal Investigators:

Wayne State University
Industrial & Manufacturing Engineering
4815 Fourth Street, Detroit, MI 48202, USA

- Ratna Babu Chinnam (PI), Associate Professor
  Tel: 313-577-4846; Fax: 313-578-5902
  E-mail: r_chinnam@wayne.edu
- Alper E. Murat (Co-PI), Assistant Professor
  Tel: 313-577-4846; Fax: 313-578-5902
  E-mail: r_chinnam@wayne.edu

University of Detroit Mercy
School of Business Administration

- Gregory Ulferts (Co-PI), Professor
  Tel: 313-993-1219; Fax: 313-993-1673
  E-mail: ulfertgw@udmercy.edu

Submitted to:

Dr. Leo Hanifin
Director, MIOH University Transportation Center
University of Detroit Mercy

PROGRESS REPORT

Our project has three major mile-stones for the first year:

- Mile-stone #1: Data collection for MI-OH road network structure and historical incident data form MDOT, ODOT, and other agencies using ArcGIS software.
- Mile-stone #2: Developing Road Network Models representative of major freight transportation routes
- Mile-stone #3: Developing Static Re-routing optimization model and implementation for a limited set of scenarios

The Research Team at Wayne State University, made up of Dr. Ratna Babu Chinnam (Project PI), Dr. Alper Murat (Project Co-PI), two doctoral students - Ali Guner and Madhusalini Saripalle, and master's student Ajay Singh, has made very good progress with respect to all these three milestones over the past five months. While there has been regular communication and exchange with Dr. Gregory Ulferts at the University of Detroit Mercy, our plan for collaboration with UDM was to utilize UDM research team (including student assistants) for data collection (other than above) and data analysis. Given that we have completed in part the modeling and algorithm development steps, our next steps will be implementation and further design and development of models/algorithms. Hence, the UDM team will be a critical player in the remainder of the first year project execution. Specifically, they will assist us in the data collection, preparation and post-implementation analysis.

By far, the vast majority of our effort went toward developing dynamic routing algorithms that enable congestion avoidance and reduction in commercial cargo transportation networks. The rest of the document describes our efforts and results to date.

The report is organized as follows: Section I describes our efforts, results, and next steps in developing dynamic routing algorithms. Section II describes our collaborations with MITS-Center, METSIM, and MDOT Metro Regions to address mile-stones #1 and #2. Section III outlines our dissemination efforts.

I: DEVELOPING DYNAMIC ROUTING ALGORITHMS UNDER REAL-TIME INFORMATION

The overall goal here is to develop effective dynamic routing algorithms for congestion avoidance and reduction for commercial cargo carriers given real-time information regarding recurring and non-recurring congestion by Advanced Traveler Information Systems (ATIS). Vast majority of our R&D efforts over the past five months targeted this goal. We have extensively reviewed the literature on state-of-the-art dynamic routing algorithms, tested promising algorithms, recognized their strengths/weaknesses, and identified means to improve their performance.

The most promising algorithms identified to date came out of the following dissertation:

1. S. Kim, *Optimal Vehicle Routing and Scheduling with Real-Time Traffic Information*, Ph.D. Dissertation, University of Michigan, 2003.

Two journal articles followed:

2. S. Kim, M. E. Lewis and C. C. White, III, "Optimal vehicle routing with real-time traffic information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 2, pp. 178-188, June 2005.
3. S. Kim, M. E. Lewis and C. C. White, III, "State space reduction for non-stationary stochastic shortest path problems with real-time traffic information," *IEEE Transactions on Intelligent Transportation Systems*, vol. 6, no. 3, pp. 273-284, Sept. 2005.

While Kim made a significant contribution to the body of knowledge on dynamic routing algorithms that can utilize real-time information and reported substantial savings on route completion times using his algorithms, there are several significant short-comings with his algorithms. The most important short-coming is that there is no provision to account for congestion resulting from non-recurring events, such as accidents. In the past few months, not only have we implemented his optimal stochastic backward dynamic programming algorithms for dynamic routing, we have extended his methods by incorporating real-time information regarding non-recurrent events into the algorithms. We do this through an incident shock-wave model that becomes an integral part of the dynamic routing algorithm. This we believe is a significant achievement given the very short time-frame within which we achieved it. We have also begun to develop different computationally efficient shock-wave models for integration as well as computationally efficient heuristics for effective dynamic routing. The rest of this section outlines these algorithmic developments and some preliminary results that demonstrate their efficacy.

I.1. The Model

We consider a road network $G=(N,A)$ with node set $n \in N$ and arc/link set $l \in A$. Given an origin-destination node pair, (n_0, n_d), the decision problem faced by trip planner is to choose shortest path using real-time traffic information on traffic congestion state. Traffic congestion state on links can be at either of the two states: Uncongested and Congested. Traffic state on each link is classified according to Congested (Uncongested) depending on whether the velocity, $v_l(t)$, is less (greater) than a cut-off speed which is assumed 50 mph for all links.

$s_l(t)$:traffic congestion state of link *l* at time *t*, i.e. $s_l(t) = \{Congested, Uncongested\} = \{1,2\}$

$$s_l(t)=\begin{cases}1 & if \quad v_l(t)<50\,mph\\ 2 & if \quad v_l(t)\geq 50\,mph\end{cases}$$

Let $S(t)$ denote the traffic congestion state vector for the entire network, i.e., $S(t)=\{s_1(t),s_2(t),...,s_{|A|}(t)\}$ at time *t*. For presentation clarity, we will suppress the (t) in the notation whenever time reference is obvious from the expression.

