Dynamic routing under recurrent and non-recurrent congestion using real-time ITS information

Ali R. Güner, Alper Murat*, Ratna Babu Chinnam

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

A R T I C L E   I N F O

Available online 5 May 2011

Keywords:
JIT logistics
Transportation
Congestion
Incidents
Dynamic routing
ITS

A B S T R A C T

In just-in-time (JIT) manufacturing environments, on-time delivery is a key performance measure for dispatching and routing of freight vehicles. Growing travel time delays and variability, attributable to increasing congestion in transportation networks, are greatly impacting the efficiency of JIT logistics operations. Recurrent and non-recurrent congestion are the two primary reasons for delivery delay and variability. Over 50% of all travel time delays are attributable to non-recurrent congestion sources such as incidents. Despite its importance, state-of-the-art dynamic routing algorithms assume away the effect of these incidents on travel time. In this study, we propose a stochastic dynamic programming formulation for dynamic routing of vehicles in non-stationary stochastic networks subject to both recurrent and non-recurrent congestion. We also propose alternative models to estimate incident induced delays that can be integrated with dynamic routing algorithms. Proposed dynamic routing models exploit real-time traffic information regarding speeds and incidents from Intelligent Transportation System (ITS) sources to improve delivery performance. Results are very promising when the algorithms are tested in a simulated network of South-East Michigan freeways using historical data from the MITS Center and Traffic.com.

© 2011 Elsevier Ltd. All rights reserved.

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. The requirements have even spread to the supply chains' service sectors with the adoption of cross docking, merge-in-transit, and e-fulfillment, especially in developed countries with keen concern in process improvement [1]. For example, in Osaka and Kobe, Japan, as early as 1997, 52% (by weight) of cargo deliveries and 45% of cargo pickups had designated time windows or specified arrival times [2]. These requirements have now become the norm in the US as well. For example, many automotive final assembly plants in South-East Michigan receive nearly 80% of all assembly parts on a JIT basis (involving 5–6 deliveries/day for each part with no more than three hours of inventory at the plant). However, road transportation networks are experiencing ever growing congestion, which greatly hinders all travel and certainly the freight delivery performance. The cost of this congestion is growing rapidly, reaching $78B by 2005 (from $20B in 1985) just in the US large metropolitan areas alone [3]. This congestion is forcing logistics solution providers to add significant travel time buffers to improve on-time delivery performance, causing idle vehicles due to early arrivals. Fig. 1, for example, illustrates the magnitude of these buffers for 2003 in the automotive industry heavy Detroit Metro area, reaching over 70% during peak congestion periods of the day to achieve 95% on-time delivery performance [4]. Given that automotive plants are heavily relying on JIT deliveries, this is increasingly forcing the automotive original equipment manufacturers (OEMs) and others to carry increased levels of safety inventory to cope with the risk of late deliveries.

The average trip travel time varies by the time of day. 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. Incidents, such as accidents, vehicle breakdowns, bad weather, work zones, lane closures, special events, etc. are other important sources of traffic congestion. This type of congestion is labeled ‘non-recurrent’ congestion in that its location and severity is unpredictable. The Texas Transportation Institute [5] reports that over 50% of all travel time delays are attributable to the non-recurrent congestion. Despite its importance, current state-of-the-art dynamic routing algorithms assume away the effect of these incidents on travel time.
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), as illustrated in Fig. 1. Intelligent Traffic Systems (ITS), run by state agencies (e.g., the Michigan Intelligent Transportation Systems (MITS) Center in South-East Michigan) and/or the private sector (e.g., Traffic.com operating in many states), are providing real-time traffic data (e.g., lane speeds and volumes) in many urban areas. These traffic monitoring systems are also beginning to provide real-time information regarding traffic incidents and their severity. 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 both recurrent and non-recurrent congestion. We limit the scope to routing a vehicle from an origin point (say depot or warehouse) to a destination point.

Our problem setting is the non-stationary stochastic shortest path problem with both recurrent and non-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 the static ‘paths’ are provably sub-optimal for this problem. 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. As for non-recurrent congestion attributable to incidents, we estimate the incident-induced arc travel time delay using a stochastic queueing model.

Our contribution is two-fold: (1) methods for accurate and efficient representation of recurrent congestion, in particular, identification of multiple congestion states and their transition patterns. (2) Integrated modeling and treatment of recurrent and non-recurrent congestion for vehicle routing and demonstrating the need and value of such integration.

The rest of the paper is organized as follows. Survey of relevant literature is given in Section 2. Modeling recurrent and non-recurrent congestion is presented in Section 3. Section 4 proposes a dynamic vehicle routing model under recurrent and non-recurrent congestion using real-time data. Section 5 presents the results of a real-world experimental study. Finally, Section 6 offers some concluding remarks and proposes avenues for future research.

2. Literature survey

In the classical deterministic shortest path (SP) problem, the cost of traversing an arc is deterministic and independent on the arrival time to the arc. The stochastic SP problem (S-SP) is a direct extension of this deterministic counterpart where the arc costs follow a known probability distribution. In S-SP, there are multiple potential objectives, and the two most common ones are the minimization of the total expected cost and maximization of the probability of being lowest cost [6]. To find the path with minimum total expected cost, Frank [7] suggested replacing arc costs with their expected values and subsequently solving as a deterministic SP. Loui [8] showed that this approach could lead to sub-optimal paths and proposed using utility functions instead of the expected arc costs. Eiger et al. [9] showed that Dijkstra’s algorithm [10] can be used when the utility functions are linear or exponential.

