

# Dynamic Routing using Real-time ITS Information

Ali R. Güner, Alper Murat and Ratna Babu Chinnam

Department of Industrial and Manufacturing Engineering, Wayne State University  
4815 Fourth St. Detroit, MI 48202, U.S.A.

**Abstract.** On-time delivery is a key performance measure for dispatching and routing of freight vehicles in just-in-time (JIT) manufacturing environments. Growing travel time delays and variability, attributable to increasing congestion in transportation networks, are negatively impacting the efficiency of JIT logistics operations. Recurrent congestion is one of the primary reasons for delivery delay and variability. In this study, we propose a stochastic dynamic programming formulation for dynamic routing of vehicles in non-stationary stochastic networks subject to recurrent congestion. Results are very promising when the algorithms are tested in a simulated network of Southeast-Michigan freeways using historical Intelligent Transportation Systems (ITS) data.

## 1 Introduction

Supply chains that rely on just-in-time (JIT) production and distribution require timely and reliable freight pick-ups and deliveries from the freight carriers in all stages of the supply chain. However, road transportation networks are experiencing ever growing travel time delays, which greatly hinders all travel and certainly the freight delivery performance. Travel time delays are mostly attributable to the so called 'recurrent' congestion that, for example, develops due to high volume of traffic seen during peak commuting hours. The standard approach to deal with congestion is to build additional 'buffer time' into the trip (i.e., starting the trip earlier so as to end the trip on time). Intelligent Transportation Systems (ITS) are providing real-time traffic data (e.g., lane speeds and volumes) in many urban areas. In-vehicle communication technologies, such as satellite navigation systems, are also enabling drivers' access to this information en-route. In this paper, we precisely consider JIT pickup/delivery service, and propose a dynamic vehicle routing model that exploits real-time ITS information to avoid recurrent congestion.

Our problem setting is the non-stationary stochastic shortest path problem with recurrent congestion. We propose a dynamic vehicle routing model based on a Markov decision process (MDP) formulation. Stochastic dynamic programming is employed to derive the routing 'policy', as static 'paths' are provably suboptimal for this problem [1]. The MDP 'states' cover vehicle location, time of day, and network congestion state(s). Recurrent network congestion states and their transitions are estimated from the ITS historical data. The proposed framework employs Gaussian mixture model based clustering to identify the number of states and their transition rates, by time of day, for each arc of the traffic network. To prevent exponential

growth of the state space, we also recommend limiting the network monitoring to a reasonable vicinity of the vehicle.

The rest of the paper is organized as follows. Survey of relevant literature is given in section 2. Section 3 establishes modeling recurrent congestion and dynamic vehicle routing for the problem. Section 4 presents experimental settings and discusses the results. Finally, section 5 offers some concluding remarks.

## 2 Literature Survey

Shortest path problems with stochastic and time-dependent arc costs (STD-SP) are first studied by Hall [1]. Hall showed that the optimal solution has to be an ‘adaptive decision policy’ (ADP) rather than a single path. Hall [1] employed dynamic programming (DP) approach to derive the optimal policy. Later, Fu [2] discussed real-time vehicle routing based on the estimation of immediate arc travel times and proposed a label-correcting algorithm as a treatment to the recursive relations in DP. Waller and Ziliaskopoulos [3] suggested polynomial algorithms to find optimal policies for stochastic shortest path problems with one-step arc and limited temporal dependencies. For identifying paths with the least expected travel (LET) time, Miller-Hooks and Mahmassani [4] proposed a modified label-correcting algorithm. Miller-Hooks and Mahmassani [5] extends [4] by proposing algorithms that find the expected lower bound of LET paths and exact solutions by using hyperpaths.