It is further assumed that link traffic congestion states are independent from each other and have single-stage Markovian property. In order to estimate the state transitions for each link, two consecutive periods' velocities are modeled as bi-variate Gaussian distribution. Accordingly, time dependent state transition probabilities ($\alpha_{l,t},\beta_{l,t}$) are estimated from this velocity distribution.

$$T_l(t,t+1)=\begin{bmatrix}\alpha_{l,t} & 1-\alpha_{l,t}\\ 1-\beta_{l,t} & \beta_{l,t}\end{bmatrix}$$

$T_l(t,t+1)$: Single-period state transition probability intensities for congestion state of link *l* at time *t*.
Link *l*'s congestion state follows single-stage inhomogeneous Markov process, i.e., $\Pr\{s_l(t)\,|\,s_l(t-1),s_l(t-2),...,s_l(t_0)\}=\Pr\{s_l(t)\,|\,s_l(t-1)\}$ and is independent of other links' states, i.e. $\Pr\{s_{l_1}(t)\,|\,s_{l_2}(t),s_{l_1}(t-1)\}=\Pr\{s_{l_1}(t)\,|\,s_{l_1}(t-1)\}$.

With the normal distribution assumption of velocities, the time to travel on a link can be modeled as non-stationary normal distribution. We further assume that the link's travel time depends on the congestion state of the link at time of departure (equivalent to arrival time whenever there is no waiting). Since the congestion state is bi-form (congested vs. uncongested), travel time is determined according to the corresponding normal distribution.

$$\delta(t,l,s_l)\sim N\left(\mu_l(t,s_l),\sigma_l^2(t,s_l)\right)$$

where

$\delta(t,l,s_l)$: Travel time on link at time *t* with congestion state $s_l(t)$.

$\mu_l(t,s_l),\sigma_l(t,s_l)$: Mean and standard deviation of travel time on link *l* at time *t* with congestion state $s_l(t)$.

We assume that objective of dynamic routing is to minimize the expected travel time based on real-time information. The nodes (intersections) of the network represent decision points where a routing decision can be made. Since total trip travel time is an additive function of individual link travel times plus a penalty function measuring earliness/tardiness of arrival time to the final destination, dynamic route selection problem can be modeled as a dynamic programming model. In this model, decision epochs are the routing decisions at nodes. The state of the system $\Omega(n,t,S)$ is composed of the state of the vehicle and network and thus characterized by the current node (*n*), arrival time (*t*), and congestion state of the links (*S*). Action space for $\Omega(n,t,S)$ is the set of outgoing links from node *n*, i.e., $CLS(n)$. Let's define:

$SN(n)$: successor nodes set of node *n*, i.e., set of nodes with incoming arc from node *n*

$CLS(n)$: current links set of node *n*, i.e., set of outgoing links from node *n*

$SLS(n)$: successor links set of node *n*, i.e., set of outgoing links from the successor nodes of node *n*

At every decision node, the trip planner evaluates the alternatives based on the remaining expected travel time. The expected travel time at a given node is composed of expected travel time on the next outgoing link chosen and expected travel time of the next node. Let's $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$ be the set of policies for the trip where *K* is the number of decision stages, which is less than $\widehat{K}$, i.e., maximum number of nodes with more than two outgoing arcs among all paths. For a given state $\Omega(n,t,S)$, policy $\pi_k(\Omega)$ a deterministic Markov policy which chooses the outgoing link from node *n*, i.e., $\pi_k(\Omega) = l' \in CLS(n)$. Therefore the expected travel cost given the policy vector $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$ is as follows.

$$F_0(n_0, t_0, S_0) = \underset{\delta_k}{E}\left\{ g_K(\Omega_K) + \sum_{k=0}^{K-1} g_k(\Omega_k, \pi_k(\Omega_k), \delta_k) \right\}$$

where

t_k : Time of decision stage *k*

n_k : Node at decision stage *k*

S_k : Network congestion state at decision stage *k*, i.e., $S_k \equiv S(t_k)$

Ω_k : State of the system (vehicle and network) at decision stage k , i.e., $\Omega_k \equiv \Omega(n_k, t_k, S_k)$

δ_k : random travel time at decision stage *k*, i.e., $\delta_k \equiv \delta(t_k, \pi_k(\Omega_k), s_l(t_k))$

$g(\Omega_k, l', \delta_k)$: cost of travel on link $l' = \pi_k(\Omega) \in CLS(n_k)$ at stage *k*. For example, if travel cost is a function (ϕ) of the travel time, then $g(\Omega_k, \pi_k(\Omega_k), \delta_k) \equiv \phi(\delta_k)$

Minimum expected travel time can be found by minimizing $F(n_0, t_0, S_0)$ over the policy vector $\pi = \{\pi_1, \pi_2, ..., \pi_K\}$.

$$F^*(n_0, t_0, S_0) = \min_{\pi=\{\pi_1, \pi_2, ..., \pi_K\}} F(n_0, t_0, S_0)$$

where $\pi^* = \underset{\pi=\{\pi_1, \pi_2, ..., \pi_K\}}{\arg\min} F(n_0, t_0, S_0)$

Hence the Bellman (cost-to-go) equation for the dynamic programming model can be expressed as follows.