Stochastic SP problems are referred as stochastic time-dependent shortest path problems (STD-SP) when arc costs are time-dependent. Hall [11] first studied the STD-SP problems and showed that the optimal solution has to be an ‘adaptive decision policy’ (ADP) rather than a single path. In an ADP, the node to visit next depends on both the node and the time of arrival at that node, and therefore the standard SP algorithms cannot be used. Hall [11] employed the dynamic programming (DP) approach to derive the optimal policy. Bertsekas and Tsitsiklis [12] proved the existence of optimal policies for STD-SP. Later, Fu and Rillett [13] modified the method of Hall [11] for problems where arc costs as continuous random variables. They showed the computational intractability of the problem based on the mean-variance relationship between the travel time of a given path and the dynamic and stochastic travel times of the individual arcs. They also proposed a heuristic in recognition of this intractability. Bander and White [14] modeled a heuristic search algorithm AO* for the problem and demonstrated significant computational advantages over DP, when there exists known strong lower bounds on the total expected travel cost between any node and the destination node. Fu [15] 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 [16] suggested polynomial algorithms to find optimal policies for stochastic shortest path problems with one-step arc and limited temporal dependencies. Gao and Chabini [17] designed an ADP algorithm and proposed efficient approximations to time and arc dependent stochastic networks. An alternative routing solution to the ADP is a single path satisfying an optimality criterion. For identifying paths with the least expected travel (LET) time, Miller-Hooks and Mahmassani [18] proposed a modified label-correcting algorithm. Miller-Hooks and Mahmassani [19] extends [18] 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 Waller and Ziliaskopoulos [16] and Gao and Chabini [17]. In most urban transportation networks, however, the change in the cost of traversing an arc over-time is stochastic and there are very few studies addressing this issue. Most of these studies model this stochastic temporal dependence through Markov chain modeling and propose using the real-time information available through ITS systems for observing Markov states. In addition, all of these studies assume that recourse actions are possible such that the vehicle’s path can be re-adjusted based on newly acquired congestion information. Accordingly, they identify optimal ADPs. Psarafitis and Tsitsiklis [20] is the first study to consider stochastic temporal dependence of arc costs and to suggest using online.
information en-route. They considered an acyclic network where the cost of outgoing arcs of a node is a function of the environment state of that node and the state changes according to a Markovian process. They assumed that the arc’s state is learned only when the vehicle arrives at the source node and the state of nodes are independent. They also proposed a DP procedure to solve the problem. Polychronopoulos and Tsiakis [21] consider a problem when recourse is possible in a network with dependent undirected arcs and the arc costs are time independent. They proposed a DP algorithm to solve the problem and discussed some non-optimal but easily computable heuristics. Azaron and Kianfar [22] extended [20] by evolving the states of current node as well as its forward nodes with independent continuous-time semi-Markov processes for ship routing problem in a stochastic but time invariant network. Kim et al. [23] studied a similar problem as in [20] 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 South-East Michigan’s road network. To address the intractable state-space issue, Kim et al. [24] proposed state space reduction methods. A limitation of Kim et al. [24], 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 multi-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.

2.1. Non-recurrent incidents and incident clearance

All of the shortest-path studies reviewed above consider stochastic arc costs that are mostly attributable to recurrent congestion. However, as stated earlier, over 50% of all traffic congestion is attributable to non-recurrent incidents and has to be accounted for dynamic routing. Incident-induced delay time estimation models are widely studied in the transportation literature. These models can be categorized into three groups based on their approaches: shockwave theory [25–27], queuing theory [28–33], and statistical (regression) models [34–36]. All of these modeling approaches have certain requirements such as loop-sensor data or assumptions regarding traffic/vehicle behavior. For instance, the shockwave theory based models require extensive loop sensor data for accurate positioning and progression of shockwave. Both queuing and shockwave theory based models require assumptions about the vehicle arrival process. Regression models, as empirical methods, cannot handle missing data without compromising on accuracy.

In all these three modeling methods, the delay due to incident is a function of incident duration. Thus, the correct estimation of incident duration is fundamental and there are various distributions suggested. Gaver [37] derived probability distributions of delay under flow stopping. Truck-involved incident duration is studied by Golob et al. [38] and employs lognormal distribution. Analysis of variance is examined by Giuliano [39] and a truncated regression model to estimate incident duration is proposed by Khattak et al. [40] for incident durations in Chicago area. Gamma and exponential distributions are also suggested as good representations of incident duration distribution [41]. Since the likelihood of ending an incident is related to how long it has lasted, hazard-based models are also suggested extensively. An overview of duration models applications is presented by Hensher and Mannering [42]. Nam and Mannering [43] applied hazard-based duration models to model distribution of detect/report, respond and clear duration of incidents. Using the empirical data of two years from the state of Washington, they showed that detect/report and respond times are Weibull distributed and the clearance duration is log-logistic distributed.

Modeling incident delay in conjunction with vehicle routing is in its nascence. Ferris and Ruszczyński [44] present a problem in which arcs with incidents fail and become permanently unavailable. They model the problem as an infinite-horizon Markov decision process. Thomas and White [45] consider the incident clearance process and adopt the models in Kim et al. [23] for routing under non-recurrent congestion. They model the incident delay using a multiplicative model and the incident clearance time as a non-stationary Markov chain, with transition probabilities following a Weibull distribution with an increasing instantaneous clearance rate. To model incident-induced delay, they multiply the incident arc’s cost by a constant and time-invariant scalar. However, they do not account for recurrent congestion and assume arc costs are time-invariant and deterministic. In our approach, we address these limitations by joint consideration of recurrent and non-recurrent congestion as well as more appropriate representation of incident-induced delay and clearance.

3. Modeling recurrent and non-recurrent congestion

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 \( a = (n, n') \in A \), if and only if, there is a road that permits traffic flow from node \( n \) to \( n' \). Given an origin–destination (OD) node 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 denote the origin and destination nodes with \( n_o \) and \( n_p \), respectively. 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, \( a = (n, n') \in A \) is labeled as observed if its real-time traffic data (e.g., velocity) is available through the traffic information system. An observed arc’s traffic congestion can be in \( r + 1 \in \mathbb{Z}^+ \) different states at time \( t \). These states represent arc’s congestion level and are associated with the real-time traffic velocity on the arc. 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 5. Let \( c_i^t(1) \) and \( c_i^t(2) \) for \( i = 1, 2, \ldots, r + 1 \) 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 \( s_{i+1}^t(1) \) as the \( i \)th traffic congestion state of arc \( a \) at time \( t \), i.e., \( s_i^t(1) = \{1\} \) and \( s_i^t(2) = \{\text{Congested at level } r \} \). For instance, if there are two congestion levels (e.g., \( r + 1 = 2 \), then there will be one congested state and the other will be uncongested state, i.e., \( s_{i+1}^t(2) = \{\text{Uncongested}\} \) or \( (0) \) and \( s_{i+1}^t(1) = \{\text{Congested}\} = \{1\} \). Congestion state, \( s_{i+1}^t(2) \) of the arc \( a \) at time \( t \) can then be determined as \( s_{i+1}^t(2) = \{i, c_i^t(1) \leq v_a(t) < c_i^t(2) \} \).