All of the studies on STD-SP assume deterministic temporal dependence of arc costs, with the exception of [3] and [6]. Polychronopoulos and Tsitsiklis [7] is the first study to consider stochastic temporal dependence of arc costs and to suggest using online information en route. They defined environment state of nodes that is learned only when the vehicle arrives at the source node. They considered the state changes according to a Markovian process and employed a DP procedure to determine the optimal policy. Kim et al. [8] studied a similar problem as in [7] except that the information of all arcs are available real-time. They proposed a DP formulation where the state space includes states of all arcs, time, and the current node. They stated that the state space of the proposed formulation becomes quite large making the problem intractable. They reported substantial cost savings from a computational study based on the Southeast-Michigan’s road network. To address the intractable state-space issue, Kim et al. [9] proposed state space reduction methods. A limitation of Kim et al. [8], is the modeling and partitioning of travel speeds for the determination of arc congestion states. They assume that the joint distribution of velocities from any two consecutive periods follows a single unimodal Gaussian distribution, which cannot adequately represent arc travel velocities for arcs that routinely experience multiple congestion states. Moreover, they also employ a fixed velocity threshold (*50 mph*) for all arcs and for all times in partitioning the Gaussian distribution for estimation of state-transition probabilities (i.e., transitions between congested and uncongested states). As a result, the value of real-time information is compromised rendering the loss of performance of the dynamic routing policy. Our proposed approach addresses all of these limitations.

### 3 Modeling

#### 3.1 Recurrent Congestion Modeling

Let the graph  $G=(N,A)$  denote the road network where  $N$  is the set of nodes (intersections) and  $A \subseteq N \times N$  is the set of directed arcs between nodes. For every node pair,  $n', n \in N$ , there exists an arc  $a \equiv (n, n') \in A$ , if and only if, there is a road that permits traffic flow from node  $n$  to  $n'$ . Given an origin,  $n_0$ -destination,  $n_d$  node (OD) pair, the trip planner's problem is to decide which arc to choose at each decision node such that the expected total trip travel time is minimized. We formulate this problem as a finite horizon Markov decision process (MDP), where the travel time on each arc follows a non-stationary stochastic process.

An arc is labeled as observed if its real-time traffic data (e.g., velocity) is available through the traffic information system. An observed arc can be in  $r+1 \in \mathbb{Z}^+$  different states that represent arc's traffic congestion level at a time. We begin with discussing how to determine an arc's congestion state given the real-time velocity information and defer the discussion on estimation of the congestion state parameters to Section 4. Let  $c_a^{i-1}(t)$  and  $c_a^i(t)$  for  $i=1,2,\dots,r+1$  denote the cut-off velocities used to determine the state of arc  $a$  given the velocity at time  $t$  on arc  $a$ ,  $v_a(t)$ . We further define  $s_a(t)$  as the state of arc  $a$  at time  $t$ , i.e.  $s_a(t) = \{\text{Congested at level } i\} = \{i\}$  and can be determined as:  $s_a(t) = \{i, \text{ if } c_a^{i-1}(t) \leq v_a(t) < c_a^i(t)\}$ . For instance, if there are two congestion levels (e.g.,  $r+1=2$ ), then the states will be i.e.,  $s_a(t) = \{\text{Uncongested}\} = \{0\}$  and  $s_a(t) = \{\text{Congested}\} = \{1\}$  and the travel time is normally distributed at each state.

We assume the state of an arc evolves according to a non-stationary Markov chain. In a network with all arcs observed,  $S(t)$  denotes the traffic congestion state vector for the entire network, i.e.,  $S(t) = \{s_1(t), s_2(t), \dots, s_{|A|}(t)\}$  at time  $t$ . For presentation clarity, we will suppress  $(t)$  in the notation whenever time reference is obvious from the expression. Let the state realization of  $S(t)$  be denoted by  $s(t)$ .

It is assumed that arc states are independent from each other and have the single-stage Markovian property. In order to estimate the state transitions for each arc, two consecutive periods' velocities are modeled jointly. Accordingly, the time-dependent single-period state transition probability from state  $s_a(t) = i$  to state  $s_a(t+1) = j$  is denoted with  $P\{s_a(t+1) = j | s_a(t) = i\} = \alpha_a^{ij}(t)$ . The transition probability for arc  $a$ ,  $\alpha_a^{ij}(t)$ , is estimated from the joint velocity distribution as follows:

$$\alpha_a^{ij}(t) = \frac{|c_a^{i-1}(t) \leq V_a(t) < c_a^i(t) \cap c_a^{j-1}(t+1) < V_a(t+1) < c_a^j(t+1)|}{|c_a^{i-1}(t) \leq V_a(t) < c_a^i(t)|}$$