$$F^*(n_k, t_k, S_k) = \min_{\pi_k} \underset{\delta_k}{E}\left\{ g(\Omega_k, \pi_k(\Omega_k), \delta_k) + F^*(n_{k+1}, t_{k+1}, S_{k+1}) \right\}$$

or in short

$$F^*(\Omega_k) = \min_{\pi_k} \underset{\delta_k}{E}\left\{ g(\Omega_k, \pi_k(\Omega_k), \delta_k) + F^*(\Omega_{k+1}) \right\}$$

where

$n_{k+1} = \{ n \in N \mid \pi_k(\Omega_k) = l' \in CLS(n_k) \;\; and \;\; l' \equiv (n_k, n_{k+1}) \}$ and $t_{k+1} = t_k + \delta_k$

For a given policy decision $\pi_k(\Omega_k) = l'$, we can re-express the cost-to-go function by writing he expectation in explicit form.

$$F(n_k, t_k, S_k \mid l') = \sum_{\delta_k} \mathrm{P}(\delta_k \mid n_k, t_k, S_k)\left[g(\Omega_k, l', \delta_k) + \sum_{S_{k+1}} \mathrm{P}(S_{k+1}(t_{k+1}) \mid S_k(t_k)) F(n_{k+1}, t_{k+1}, S_{k+1}) \right]$$

where $\mathrm{P}(S_{k+1}(t_{k+1}) \mid S_k(t_k))$ is the $t_{k+1} - t_k = \delta_k$ period state transition probability which is found by calculating the following.

$$T_l(t_k, t_k + \delta_k) = \begin{bmatrix} \alpha_{l',t_k} & 1-\alpha_{l',t_k} \\ 1-\beta_{l',t_k} & \beta_{l',t_k} \end{bmatrix} \times \begin{bmatrix} \alpha_{l',t_{k+1}} & 1-\alpha_{l',t_{k+1}} \\ 1-\beta_{l',t_{k+1}} & \beta_{l',t_{k+1}} \end{bmatrix} \times \ldots \times \begin{bmatrix} \alpha_{l',t_k+\delta_k} & 1-\alpha_{l',t_k+\delta_k} \\ 1-\beta_{l',t_k+\delta_k} & \beta_{l',t_k+\delta_k} \end{bmatrix}$$

$\mathrm{P}(\delta_k \mid n_k, t_k, S_k)$ is the probability of travel time δ_k on link l' which can be calculated from $\delta(t_k, l', s_{l'}) \sim N\left(\mu_{l'}(t_k, s_{l'}), \sigma^2_{l'}(t_k, s_{l'})\right)$.

Using the backward induction we could solve $F_k^*(\Omega_k)$ for *k=K,K-1,…0*. In the next section we describe the implementation on a test problem.

I.2. Example Implementation

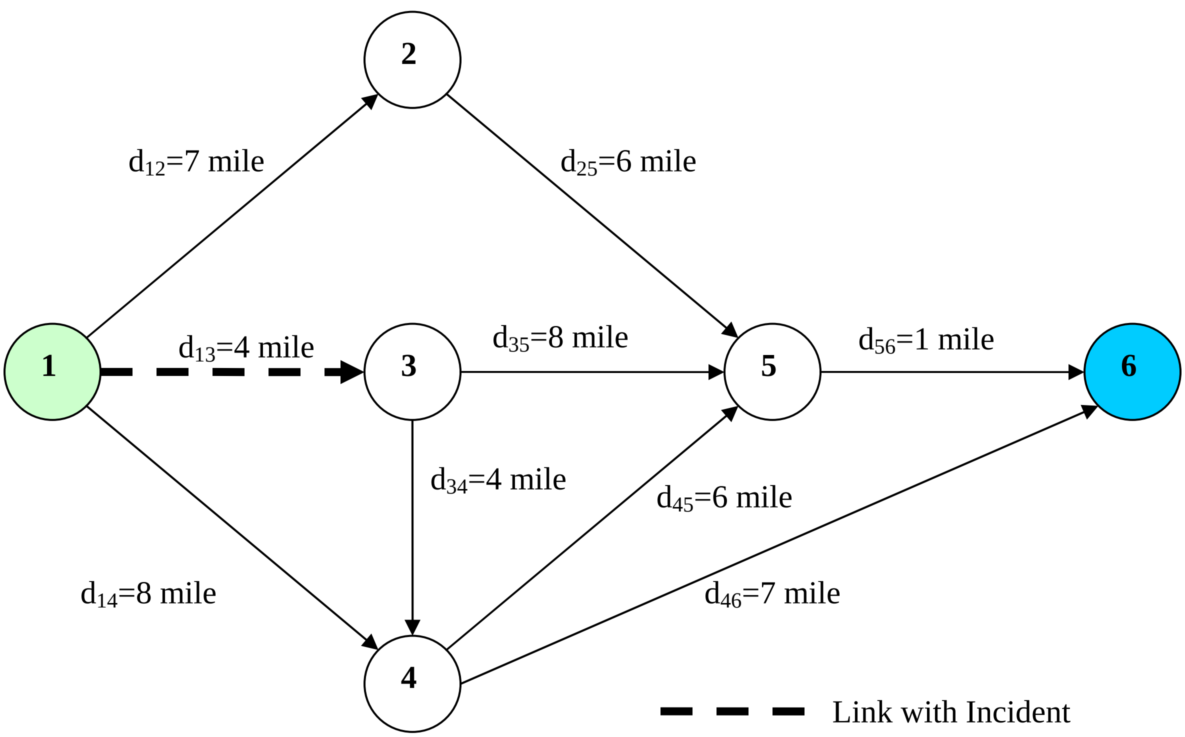


Figure 1. Example network with 6 nodes and 10 links.