We assume the congestion state of an arc evolves according to a non-stationary Markov chain and the travel time is normally distributed at each state. In a network with all arcs observed, \( S(t) \)
denotes the traffic congestion state vector for the entire network, i.e., \( S(t) = [s_1(t), s_2(t), \ldots, s_9(t)] \) 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 traffic congestion 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 \( i \) to state \( j \) at time \( t+1 \) is denoted with \( P_{ij}(s_{j}(t+1) = j | s_i(t) = i) = \varphi_{ij}(t) \). The transition probability for arc \( a \), \( \varphi_{a}(t) \), is estimated from the joint velocity distribution as follows:

\[
\varphi_{a}(t) = \frac{\varphi_{i}^{a-1}(t) \leq V_{i}(t) < \varphi_{i}^{a+1}(t+1) \leq V_{a}(t+1) < \varphi_{i}^{a+1}(t+1)}{\varphi_{i}^{a-1}(t) \leq V_{i}(t) < \varphi_{i}^{a+1}(t+1)}
\]

Let \( T_a(t+1) \) denote the matrix of state transition probabilities from time \( t \) to time \( t+1 \), then we have \( T_a(t+1) = [\varphi_{ij}(t)] \). We further assume that arc \( a \)’s congestion state is independent of other arcs’ states, i.e. \( P(s_{j}(t+1) | s_{i}(t+1), s_{2}(t)) = P(s_{j}(t+1) | s_{i}(t)) = \varphi_{ij}(t) \) for \( \forall r \in A \). 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 [46]. 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:

\[
P(S(t+\delta) | S(t)) = \prod_{a \in A} P(s_{i}(t+\delta) | s_{i}(t))
\]

Then the congestion state transition probability matrix for each arc in \( \delta \) periods can be found by the Kolmogorov’s equation [47]:

\[
T_a(t+\delta) = [\varphi_{ij}(t)] \times [\varphi_{ij}(t+1)] \times \cdots \times [\varphi_{ij}(t+\delta)]
\]

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:

\[
\hat{\delta}(t,a,s_a) \sim N[\mu(t,a,s_a),\sigma^2(t,a,s_a)],
\]

where \( \hat{\delta}(t,a,s_a) \) is the travel time on arc \( a \) at time \( t \) with congestion state \( s_a \); \( \mu(t,a,s_a) \) and \( \sigma(t,a,s_a) \) are the mean and standard deviation of the travel time on arc \( a \) at time \( t \) with congestion state \( s_a \). For the clarity of notation, we hereafter suppress the arc label from the parameter space wherever it is obvious, i.e., \( \hat{\delta}(t,a,s_a) \) will be referred as \( \hat{\delta}(t,s_a) \).

We assume that objective of dynamic routing is to minimize the expected travel time based on the real-time information. The nodes (intersections) of the network represent decision points where a routing decision can be made. Since our algorithm is also applicable for a network with incidents, in the next section we present our incident modeling approach, and then integrate the recurrent congestion and incident models.

### 3.2. Incident modeling

In this section, we develop incident models which measure the incident clearance time and the delay experienced as a result of incident. In Section 4, we integrate recurrent congestion and incident models with the dynamic routing model.

#### 3.2.1. Estimating incident duration

The incident duration is defined as the total of detection/reporting, response, and clearance times. Due to the nature of most incident response mechanisms, the longer the incident has not been cleared, the more likely that it will be cleared in the next period. For example, the probability of an incident being cleared in the 15th min, given that it has lasted 14 min, is greater than the probability of it being cleared in the 14th min given that it has lasted 13 min. This is because it is more likely that someone has already reported the incident and an incident response team is either on the way or has already responded. Let \( t \) be the time to clear the incident. Then, we have the increasing hazard rate property, e.g., \( \lambda(t+1) > \lambda(t) \), where \( \lambda(t) = \lambda(1-F(t)) \) is the hazard rate of incident clearance in duration \( t \), and \( F(t) \) and \( f(t) \) are the density and cumulative density functions of the clearance duration, respectively. We choose the Weibull distribution with increasing hazard rate to model the incident clearance duration.

Whenever there is an incident on an arc in the network, we assume that its starting time \( t_{0e} \), current status (i.e., cleared/not cleared), expected duration \( \mu \), and standard deviation \( \sigma \) are available through ITS’s incident management and incident database systems. Hence, we can estimate the parameters of the Weibull distribution \( (\lambda, \mu, \sigma) \) of the incident clearance duration [47]. Furthermore, if an incident occurs en-route, we may simply re-optimize the routing policy by assuming that the new origin node is the node that the driver is at or arrives next.

#### 3.2.2. Estimating incident-induced delay

Our incident delay model is based on [29]. Here incident-induced delay function, \( \Theta(\cdot) \), is based on the incident duration \( \phi \), road non-incident capacity denoted with \( c \) (vehicle per hour, or vph in short), road capacity during the incident denoted with \( c'(vph) \), and arrival rate of vehicles to the incident arc denoted with \( q(vph) \). Given these parameters for an incident started at \( t_{0e} \), the vehicle arriving to the incident arc at time \( t \) experiences the following expected incident-induced delay:

\[
E(\Theta(\cdot)) = \frac{\text{mean}(\varphi_{inc})}{c} (D_{12} + D_{13} + \text{PDM}),
\]

where \( D_{12} = \int_{t_{0e}}^{t} \varphi_{inc}(x) \ dx \), \( D_{13} = (c-p)/c-q(t-t_{0e}) \), \( D_{2} = (q/p) (t-t_{inc}) \), \( \text{PDM} = \frac{1}{P_{1}+P_{2}} \), \( P_{1} = \int_{t_{0e}}^{t} \varphi_{inc}(x) \ dx \), and \( P_{2} = \int_{t_{inc}}^{t} \varphi_{inc}(x) \ dx \).