Let  $T_a(t, t+1)$  denote the matrix of state transition probabilities from time  $t$  to time  $t+1$ , then we have  $T_a(t, t+1) = [\alpha_a^{ij}(t)]_{ij}$ . Note that the single-stage Markovian assumption is not restrictive for our approach as we could extend our methods to the multi-stage case by expanding the state space [10]. Let network be in state  $S(t)$  at time  $t$  and we want to find the probability of the network state  $S(t+\delta)$ , where  $\delta$  is a positive integer number. Given the independence assumption of arcs' congestion states, this can be formulated as follows:

$$P(S(t+\delta) | S(t)) = \prod_{a=1}^{|A|} P(s_a(t+\delta) | s_a(t)).$$

Then the congestion state transition probability matrix for each arc in  $\delta$  periods can be found by the Kolmogorov's equation:

$$T_a(t, t+\delta) = [\alpha_a^{ij}(t)]_{ij} \times [\alpha_a^{ij}(t+1)]_{ij} \times \dots \times [\alpha_a^{ij}(t+\delta)]_{ij}.$$

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

$$\delta(t, a, s_a) \sim N(\mu(t, a, s_a), \sigma^2(t, a, s_a)),$$

where  $\delta(t, a, s_a)$  is the travel time;  $\mu(t, a, s_a)$  and  $\sigma(t, a, s_a)$  are the mean and the standard deviation of the travel time on arc  $a$  at time  $t$  with congestion state  $s_a(t)$ .

### 3.2 Dynamic Routing Model with Recurrent Congestion

We assume that the objective of our dynamic routing model is to minimize the expected travel time based on real-time information where the trip originates at node  $n_0$  and concludes at node  $n_d$ . Let's assume that there is a feasible path between  $(n_0, n_d)$  where a path  $p = (n_0, \dots, n_k, \dots, n_{K-1})$  is defined as sequence of nodes such that  $a_k \equiv (n_k, n_{k+1}) \in A$ ,  $k = 0, \dots, K-1$  and  $K$  is the number of nodes on the path. We define set  $a_k \equiv (n_k, n_{k+1}) \in A$  as the *current arc* set of node  $n_k$ , and denoted with  $CrAS(n_k)$ . That is,  $CrAS(n_k) \equiv \{a_k : a_k \equiv (n_k, n_{k+1}) \in A\}$  is the set of arcs emanating from node  $n_k$ . Each node on a path is a decision stage (or epoch) at which a routing decision (which node to select next) is to be made. Let  $n_k \in N$  be the location of  $k^{\text{th}}$  decision stage,  $t_k$  is the time at  $k^{\text{th}}$  decision stage where  $t_k \in \{1, \dots, T\}$ ,  $T > t_{K-1}$ . Note that we are discretizing the planning horizon.

While optimal dynamic routing policy requires real-time consideration and projection of the traffic states of the complete network, this approach makes the state space prohibitively large. In fact, there is little value in projecting the congestion



states well ahead of the current location. This is because the projected information is not different than the long run average steady state probabilities of the arc congestion states. Hence, an efficient but practical approach would tradeoff the degree of look ahead (e.g., number of arcs to monitor) with the resulting projection accuracy and routing performance. This has been very well illustrated in Kim et al. [9]. Thus we limit our look ahead to finite number of arcs that can vary by the vehicle location on the network. The selection of the arcs to monitor would depend on factors such as arc lengths, value of real-time information, and arcs' congestion state transition characteristics. For ease of presentation and without loss of generality, we choose to monitor only two arcs ahead of the vehicle location and model the rest of the arcs' congestion states through their steady state probabilities. Accordingly, we define the following two sets for all arcs in the network.  $ScAS(a_k)$ , the successor arc set of arc  $a_k$ ,  $ScAS(a_k) \equiv \{a_{k+1} : a_{k+1} \equiv (n_{k+1}, n_{k+2}) \in A\}$ , i.e., the set of outgoing arcs from the destination node ( $n_{k+1}$ ) of arc  $a_k$ .  $PScAS(a_k)$ , the post-successor arc set of arc  $a_k$ ,  $PScAS(a_k) \equiv \{a_{k+2} : a_{k+2} \equiv (n_{k+2}, n_{k+3}) \in A\}$  i.e., the set of outgoing arcs from the destination node ( $n_{k+2}$ ) of arc  $a_{k+1}$ .