Consider a stylized network example with 6 nodes and 10 links in Figure 1. Links lengths are as shown on the graph. Note that we are using the following referencing for links $l_{n_o^l n_d^l}$ where n_o^l and n_d^l are origin and destination nodes of link *l*, respectively.

We consider Node 1 as the origin node and Node 6 as the destination node of the trip. Velocities on the links follow stochastic non-stationary distributions which vary with the time of the day. Velocity data is sampled for every 15-minute interval and then linearly interpolated on a minute basis. Traffic state on each link is classified according to Congested (Uncongested) depending on whether the velocity is less (greater) than a cut of speed which is assumed 50 mph for all links. Our assumptions in this implementation are as follows (mostly consistent with Kim, 2003):

1. Traffic congestion on links can be at either of the two states: Uncongested and Congested
2. Links' congestion state follow Markov process and evolve independent of each other.
3. Link velocities follow non-stationary normal distribution
4. Travel times follow normal distribution
5. Link travel times are calculated based on the link's congestion state at the time of arrival
6. Acyclic network
7. No waiting at the nodes
8. Incidents take place on the links and affect link travel time of the corresponding link, i.e., no secondary shockwave affect on the remainder of the network
9. Travel cost is proportionate to the travel time
10. All links in the network are observed through traffic flow sensors

Recall that the state vector in the previous section, $\Omega(n,t,S)$, includes congestion states of all links in the network, i.e., $S(t)=\{s_1(t),s_2(t),...,s_{|A|}(t)\}$. When no real-time information is used then the state vector would be $\widetilde{\Omega}(n,t)$ and the trip planning problem becomes a static routing problem with time-dependent stochastic travel times. Including state of all network links in the state vector is computationally challenging due to exponentially increasing size of the state space. Therefore we include the states of links which are more important than the others. Specifically, given a node where a routing decision to be made, we include states of links (called *current links*) outgoing from that node and links that are successors of current links (called *successor links*). Next we define the following for exposition clarity. First, for nodes we define the following sets:

$SN(n)$*: Successor Nodes Set* of node *n*, i.e., set of nodes with incoming arc from node *n*

$NCLS(n)$*: Current Links Set* of node *n*, i.e., set of outgoing links from node *n*

$NSLS(n)$*: Successor Links Set* of node *n*, i.e., set of outgoing links from the successor nodes of node *n*

Let *Node Links Set* of node *n* as $NLS(n)=\{NCLS(n)\cup NSLS(n)\}$.

Following examples illustrate these sets $SN(n=1)=\{2,3,4\}$, $NCLS(n=1)=\{l_{12},l_{13},l_{14}\}$, and $NSLS(n=1)=\{l_{25},l_{35},l_{34},l_{45},l_{46}\}$.

Next, we define the following sets for links:

$LSLS(l)$*: Successor Links Set* of link *l*, i.e., set of outgoing links from the destination node of *l*

$LPSLS(l)$*: Post-Successor Links Set* of link *l*, i.e., set of outgoing links from the destination nodes of all links $l'\in LSLS(l)$.

Following examples illustrate these sets $NCLS\left(n=1\right)=\{l_{12},l_{13},l_{14}\}$, $LSLS\left(l_{13}\right)=\{l_{34},l_{35}\}$, and $LPSLS\left(l_{13}\right)=\{l_{45},l_{46},l_{56}\}$

Note that for a given $l\in NCLS\left(n\right)$, $LPSLS\left(l\right)\subseteq NSLS\left(n\right)$. Furthermore, for given a link $l\in A$ and $\left(n_0^l,n_d^l\right)$, the successor links set of link's destination node is equivalent to the union of successor and post-successor links set of the link *l*, i.e., $NSLS\left(n_d^l\right)=\left\{LSLS\left(l\right)\cup LPSLS\left(l\right)\right\}$.

Our backward dynamic programming algorithm requires structuring a tree form (called *stage tree*) to implement stage-wise backward induction. To obtain the stage tree, we convert a given network such that at every stage there is one outgoing arc from any node. Therefore in a stage tree, a node can appear in multiple stages. Figure 2 illustrates the stage tree for the example network.

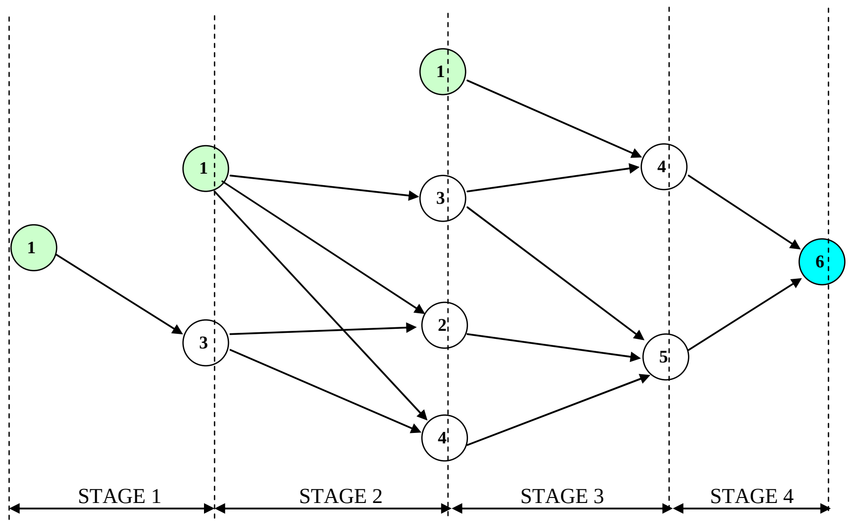


Figure 2. Illustration of stage tree for the example network in Figure 1.