In order to track the amount of time that each arc has spent in the incident state, we define an incident duration vector defined over all the arcs, \( \varphi(\cdot) \), i.e. \( \varphi(t) = [\varphi_{1}(t), \varphi_{2}(t), \ldots, \varphi_{9}(t)] \). Note that if an arc \( a \) is not an incident arc, then \( \varphi_{a}(t) = 0 \), otherwise \( \varphi_{a}(t) = t-t_{0e}(a) \) and \( 0 < \varphi_{a}(t) < \infty \), where \( t_{0e}(a) \) is the incident onset time on arc \( a \). For presentation clarity, we will hereafter omit the arc reference from the incident onset time, i.e. \( t_{e} = t_{0e}(a) \), whenever incident arc reference is obvious.

The incident delay model is an additive model, in that, \( \Theta(\cdot) \) represents the delay time by which the arc travel time under similar conditions (congestion state and the time) will be increased by a duration amounting to the incident induced delay. Specifically, given the arc travel time without the incident, \( \delta_{0}(t,s,i) = 0 \), and the incident parameters, \( (\varphi,c,p,q) \), we can express the arc travel time with incident as \( \delta_{i}(t,s,i) = \delta_{0}(t,s,i) + \Theta_{a}(\varphi,c,p,q,i:i = t-t_{0e}) \).

We make the following assumptions for the incident delay function:

**Assumption 1.** Incident delay is only experienced on the incident arc (no propagation of the incident delay effect in the remainder of the network).
Assumption 2. Incident delay function is additive which amplifies the incumbent arc travel time.

Assumption 3. Incident delay function, $\Theta(\cdot)$, is such that the total delay associated by deciding to wait at a node (e.g., waiting time plus the incident delay), is not less than the case without waiting.

In practice, the incident effect propagates in the network in the form of a shockwave after a certain duration following the incident. Since our goal is to investigate the impact of incidents on the travel time, we choose to focus on the most important ingredient, namely the incident-induced delay on the incident arc. Hence, Assumption 1 is acceptable under certain scenarios. One scenario is where the incident duration is not long enough that vehicles divert to alternative arcs or the capacity of alternative arcs is sufficiently large to accommodate the diversion without any change in their congestion state. The additive model assumption (Assumption 2) is appropriate since the travel time delay of a particular incident depends on both the incident characteristics and the incumbent travel time on the arc. Assumption 3 is consistent with our network and travel time assumptions where we assume that waiting at a node (or on an arc) is not permitted and/or does not provide travel time savings (first-in-first-out property). The following lemma provides a requirement for the incident model parameters such that the Assumption 3 holds.

**Lemma 1.** The incident-induced delay parameters (c.q., satisfying the following condition for the minimal waiting time of $\Lambda$ (smallest discrete time interval), ensures that waiting at the incident node does not reduce the expected travel time

$$\mu_q(L_k + \Delta, s) - \mu_q(L_k, s) \geq -\frac{q}{\Delta},$$

**Proof.** Proof of this lemma is provided in Appendix. □

## 4. Dynamic routing model with recurrent and non-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 $n_T$. Let's assume that there is a feasible path between $(n_0, n_T)$ where a path $p = (n_0, n_1, \ldots, n_{T-1})$ is defined as sequence of nodes such that $a_k = (n_{k-1}, n_k) \in A$, $k = 0, \ldots, K - 1$ and $K$ is the number of nodes on the path. We define set $a_k = (n_{k-1}, n_k) \in A$ as the current arc set of node $n_k$, and denote with $\text{CRA}(n_k)$. That is, $\text{CRA}(n_k) = \{a_k = (n_{k-1}, n_k) : n_k \in A\}$ 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 $k$th decision stage, $t_k$ is the time at $k$th decision stage where $t_k \in [1, \ldots, T]$ and $T > t_k$. Note that we are discretizing the planning horizon. We next define our look ahead policy for projecting the congestion states in the network. 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. [24]. 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. $\text{CRA}(n_k)$, the successor arc set of arc $a_k$, $\text{CRA}(n_k) = \{a_k = (n_{k-1}, n_k) : (n_{k-1}, n_k) \in A\}$, i.e., the set of outgoing arcs from the destination node $(n_{k-1})$ of arc $a_k$. $\text{PSA}(n_k)$, the post-successor arc set of arc $a_k$, $\text{PSA}(n_k) = \{a_k = (n_{k-1}, n_k) : (n_{k-1}, n_k) \in A\}$ i.e., the set of set of outgoing arcs from the destination node $(n_{k-1})$ of arc $a_k$. 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 final destination, the dynamic route selection problem can be modeled as a dynamic programming model. The state of the system at $k$th decision stage is denoted by $\Omega(n_k, t_k, \text{PSA}(n_k), \text{CRA}(n_k))$. 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 $\text{PSA}(n_k)$, the congestion state of arcs $a_k$ where $a_k = 1$, $\text{CRA}(n_k)$, i.e., the set of current arcs of node $n_k$, denoted with $\text{CRA}(n_k)$. Therefore the expected travel cost for a given policy vector $\pi = (\pi_0, \pi_1, \ldots, \pi_{T-1})$ is as follows:

$$F_0(n_0, t_0, S_0, \pi_0) = E\left\{\sum_{k=0}^{T-1} g(s) \cdot \pi_k(S_0, \pi_k)\right\},$$

where $(n_0, t_0, S_0, \pi_0)$ is the starting state of the system. $\delta_k$ is the random travel time at decision stage $k$, i.e., $\delta_k = \Theta(t_k, \pi_k(S_0, \pi_k), g(s)) + \Theta(\theta_k, \pi_k(S_0, \pi_k), g(s)) + \Theta(\theta_k, \pi_k(S_0, \pi_k), g(s))$, i.e., the incident delay of an arc without incident, $g(s)\delta_k$ is cost of travel on arc $a = \pi_k(S_0, \pi_k)$ at decision stage $k$, i.e., if travel cost is a function of the travel time, then $g(s)\delta_k = g(s)\delta_k$. Then the minimum expected travel time can be found by minimizing $F(n_0, t_0, S_0, \pi_0)$ over the policy vector $\pi = (\pi_0, \pi_1, \ldots, \pi_{T-1})$ as follows:

$$F^*(n_0, t_0, S_0) = \min_{\pi_0, \ldots, \pi_{T-1}} F(n_0, t_0, S_0, \pi_0).$$