Since the total trip travel time is an additive function of the individual arc travel times on the path plus a penalty function measuring earliness/tardiness of arrival time to the destination node, the dynamic route selection problem can be modeled as a dynamic programming model. The state,  $(n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k})$ , of the system at  $k$  th decision stage is denoted by  $\Omega_k$ . This state vector is composed of the state of the vehicle and network and thus characterized by the current node ( $n_k$ ), the current node arrival time ( $t_k$ ), and  $s_{a_{k+1} \cup a_{k+2}, k}$  the congestion state of arcs  $a_{k+1} \cup a_{k+2}$  where  $\{a_{k+1} : a_{k+1} \in ScAS(a_k)\}$  and  $\{a_{k+2} : a_{k+2} \in PScAS(a_k)\}$  at  $k$  th decision stage. The action space for the state  $\Omega_k$  is the set of current arcs of node  $n_k$ ,  $CrAS(n_k)$ .

At every decision stage, the trip planner evaluates the alternative arcs from  $CrAS(n_k)$  based on the remaining expected travel time. The expected travel time at a given node with the selection of an outgoing arc is the expected arc travel time on the arc chosen and the expected travel time of the next node. Let  $\pi = \{\pi_0, \pi_1, \dots, \pi_{K-1}\}$  be the policy of the trip and is composed of policies for each of the  $K-1$  decision stages. For a given state  $\Omega_k = (n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k})$ , the policy  $\pi_k(\Omega_k)$  is a deterministic Markov policy which chooses the outgoing arc from node  $n_k$ , i.e.,  $\pi_k(\Omega_k) = a \in CrAS(n_k)$ . Therefore the expected travel cost for a given policy vector  $\pi$  is as follows:

$$F^\pi(\Omega_0) = E_{\delta_k} \left\{ \sum_{k=0}^{K-2} g(\Omega_k, \pi_k(\Omega_k), \delta_k) + \bar{g}(\Omega_{K-1}) \right\},$$

where  $\Omega_0 = (n_0, t_0, S_0)$  is the starting state of the system.  $\delta_k$  is the random travel

time at decision stage  $k$ , i.e.,  $\delta_k \equiv \delta(t_k, \pi_k(\Omega_k), s_a(t_k))$ .  $g(\Omega_k, \pi_k(\Omega_k), \delta_k)$  is cost of travel on arc  $\pi_k(\Omega_k) = a \in CrAS(n_k)$  at stage  $k$ , i.e., if travel cost is a function ( $\phi$ ) of the travel time, then  $g(\Omega_k, \pi_k(\Omega_k), \delta_k) \equiv \phi(\delta_k)$  and  $\bar{g}(\Omega_{K-1})$  is terminal cost of earliness/tardiness of arrival time to the destination node under state  $\Omega_{K-1}$ . Then the minimum expected travel time can be found by minimizing  $F(\Omega_0)$  over the policy vector  $\pi$  as follows:

$$F^*(\Omega_0) = \min_{\pi = \{\pi_0, \pi_1, \dots, \pi_{K-1}\}} F(\Omega_0).$$

The corresponding optimal policy is then  $\pi^* = \arg \min_{\pi = \{\pi_0, \pi_1, \dots, \pi_{K-1}\}} F(\Omega_0)$ . Hence, the Bellman's cost-to-go equation for the dynamic programming model can be expressed as follows [10]:

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

For a given policy  $\pi_k(\Omega_k)$ , we can re-express the cost-to-go function by writing the expectation in the following explicit form:

$$F(\Omega_k | a_k) = \sum_{\delta_k} P(\delta_k | \Omega_k, a_k) \left[ g(\Omega_k, a_k, \delta_k) + \sum_{s_{a_{k+1}, k+1}} P(s_{a_{k+1}, k+1}(t_{k+1}) | s_{a_{k+1}, k}(t_k)) \sum_{s_{a_{k+2}, k+1}} P(s_{a_{k+2}, k+1}(t_{k+1})) F(\Omega_{k+1}) \right]$$