Finally define $A\left(k\right)$ as the set of links at decision stage k. For instance, $A\left(k=3\right)=\left\{l_{14},l_{34},l_{35},l_{25},l_{45}\right\}$. We now provide our backward dynamic programming algorithm.

I.2.A. The Algorithm

We now provide the algorithm for recurring congestion. In the end of this section we describe the changes in this algorithm to accommodate for non-recurring congestion namely incidents that affect link travel time.

Backward DP Algorithm for Dynamic Routing with Real-Time Information

Step 1. Initializations

- Initialize network $G=(N,A)$, origin/destination nodes of links and trip, latest arrival time to destination M, and successor nodes sets $SN(n)$ for $\forall n \in N$
- Generate tree form of decision making stages for backward induction
- Generate link lists for nodes and links
- Generate $NCLS(n)$ and $NSLS(n)$ for $\forall n \in N$
- Generate $LSLS(l)$ and $LPSLS(l)$ for $\forall l \in A$
- Generate velocity distributions for all links
- Set length of unit time period t
- Interpolate velocity for each time unit t
- Partition velocities into congestion states. Fit distributions for each congestion state, link, and time and estimate distribution parameters such as mean and standard deviation
- Set maximum link travel times, i.e. $\delta_l^{\max} = \max\limits_{t,s_l} \delta(t,l,s_l)$
- Calculate latest arrival times ($AT_{\max}^{n}$) for each for $\forall n \in N$ such that $AT_{\max}^{n} \leq \min\limits_{\substack{n' \in SN(n) \\ l \equiv (n,n')}} \left(AT_{\max}^{n'} - \delta_l^{\max}\right)$ and $AT_{\max}^{n_d} = M$
- Estimate the single step congestion state transition probabilities $T_l(t,t+1)$ by fitting bi-variate Gaussian distribution for every consecutive time-period velocity pair for $\forall l \in A$.
- Calculate steady state probabilities of congestion states $\forall l \in A$
- Set probability tolerance parameter $\varepsilon_{Prob} = 10^{-6}$

Step 2. Backward Dynamic Program

Step 2.a.

For k=K to 1, Repeat \\ starting from the last stage

For $\forall l \in A(k)$, Repeat \\ for link in the current stage

For $t_k = t_0$ to $AT_{\max}^{n_0^l}$, Repeat \\ for every travel starting time on the current link

For $s_l(t_k) = \{1,2\}$, Repeat \\ for every congestion state of the current link

For $s_{l'}(t_k) = \{1,2\} \quad \forall l' \in \{LSLS(l) \cup LPSLS(l)\}$, Repeat \\ for every state \\ prior to link l travel

GoTo Step 2.b.

Set $F\left(n_0^l, t_k, s_{l \in NLS(n)}(t_k)\right) = \min\left(\widehat{F}\left(n_0^l, t_k, s_l(t_k) \mid l\right), F\left(n_0^l, t_k, s_{l \in NLS(n)}(t_k)\right)\right)$

Set $\pi\left(n_0^l, t_k, s_{l \in NLS(n)}(t_k)\right) = l \;\; if \;\; \widehat{F}\left(n_0^l, t_k, s_l(t_k) \mid l\right) \leq F\left(n_0^l, t_k, s_{l \in NLS(n)}(t_k)\right)$

Next $s_{l'}(t_k)$

Next $s_l(t_k)$

```
      Next  t_k
    Next l
Next k
```

Step 2.b. Calculation of the cost-to-go for a given action $l=\pi_k\left(\Omega_k\right)$ for the current state $\Omega_k=\left(n_0^l,t_k,s_l\left(t_k\right)\right)$

Set $\widehat{F}\left(n_0^l,t_k,s_l\left(t_k\right)\mid l\right)$:=0

For $\delta_k=1$ to $\left(AT_{\max}^{n_d^l}-t_k\right)$, Repeat \\ for every travel time on link *l*

  For $s_{l'}\left(t_{k+1}=t_k+\delta_k\right)=\{1,2\}\quad\forall l'\in\left\{LSLS\left(l\right)\cup LPSLS\left(l\right)\right\}$, Repeat \\ for every state \\ following link *l* travel

    Calculate $g(\Omega_k,l,\delta_k)$ where $\Omega_k=\left(n_0^l,t_k,s_l\left(t_k\right)\right)$ \\ cost of travel on the current link *l*

    Calculate $\mathrm{P}\left(\delta_k\mid n_0^l,t_k,s_l\left(t_k\right)\right)$ \\ probability of travel time δ_k on the current link *l*

    Calculate $\mathrm{P}\left(s_{l'}\left(t_k+\delta_k\right)\mid s_{l'}\left(t_k\right)\right)\ \forall l'\in\left\{LSLS\left(l\right)\cup LPSLS\left(l\right)\right\}$ \\ transition probabilities