The corresponding optimal policy is then

$$\pi^* = \arg\min_{\pi_0, \ldots, \pi_{T-1}} F(n_0, t_0, S_0, \pi_0).$$

Hence, the Bellman's cost-to-go equation for the dynamic programming model can be expressed as follows [46]:

$$F^*(\Omega_k) = \min_{\pi_k} \{g(\Omega_k, \pi_k(\Omega_{k+1}), \delta_{k+1}) + F^*(\Omega_{k+1})\}.$$
ingly, a penalty cost is accrued whenever there is delivery tardiness, for a network with multiple origin–destination (OD) pairs. Section 5.4 presents the experimental setup that involves an accident and reports results and savings from employing the proposed dynamic routing model under both recurrent and non-recurrent congestion. Section 5.5 discusses the computational performance of the proposed approach and presents implementation recommendations under different congestion scenarios.

5. Experimental studies

This section demonstrates the performance of the proposed algorithm on a network from South-East Michigan with real-time traffic data from the Michigan Intelligent Transportation Systems (MITS) Center. MITS center is the hub of ITS technology applications at the Michigan Department of Transportation (MDOT) and oversees a traffic monitoring system composed of 180 freeway miles instrumented with 180 Closed Circuit TV Cameras, Dynamic Message Signs, and 2260 Inductive Loops. The methods also utilize real-time and archived data from Traffic.com, a private company that provides traffic information services in several states and also operates additional sensors and traffic monitoring devices in Michigan. Traffic.com also provides information regarding incidents causing non-recurrent congestion (e.g., incident location, type, severity, and times of incident occurrence and clearance). We implemented all our algorithms and methods in Matlab R2010b and executed on a machine (with Intel Core 2 2.13 GHz speed processor and 2 GB RAM) running Microsoft Windows 7 32-bit operating system.

Our experimental study is outlined as follows: Section 5.1 introduces two road networks from South-East Michigan used for demonstrating the performance of the proposed algorithms along with a description of their general traffic conditions. Section 5.2 describes the process and the results from modeling of recurrent congestion for the networks. Section 5.3 reports savings from employing the proposed dynamic routing model under recurrent congestion for a network with multiple origin–destination (OD) pairs. Section 5.4 presents the experimental setup that involves an accident and reports results and savings from employing the proposed dynamic routing model under both recurrent and non-recurrent congestion. Section 5.5 discusses the computational performance of the proposed approach and presents implementation recommendations under different congestion scenarios.

5.1. Sample networks and traffic data

This section introduces the road networks from South-East Michigan used for demonstrating the performance of the proposed algorithms along with a description of their general traffic conditions. As illustrated in Fig. 2, the sample network covers South-East Michigan 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 (with real-time ITS information from MITS Center) and 55 unobserved arcs. Real-time traffic data for the observed arcs is collected from MDOT Center for 23 weekdays from January 21, 2008 to February 20, 2008 for the full 24 h of each day at a resolution of an observation every minute. The raw traffic speed data from MITS Center is cleaned with a series of procedures from Texas Transportation Institute and Cambridge Systematics [4] to improve quality and reduce data errors.

A small part of our full network, labeled sub-network, is used here to better illustrate the methods and results (Fig. 2b). The sub-network has 5 nodes and 6 observed arcs, with more details provided in Table 1. 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. Given the OD pair, we present the speed data for the six different arcs of the sub-network in Fig. 3.1. It can be seen clearly that the traffic speeds follow a stochastic non-stationary distribution that vary with the time of the day. The mean speeds and standard deviations for these same arcs are shown in Fig. 4, clearly revealing the non-stationary nature of traffic.

\[ P(\delta_k|n_K, s, t, I) \] is the probability of traveling arc \( a_k \) in \( \delta_k \) periods. \( P(\delta_{k+1}|a_k, s, t, I) \) is the long run probability of arc \( a_{k+2} \), \( a_{k+2} \in \text{PScAS}(a_k) \) being in state \( s_{k+1} \) in stage \( k+1 \). This probability can be calculated from the historical state frequency of a given arc and time.

We use backward dynamic programming algorithm to solve for \( P_k(x_k, k=K-1, K-2, ..., 0) \). In the backward induction, we initialize the final decision epoch such that, \( E_{k-1} = E(n_{k-1}, I_{k-1}) \), \( n_{k-1} \) is destination node, and \( E_k(x_k, I_{k-1})=0 \) if \( I_{k-1} \leq T \). Accordingly, a penalty cost is accrued whenever there is delivery tardiness, e.g., \( I_{k-1} > T \).