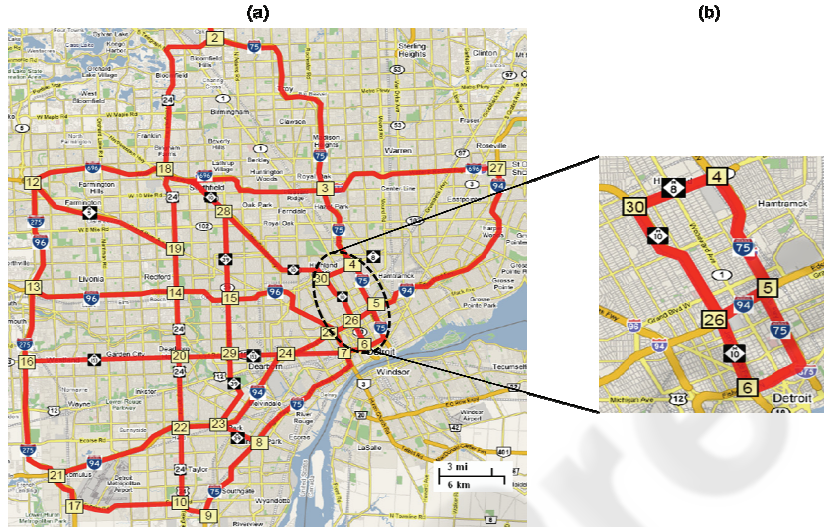
where  $P(\delta_k | \Omega_k, a_k)$  is the probability of travelling arc  $a_k$  in  $\delta_k$  periods.  $P(s_{a_{k+2}, k+1}(t_{k+1}))$  is the long run probability of arc  $a_{k+2} : a_{k+2} \in PScAS(a_k)$  being in state  $s_{a_{k+2}, k+1}$  in stage  $k+1$ . This probability can be calculated from the historical frequency of a state for a given arc and time.

We use backward dynamic programming algorithm to solve for  $F_k^*(\Omega_k)$ ,  $k = K-1, K-2, \dots, 0$ . In the backward induction, we initialize the final decision epoch such that,  $\Omega_{K-1} = (n_{K-1}, t_{K-1}, s_{K-1})$ ,  $n_{K-1}$  is destination node, and  $F_{K-1}(\Omega_{K-1}) = 0$  if  $t_{K-1} \leq T$ . Accordingly, a penalty cost is accrued whenever there is delivery tardiness, e.g.,  $t_{K-1} > T$ . Note that  $s_{K-1} = \emptyset$  since destination node current and successor arcs doesn't have value of information.

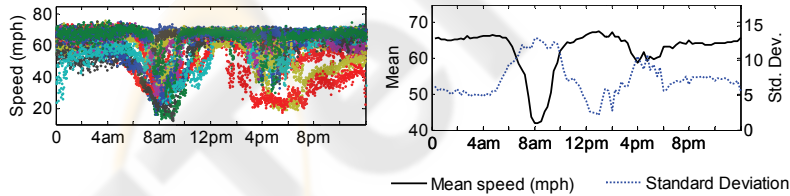
## 4 Experimental Studies

In this section we first introduce two road networks for demonstrating the performance of the proposed algorithms along with a description of their general traffic conditions. Then describe the process of how to model recurrent congestion. Finally, we report savings from employing the proposed model.

We test our procedure on a road network from South-East Michigan (Fig. 1). The sample network covers major freeways and highways in and around the Detroit metropolitan area. The network has 30 nodes and a total of 98 arcs with 43 observed arcs and 55 unobserved arcs. Real-time traffic data for the observed arcs is collected by Michigan ITS Center for 23 weekdays from January 21, 2008 to February 20, 2008 for the full 24 hours of each day at a resolution of an observation every minute. A small part of our full network, labeled *sub-network* (Fig. 1b), with 5 nodes and 6 observed arcs is used here to better illustrate the methods and results.