    If $\mathrm{P}\left(\delta_k\mid n_0^l,t_k,s_l\left(t_k\right)\right)\le\varepsilon_{Prob}$ or $\mathrm{P}\left(s_{l'}\left(t_k+\delta_k\right)\mid s_{l'}\left(t_k\right)\right)\le\varepsilon_{Prob}$, then Next δ_k

    Recall $F\left(n_d^l,t_k+\delta_k,s_{l'}\left(t_k+\delta_k\right)\right)$ \\ cost-to-go for the destination \\ node of the current link *l*

    Calculate

$$\widehat{F}\left(n_0^l,t_k,s_l\left(t_k\right)\mid l\right)+=\left[\prod_{l'\in\left\{LSLS(l)\cup LPSLS(l)\right\}}\mathrm{P}\left(s_{l'}\left(t_k+\delta_k\right)\mid s_{l'}\left(t_k\right)\right)\right]\mathrm{P}\left(\delta_k\mid n_0^l,t_k,s_l\left(t_k\right)\right)\times\left[\begin{array}{l}g_k(\Omega_k,l,\delta_k)\\+F\left(n_d^l,t_k+\delta_k,s_{l'}\left(t_k+\delta_k\right)\right)\end{array}\right]$$

  Next $s_{l'}\left(t_k\right)$

Next δ_k

Return to Step 2.a.

For the non-recurring congestion case we modify the above algorithm by there is an incident, we modify the step 2.b. of the above algorithm. We now illustrate the modification with a simple integration of incident information in the algorithm. Let's assume that there is an incident on link l_{13} at the start of the trip. This incident brings about a delay in the travel time of link l_{13}. This incident delay (θ) is a function of link(*l*), time(*t*), initial incident severity(κ), incident stage(ρ), i.e., $\theta\left(l,t,\kappa,\rho\right)$. Here we assume that $\theta\left(l,t,\kappa,\rho\right)=\theta$=constant for this simple case. Then the step 2.b. of the previous algorithm is modified as below.

Modification of Step 2.b. of the Backward DP Algorithm for Dynamic Routing with Real-Time Information and Incidents

Step 2.b. Calculation of the cost-to-go for a given action $l=\pi_k\left(\Omega_k\right)$ for the current state $\Omega_k=\left(n_0^l,t_k,s_l\left(t_k\right)\right)$

```
Set  F̂(n_0^l, t_k, s_l(t_k) | l) :=0
For  δ_k = 1 to (AT_max^{n_d^l} − t_k), Repeat  \\  for every travel time on link l
  For  s_l'(t_{k+1} = t_k + δ_k) = {1,2}  ∀l' ∈ {LSLS(l) ∪ LPSLS(l)} , Repeat  \\ for every state following link l travel
        Calculate  g(Ω_k, l, δ_k)  where  Ω_k = (n_0^l, t_k, s_l(t_k))  \\ cost of travel on the current link l
        Calculate  P(δ_k | n_0^l, t_k, s_l(t_k))  \\ probability of travel time  δ_k  on the current link l
        Calculate  P(s_l'(t_k + δ_k + θ) | s_l'(t_k))  ∀l' ∈ {LSLS(l) ∪ LPSLS(l)} \\  transition probabilities
        If  P(δ_k | n_0^l, t_k, s_l(t_k)) ≤ ε_Prob  or  P(s_l'(t_k + δ_k + θ) | s_l'(t_k)) ≤ ε_Prob , then Next  δ_k
        Recall  F(n_d^l, t_k + δ_k + θ, s_l'(t_k + δ_k + θ))  \\ cost-to-go for the destination node of the current link l
        Calculate
```

$$\widehat{F}\left(n_0^l,t_k,s_l\left(t_k\right)|\,l\right)+=\left[\prod_{l'\in\left\{LSLS(l)\cup LPSLS(l)\right\}}\mathrm{P}\left(s_{l'}\left(t_k+\delta_k+\theta\right)|\,s_{l'}\left(t_k\right)\right)\right]\mathrm{P}\left(\delta_k\,|\,n_0^l,t_k,s_l\left(t_k\right)\right)\times\left[\begin{array}{l}g_k(\Omega_k,l,\delta_k)+\theta\\+F\left(n_d^l,t_k+\delta_k+\theta,s_{l'}\left(t_k+\delta_k+\theta\right)\right)\end{array}\right]$$

```
  Next  s_l'(t_k)
Next  δ_k
Return to Step 2.a.
```

I.2.B. Computational Results

We have implemented the algorithm in the previous section to the test problem and identified the optimal policies $\pi^*\left(1,t_0,S\left(t_0\right)\right)$. The maximum trip time parameter M is set as M=50. Accordingly the latest node arrival times are $AT_{\max}^n=\left\{1,34,25,35,48,50\right\}$. Trip start time is set at $t_0=1$.

We simulated the trip 1000 times for the starting state of $S\left(t_0=1\right)=\left\{1,1,1,1,1,1,1,1,1,1\right\}$. Expected travel time is found to be 17.60 minutes with a standard deviation of 1 minute. Distribution of the travel time is illustrated in Figure 3.