Fig. 2. (a) South-East Michigan road network considered for experimental study. (b) Sub-network from South-East Wayne County.
5.2. Recurrent congestion modeling

The proposed dynamic routing algorithm calls for identification of different congestion states and estimation of their state transition rates as well as arc traverse times by time of day. Given the traffic speed data from MITS Center, 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 et al. [48] for its computational efficiency and performance. To estimate the number of congestion

<table>
<thead>
<tr>
<th>Arc ID</th>
<th>Freeway</th>
<th>Length (miles)</th>
<th>FROM Node # Description (exit #)</th>
<th>TO Node # Description (exit #)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I-94</td>
<td>1.32</td>
<td>5</td>
<td>216</td>
</tr>
<tr>
<td>2</td>
<td>M-8</td>
<td>1.75</td>
<td>4</td>
<td>56A (I-75)</td>
</tr>
<tr>
<td>3</td>
<td>I-75</td>
<td>3.13</td>
<td>4</td>
<td>56A</td>
</tr>
<tr>
<td>4</td>
<td>I-75</td>
<td>2.81</td>
<td>5</td>
<td>53B</td>
</tr>
<tr>
<td>5</td>
<td>M-10</td>
<td>3.26</td>
<td>30</td>
<td>7C (M-10)</td>
</tr>
<tr>
<td>6</td>
<td>M-10</td>
<td>1.42</td>
<td>26</td>
<td>48</td>
</tr>
</tbody>
</table>

Fig. 3. Raw traffic speeds for arcs on sub-network (mph) at different times of the day (data: weekday traffic from January 21 to February 20. Each color represents a distinct day of 23 days). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Fig. 4. Traffic mean speeds (mph) and standard deviations by time of the day for arcs on sub-network (15 min time interval resolution).
states, traffic speed data from every pair of two consecutive time periods, $t$ and $t+1$, are clustered and modeled using a bi-variate joint Gaussian distribution $(\mu_{i,t+1}, \Sigma_{i,t+1})$, where $i$ denotes the $i$th cluster. The Gaussian distribution assumption has been employed by others in the literature (see Kim et al. [23]). The clusters are ordered by their means and the densities of their projections onto the two axes are employed to identify the congestion state speed from GMM and (c) partitioned traffic states based on projections. A.R. Güner et al. / Computers & Operations Research 39 (2012) 358–373 365

Fig. 7 plots these transition rates for the different arcs of the sub-network. Note that the state transitions to same states (i.e., congested to congested or uncongested to uncongested) are more likely during peak demand time periods, which increase the value of the congestion state information, and is the case in practice. For the sub-network, the mean and standard deviation of arc travel times are illustrated in Figs. 8 and 9, respectively, by traffic state and time of day.

5.3. Results from modeling recurrent congestion

This section highlights the potential savings from explicit modeling of recurrent congestion during dynamic vehicle routing. First, we discuss the results for routing on the sub-network. 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). Note that our aim is not to identify an optimal path, rather, to identify the best policy based on the time of the day, location of the vehicle, and the traffic state of the network (for paths can be sub-optimal under non-stationary networks). However, in practice, almost all commercial logistics software aim to identify a robust (static) path that is best on the average. In this context, given the traffic flow histories for the arcs of the sub-network, path 1: 4-5-6 would be most robust, for it dominates other paths most of the day under all network states. Hence, we identify path 1 as the baseline path and show the savings from using the proposed dynamic routing algorithm with regard to baseline path. Since we limit the traffic state look ahead to only successor and post-successor arcs, there are 5 arc states to be considered at the starting node of the trip. This implies that there are $2^5 = 32$ starting network traffic state combinations. We simulated the trip 10,000 times for each of these starting network traffic state combinations throughout the day for 15 min interval starting times (yielding $(24 \times 60)/15 = 96$ trip start times). Fig. 10a plots the mean baseline path
travel times over 10,000 simulation runs for every combination of the sub-network traffic state (all 32 of them) and Fig. 10b plots the mean travel times for the dynamic policy.

Fig. 11a plots the corresponding percentage savings from employing the dynamic vehicle routing policy over the baseline path for each network traffic state combination and Fig. 11b 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.

Besides the sub-network (Fig. 2b), as listed in Table 2, we have also identified 5 other origin and destination (OD) pairs in South-East Michigan road network (Fig. 2a) to investigate the potential savings from using real-time traffic information under a dynamic routing policy. Unlike the sub-network, these OD pairs have both observed and unobserved arcs and each OD pair has several alternative paths from origin node to destination node.

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. Fig. 12 plots the percentage savings for each network traffic state combination and Fig. 13 shows the average savings (averaged across all network traffic states, treating them equally likely). The
savings are consistent with results from the sub-network, somewhat validating the sub-network results, with higher savings once again during peak traffic times.

5.4. Impact of modeling incidents

This section highlights the potential savings from explicit modeling of non-recurrent congestion along with modeling of recurrent congestion during dynamic vehicle routing. As for the setting, we focus on the sub-network (Fig. 2b). We derive the dynamic routing policies in two ways. Initially, the dynamic policy

Table 2

<table>
<thead>
<tr>
<th>OD pair</th>
<th>Origin</th>
<th>Destination</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2</td>
<td>I-75 and US-24</td>
</tr>
<tr>
<td>2</td>
<td>12</td>
<td>I-96 and I-696</td>
</tr>
<tr>
<td>3</td>
<td>19</td>
<td>M-5 and US-24</td>
</tr>
<tr>
<td>4</td>
<td>23</td>
<td>I-94 and M-39</td>
</tr>
<tr>
<td>5</td>
<td>3</td>
<td>I-75 and I-696</td>
</tr>
</tbody>
</table>
does not account for non-recurrent congestion even though there is an incident in the network. Later, we allow the dynamic policy to explicitly account for non-recurrent congestion information to generate the optimal policy. We show the results for 6 starting times during the day (to study the impact of non-stationary traffic on savings): 6:30 am, 9:00 am, 10:30 am, 4:00 pm, 5:30 pm, and 7:00 pm. To achieve a good comparison, we set all parameters of the incident to be the same for all starting times. We create an incident on either arc 3, or 4, or 6 with duration mean of 10 min and standard deviation of 5 min, following a Weibull distribution (scale parameter of 11.3 and a shape parameter of 2.1). We assume that all the arcs of the sub-network have a capacity of 1800 vehicles per hour (vph) under normal conditions and that the incident reduces their capacity to 1080 vph. Also, we assume inflow traffic arrival rate for each arc to be 1500 vph during these operation times. We have also validated the assumption of no node waiting for incident arcs using Lemma 1.

Fig. 12. Savings of dynamic policy over baseline path during the day for all starting states of given OD pairs (with 15 min time interval resolution).