**Fig. 1.** (a) South-East Michigan road network considered for experimental study. (b) Sub-network from South-East Wayne County.

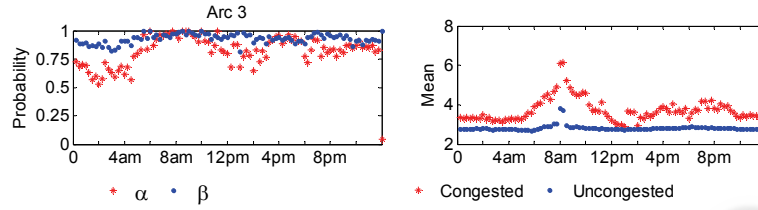


**Fig. 2.** For arc 4-to-5 (a) raw traffic speeds for 23 weekdays (b) mean (mph) and standard deviations of speeds by the time of day with 15 minute time interval resolution.

We present the speed data for arc 4-to-5 for the given days in Fig. 2.a as an example. The mean and standard deviations of speed for the arc 4-to-5 is also illustrated (Fig. 2.b). It can be seen clearly that the traffic speeds follow a stochastic non-stationary distribution that vary with the time of the day.

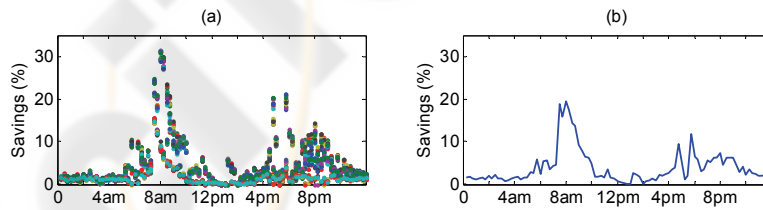
Given the traffic speed data, we employed the Gaussian Mixture Model (GMM) clustering technique to determine the number of recurrent-congestion states for each arc by time of day. In particular, we employed the greedy learning GMM clustering method of Verbeek [11] for its computational efficiency and performance. The

parameters of the traffic state joint Gaussian distributions (i.e.,  $\mu_{t,t+1}^i; \Sigma_{t,t+1}^i$ ) along with the computed cut-off speeds (if GMM yields more than one state) are employed to calculate travel time distribution parameters and the transition matrix elements as explained in section 3. In the event that two states are identified by GMM,  $\alpha_t$  denotes the probability of state transition from congested state to congested state whereas  $\beta_t$  denotes the probability of state transition from uncongested state to uncongested state. Fig. 3a plots these transition rates for the arc 4-to-5 with a 15 minute time interval resolution. The mean travel time of arc 4-to-5 for congested and uncongested traffic states are given in Fig. 3b.



**Fig. 3.** For arc 4-to-5 (a) recurrent congestion state-transition probabilities where  $\alpha$ : congested to congested transition;  $\beta$ : uncongested to uncongested transition probability; (b) mean travel time for congested and uncongested traffic states.

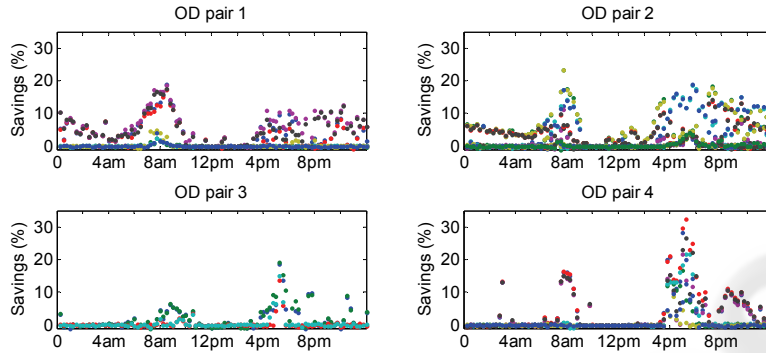
In the experiments based on the sub-network, node 4 is considered as the origin node and node 6 as the destination node of the trip. As stated earlier, we consider *node 4* as the origin node and *node 6* as the destination node of the trip. Three different path options exist (*path 1*: 4-5-6; *path 2*: 4-5-26-6; and *path 3*: 4-30-26-6). Given the historical traffic data, *path 1*: 4-5-6 dominates other paths most of the time of a day under all network states. Hence we identify *path 1* as the *baseline path* and show the savings (averaged over 10,000 runs) from using the proposed dynamic routing algorithm with regard to baseline path. Fig. 4a plots the corresponding percentage savings from employing the dynamic vehicle routing policy over the baseline path for each network traffic state combination and Fig. 4b shows the average savings (averaged across all network traffic states, treating them equally likely). It is clear that savings are higher and rather significant during peak traffic times and lower when there is not much congestion, as can be expected.