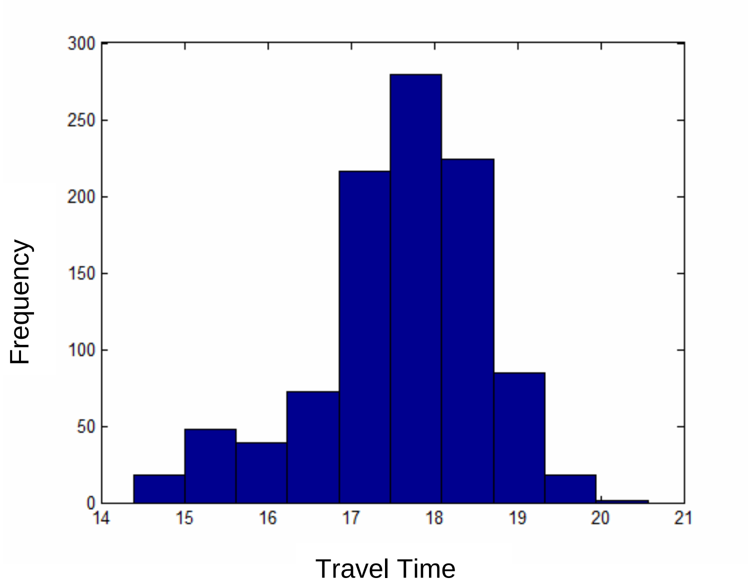


Figure 3. Distribution of travel time with recurring congestion.

Note that there are six alternative paths from origin to destination node in Figure 1. As a result of this simulation, the following paths are chosen with the following frequencies.

Path 1: $\{1 \to 3 \to 5 \to 6\}$, Frequency=155 times
Path 1: $\{1 \to 3 \to 4 \to 6\}$, Frequency=559 times
Path 1: $\{1 \to 3 \to 4 \to 5 \to 6\}$, Frequency=286 times

Next, we have implemented the algorithm with an incident on link l_{13} at the start of the trip. We set the parameters such that this incident brings about a delay of θ=10 minutes in the travel time of link l_{13}. We solved the problem using the modified step 2.b. for incident delay and identified the optimal policies. Then we re-simulated the trip for another 1000 times with the same starting state. This time expected travel time is found as 18.33 minutes with a standard deviation of 0.74 minutes. Distribution of the travel time with incident delay is illustrated in Figure 4.

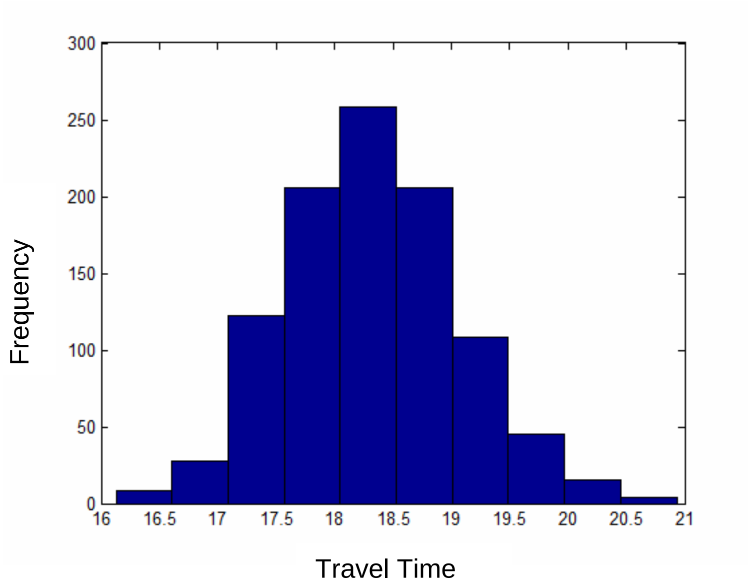


Figure 4. Distribution of travel time with non-recurring congestion.

As a result of this simulation, the following paths are chosen with the following frequencies.

Path 1: $\{1 \rightarrow 4 \rightarrow 5 \rightarrow 6\}$, Frequency=276 times
Path 1: $\{1 \rightarrow 4 \rightarrow 6\}$, Frequency=724 times

Because of the incident induced delay, our algorithm identified alternative routed to avoid the link l_{13}.

I.3. Summary and Next Steps

Over the course of this multi-year project, we first intend to relax many of the assumptions made by our current algorithms (e.g., allow traffic conditions on adjacent links to be correlated and not treat them to be independent). Subsequently, our goals (nor prioritized) are as follows:

- Extend the dynamic routing algorithms to account for: Cycles within the network, Incidents on distant links, and Travel within larger networks through modifications to stage tree construction procedure (that accounts for incoming links to origin node and outgoing links from destination node and existence of nodes and links not practically reachable from the point of origin).
- Methods for estimation of travel time (or cost) variance besides the expected travel time (or cost).
- Heuristics for computational/memory efficiency; In particular, our first preference here is to adapt the A* algorithm for our needs.
- Develop computationally efficient yet effective shockwave models that account for shockwaves propagating through nodes.

Our ultimate goal will be to develop dynamic routing algorithms interfaced with real-time commercial GIS software that support constraints (e.g., user request to avoid local roads) and trips with multiple-legs (e.g., to support milk-run shipments).

II: COLLABORATIONS WITH MITS-CENTER, METSIM, AND MDOT METRO REGIONS

II.1 Collaboration with MITS-Center and Traffic.com

We have to date had multiple meetings with Mia Silver of MITS Center and her team to develop a better understanding for their traffic monitoring system (for southeast Michigan highways) and have also received some preliminary data representing several weeks of traffic for the southeast Michigan highways. We are currently in the process of analyzing this data to improve the quality of the models being developed for dynamic routing decision support when operating with access to ATIS (Advanced Traveler Information Systems) information.