Fig. 13. Average savings of dynamic policy over baseline path during the day for all starting states of given OD pairs (with 15 min time interval resolution).
The savings for the first scenario are presented in Fig. 14a. Since arc 3 is close to the origin node, the effect of incident is generally high which leads to greater savings. Arc 4 is a downstream arc (i.e., it is not connected to the origin node), thus the incident is partially cleared by the time the vehicle reaches there. Subsequently, the impact of the incident on arc travel time and the savings are lesser. Arc 6 is also a downstream arc but the dynamic policy (without taking into account the non-recurrent congestion) sometimes chooses this arc, thus there are savings associated with explicit modeling of non-recurrent congestion. Due to space constraints, we are not presenting results from incidents on other arcs. The results for other arcs vary for similar reasons. The results for the second scenario (e.g., 20 min into the incident) are presented in Fig. 14b. The savings for this scenario are less than the first scenario since the incident has partially or fully cleared by the time the vehicle reaches the incident arcs. Otherwise, we generally see consistency in savings with the first scenario. Fig. 14c presents the results for the third scenario and savings for this scenario are mostly less than the other scenarios since the incident is more likely to be fully cleared by the time the vehicle reaches the incident arcs. To illustrate the results better, we also report the path distributions for the case where incident took place on arc 4 (other results are not shown for conciseness). Fig. 15a reports the path distribution of the dynamic policy in the absence of explicit modeling of non-recurrent congestion due to the incident that took place 10 min before trip start time. Fig. 15b–d report path distributions under explicit modeling of incidents and the resulting non-recurrent congestion, with trip start times of 10, 20, and 30 min into the incident, respectively. Since the incident is on path 1, there is no routing on path 1 for the case when trip starts just 10 min after the incident occurred (Fig. 15b). As time passes, since the probability of incident clearance and no delay regime increases, dynamic routing policy starts to select this path as well (Fig. 15c and d).

5.5. Computational performance

A key ingredient of practical routing algorithms is their computational efficiency. This is especially important for routing under real-time traffic information where using the latest information provides better routing performance. We use backward dynamic programming algorithm to identify the optimal dynamic policy for the MDP presented in Section 4. The computational performance of the backward recursion suffers from the curse of dimensionality which is determined by the size of the network (e.g., number of nodes and links) as well as the cardinality of other state space dimensions. Since we consider JIT pickup/delivery service for a limited set of origin and destination nodes (e.g., plants and depots), we identify the optimal routing policies offline (e.g., on a regular basis such as every month). These policies are identified and stored for all state combinations at the origin nodes (e.g., different start times and congestion states) while accounting for only the recurring congestion. Hence, the computational complexity of dynamic routing under recurrent congestion is simply the burden of querying the optimal policy table, which is negligible with efficient data structures and fast and reliable communications data link.

In the case of non-recurrent congestion, the number of possible state combinations increases significantly since the state space includes the incident link location, time elapsed since the onset of the incident, and other characteristics of the incident. A priori consideration of all possible incident scenarios is thus not
practical and the optimal policy needs to be recalculated in real-time as the incident information becomes available. When the incident occurs long before the trip start time, then the computational complexity of calculating optimal policy is not important as the impact of incident will dissipate through clearance. The computational performance is a concern if the incident occurs just before the trip start or en-route to the destination. In such cases, we cope with the computational complexity by using a sub-network \( G^0 = (N^0, A^0) \) which is smaller than the entire road network \( G = (N, A) \) where \( N^0 \subseteq N \) and \( A^0 \subseteq A \). The rationale behind using a restricted network is that not all nodes and links are important and their exclusion from the network is not crucial for the optimal policy. While some nodes and links are not at all included in the optimal policy for either being congested or too distant, some links that are a part of the policy might hardly be selected. Hence, a restricted network which includes majority of the links that are in the optimal policy could provide a near optimal dynamic routing policy. In order to identify the restricted network, we employ the \( k \)-shortest path approach presented in [50] which is an improved version of the algorithm introduced in [49]. Since this approach is based on deterministic and static link travel times, we modify the method by using mean link travel times at the link arrival times. The restricted network consists of all links and nodes present in any of the \( k \)-shortest paths identified. The choice of \( k \) is important since larger \( k \) values increase the chance of finding the optimum dynamic policy, but at same time, will require greater computational effort. Fig. 16 illustrates the impact of \( k \) on the size of the restricted network (denoted by \( G^0 \)) for four of the OD pairs listed in Table 2. While the number of links and nodes for the sub-network \( G \) are monotonously increasing with \( k \), those that are actually used in the optimal policy are mostly steady (as shown with mean, minimum and maximum number of links and nodes across all start times).

Table 3 illustrates the effect of network size on the CPU times for finding the optimal dynamic policy. As \( k \) increases, the sub-network \( G^0 \) grows proportionally. In comparison, the CPU time increases exponentially and is sometimes excessive for real-time dynamic routing, e.g., OD pair 3 with \( k=25 \). However, including more nodes and links in \( G \) has diminishing improvement in the performance of the optimal dynamic policy (Fig. 16). This is because some nodes and links are more preferable (dominant) at all congestion states and trip start times than other links and nodes.