**Fig. 4.** (a) Savings for each of 32 network state combinations and (b) average savings for all state combinations during different times of the day.

Besides the sub-network (Fig. 1b), we have also randomly selected 4 other origin

and destination (OD) pairs (OD pair 1: 2-21, 2: 12-25, 3: 19-27, and 4: 23-13) to investigate the potential savings from using real-time traffic information under a dynamic routing policy. Once again, we identify the *baseline path* for each OD pair (as explained for the case of routing on the *sub-network*) and show percentage savings in mean travel times (over 10,000 runs) over the baseline paths from using the dynamic routing policy. The savings, Fig. 5, are consistent with results from the *sub-network*, further validating the sub-network results.



**Fig. 5.** Savings of dynamic policy over baseline path during the day for all starting states of given OD pairs of full network (with 15 minute time interval resolution).

## 5 Conclusions

The paper proposes practical dynamic routing models that can effectively exploit real-time traffic information from ITS regarding recurrent congestion in transportation networks. With the aid of this information and technologies, our models can help drivers avoid or mitigate trip delays by dynamically routing the vehicle from an origin to a destination in road networks. We model the problem as a non-stationary stochastic shortest path problem under recurrent congestion. We propose effective data driven methods for accurate modeling and estimation of recurrent congestion states and their state transitions.

ITS data from South-East Michigan road network, collected in collaboration with Michigan ITS Center, is used to illustrate the performance of the proposed models. Experiments show that as the uncertainty (standard deviation) in the travel time information increases, the dynamic routing policy that takes real-time traffic information into account becomes increasingly superior to static path planning methods. The savings however depend on the network states as well as the time of day. The savings are higher during peak times and lower when traffic tends to be static (especially at nights).

## Acknowledgements

This work was supported by funds from the US Department of Transportation and the Michigan Department of Transportation through the MIOH University Transportation Center.

## References

1. Hall, R.: The fastest path through a network with random time-dependent travel time. *Transportation Science* 20 (1986) 182-188
2. Fu, L.: An adaptive routing algorithm for in vehicle route guidance systems with real-time information. *Transportation Research Part B* 35B (2001) 749-765
3. Waller, S.T., Ziliaskopoulos, A.K.: On the online shortest path problem with limited arc cost dependencies. *Networks* 40 (2002) 216-227
4. Miller-Hooks, E.D., Mahmassani, H.S.: Least possible time paths in stochastic, time-varying networks. *Computers & Operations Research* 12 (1998) 1107-1125
5. Miller-Hooks, E.D., Mahmassani, H.S.: Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science* 34 (2000) 198-215
6. Gao, S., Chabini, I.: Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B* 40B (2006) 93-122
7. Psaraftis, H.N., Tsitsiklis, J.N.: Dynamic shortest paths in acyclic networks with Markovian arc costs. *Operations Research* 41 (1993) 91-101
8. Kim, S., Lewis, M.E., White III, C.C.: Optimal vehicle routing with real-time traffic information. *IEEE Transactions on Intelligent Transportation Systems* 6 (2005) 178-188
9. Kim, S., Lewis, M.E., White III, C.C.: State space reduction for non-stationary stochastic shortest path problems with real-time traffic congestion information. *IEEE Transactions on Intelligent Transportation Systems* 6 (2005) 273-284
10. Bertsekas, D.P.: *Dynamic Programming and Optimal Control*. Athena Scientific (2001)
11. Verbeek, J.J., Vlassis, N., Kröse, B.: Efficient Greedy Learning of Gaussian Mixture Models. *Neural Computation* 5 (2003) 469-485