We had some preliminary discussions with Traffic.com and they have expressed their willingness to provide data for majority of high-ways in the Southeast-Michigan corridor. These datasets (and the networks resulting from them) will play a critical role for evaluating and refining our dynamic routing algorithms.

II.2 Collaboration with METSIM and MDOT Metro Region

METSIM and Gateway projects, headed by Matt Webb of METSIM and Catherine Jensen at MDOT Metro Region, respectively, are two projects that aim to develop tools for strategic and operational planning of highway projects through micro-simulation models. Both of these projects are utilizing the Paramics Suite software package for traffic simulation. The METSIM model is nearly complete undergoing extensive calibration studies for simulation accuracy. The Gateway project extends the METSIM model to include major local roads in the Metro Region. Both Matt and Catherine are very supportive of our research efforts and plan to share nearly fully-specified micro-simulation models for the southeast Michigan corridor to test our dynamic routing decision models as well future analysis on the effect of traffic incidents (i.e. accidents, breakdowns) on the delivery reliability within JIT supply chain operations. Both the METSIM model and the Gateway project model will be available by late summer. After consulting with Matt and Catherine and researching other traffic simulation software alternatives, we have decided to use Paramics for two reasons. First, it is versatile and allows us to incorporate our dynamic re-routing algorithms. Second, we will be able to use the traffic simulation models produced by the two projects. We have also been in contact with the company that provides Paramics Suite software. The software is relatively expensive, even for an academic license (priced between $5k and $10k depending on options). We are trying to identify other groups at Wayne State that might be interested in jointly purchasing a license.

III: RESULTS DISSEMINATION

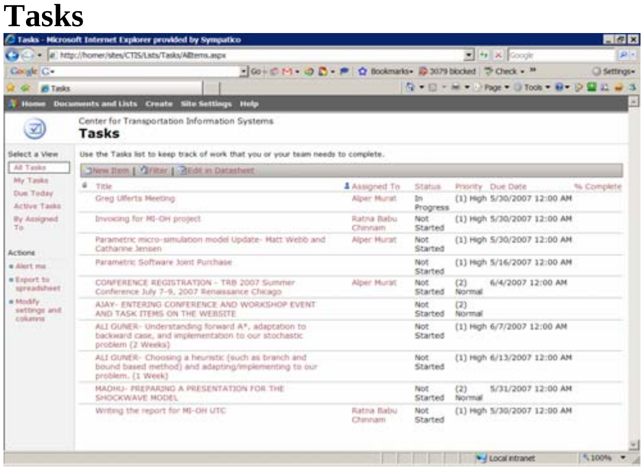
III.1 Project Website

We have established a Microsoft SharePoint Website for the project that helps us track/store all project related documents/information in one place. Currently, it carries all our literature, data sets, code, weekly research group meeting minutes, long-term mile-stones, short-term tasks, calendar, and contacts. While we currently control access to this website through password protection, we are in the process of opening parts of the website for anonymous access. The screen shots below highlight different parts of our website.

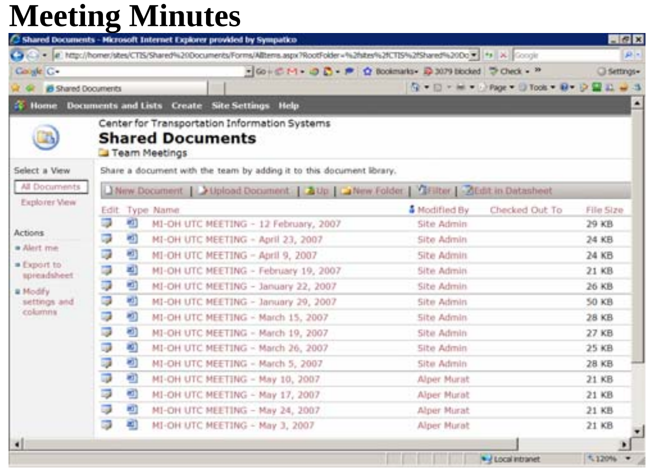
Homepage

Literature

Tasks



Meeting Minutes



III.2 Conference Activity

Conferences Attended:
Drs. Chinnam and Murat attended the Michigan Intelligent Transportation Systems Conference, May 16-17, 2007. Dr. Khasnabis, Associate Dean for Research College of Engineering has presented on our project along with other WSU efforts related to transportation.

Conferences Planning to Attend:
1. 2007 TRB Summer Conference July 7-9 and 32nd Annual TRB Summer Ports, Waterways, Freight & International Trade Conference, Joint Conference, 2007 Chicago
2. Meeting Freight Data Challenges July 9–10, 2007 Renaissance Chicago Hotel, Chicago
3. European Transport Conference, October 17-19, 2007 Leiden, The Netherlands
4. Research Issues in Freight Transportation -- Congestion and System Performance October 22-23, 2007 Washington, DC
5. 2nd Annual National Urban Freight Conference- December 5-7, 2007 Hyatt Regency Long Beach CA
6. TRB 87th Annual Meeting- January 13-17, 2008 Washington, DC

III.3 Journal Publications

Two journal manuscripts are currently under preparation to publish algorithmic developments and results to date. One manuscript will be dispatched this summer and the other by the end of this year.