<table>
<thead>
<tr>
<th>#k SP</th>
<th>OD pair 1</th>
<th>OD pair 2</th>
<th>OD pair 3</th>
<th>OD pair 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>( k )</td>
<td>( G^0 )</td>
<td>CPU time (s)</td>
<td>( G^0 )</td>
<td>CPU time (s)</td>
</tr>
<tr>
<td>10</td>
<td>10 4.1 0.4 0.1 0.7</td>
<td>12 2.9 0.5 0.3 1.6</td>
<td>15 2.3 0.4 0.2 1.9</td>
<td>18 3.4 0.6 0.3 1.8</td>
</tr>
<tr>
<td>20</td>
<td>14 5.0 0.5 0.2 1.3</td>
<td>16 3.9 0.6 0.3 1.6</td>
<td>18 4.4 0.6 0.3 1.7</td>
<td>20 5.1 0.6 0.3 1.7</td>
</tr>
<tr>
<td>25</td>
<td>18 7.1 0.6 0.4 1.9</td>
<td>20 6.0 0.7 0.4 2.0</td>
<td>22 7.6 0.7 0.4 2.1</td>
<td>24 8.1 0.7 0.4 2.2</td>
</tr>
</tbody>
</table>
The choice of \( k \) depends on the particular OD pair and trip start time. For instance, the path distributions for OD pairs 2, 3, and 4 with \( k=25 \) shortest paths are almost identical with \( k=5 \) or 10. In the case of OD pair 1, while the path distributions differ by less than 5\%, the differences in the expected trip times for \( k=10 \) and 25 are statistically insignificant. Therefore, when there is an incident just before the trip start or en-route, the proposed dynamic routing can be used to obtain a policy using a restricted network obtained through \( k \)-shortest paths specific to the particular OD pair and start time. The choice of \( k \) depends on the available computational time to support the real-time routing (e.g., \( k=10 \) for OD pair 3 and \( k=25 \) for OD pairs 1, 2, and 4) and is determined offline for each OD pair and trip start time combination.

6. Conclusions

The paper proposes practical dynamic routing models that can effectively exploit real-time traffic information from Intelligent Transportation Systems (ITS) regarding recurrent congestion, and particularly, non-recurrent congestion stemming from incidents (e.g., accidents) 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. While non-recurrent congestion is known to be responsible for a major part of network congestion, extant literature mostly ignores this in proposing dynamic routing algorithms. We model the problem as a non-stationary stochastic shortest path problem with both recurrent and non-recurrent congestion. We propose effective data driven methods for accurate modeling and estimation of recurrent congestion states and their state transitions. A Markov decision process (MDP) formulation that generates a routing “policy” to select the best node to go next based on a “state” (vehicle location, time of day, and network congestion state) is proposed to solve the problem. While optimality is only guaranteed if we employ the full state of the transportation network to derive the policy, we recommend a limited look ahead approach to prevent exponential growth of the state space. The proposed model also estimates incident-induced arc travel time delay using a stochastic queuing model and uses that information for dynamic re-routing (rather than anticipate these low probability incidents).

ITS data from South-East Michigan road network, collected in collaboration with Michigan Intelligent Transportation System Center, is used to illustrate the performance of the proposed models. Our experiments clearly illustrate the superior performance of the SDP derived dynamic routing policies when they accurately account for recurrent congestion (i.e., they differentiate between congested and uncongested traffic states) and non-recurrent congestion attributed to incidents. 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). Experiments also show that explicit treatment of non-recurrent congestion can yield significant savings.

Further research will focus on developing dynamic routing algorithms for supporting ‘milk-runs’ where a vehicle departs from an origin to serve several destinations in a network with one or more of the following settings: (1) stochastic time-dependent network where vehicles may encounter recurrent and/or non-recurrent congestion during the trip, (2) vehicle must pickup/deliver within specific time-windows at customer locations, (3) stochastic dependencies and interactions between arcs’ congestion states, and (4) anticipate and respond to the behavior of the rest of the traffic to the real-time ITS information.

Acknowledgments

We are grateful to the anonymous reviewers whose valuable suggestions and comments improved the technical content and
presentation clarity of this manuscript. This work was supported by funds from the US Department of Transportation and the Michigan Department of Transportation through the MIOH University Transportation Center.

Appendix

Lemma 1. The incident-induced delay parameters $(c,q)$, satisfying the following condition for the minimal waiting time of $A$ (smallest discrete time interval), ensures that waiting at the incident node does not reduce the expected travel time:

$$\mu_d(t_k, A, s) - \mu_q(t_k, s) \geq -q/c A.$$ 

Proof. Let $a \in A$ denote the incident arc with origin and destination nodes $(n_k, n_{k+1})$. Further, let $t_{k+1} = t_k + \delta_t(t_k, s, t_k - t_{inc}, 0) + \delta_d(t_k, s, t_k - t_{inc}, 0)$ represent the arrival time to the node $n_{k+1}$ after departing from $n_k$ at time $t_k$. Then the expected travel time from node $n_k$ to the trip destination node $(n_{nk})$ under an optimal policy is $E(\delta_d(t_k, s, l) = (t_k - t_{inc}^0) + F'(n_k, t_k, \delta_d(t_k, s, t_k - t_{inc}, 0; w)))$, where the second term is the cost-to-go from node $n_{k+1}$ at time $t_k$ with congestion state vector $w$ for future arcs at $t_{k+1}$. Let’s denote the expected travel time from node $n_k$ to the trip destination node $(n_{nk})$ at time $t_k$ and $t_k + A$ with $D(t_k)$ and $D(t_k + A)$, respectively.

$$D(t_k + A) = \delta_d(t_k, s, t_k - t_{inc}^0) + F'(n_k, t_k, \delta_d(t_k, s, t_k - t_{inc}^0, 0; w)),$$

$$D(t_k + A) = \delta_d(t_k, s, t_k - t_{inc}^0) + F'(n_k, t_k, \delta_d(t_k, s, t_k - t_{inc}^0, 0; w)) - A.$$ 

Assumption 3 states that at any node arrival time $(t_k)$, waiting at the node does not lead to lower destination arrival time than without waiting. We write this condition for the minimal waiting time of $A$ unit time (smallest discrete time interval),

$$E(D(t_k + A)) - E(D(t_k)) \geq -A.$$ 

We assume that cost-to-go functions alone satisfy this relationship as we assumed that link travel times (in both congestion states) and state transitions are such that waiting at a node does not provide travel time savings in the recurrent congestion (e.g., first-in-first-out property). For a waiting time this leads to the following relation for every $t_k$:

$$F'(n_k, t_k, A, s, t_k + A - t_{inc}^0, 0; w) - F'(n_k, t_k, A, s, t_k, 0; w) \geq -A.$$ 

Hence, we have the following relation:

$$E(\delta_d(t_k, s, t_k, A - t_{inc}^0)) - E(\delta_d(t_k, s, t_k, 0; w)) \geq -A,$$

where,

$$E(\delta_d(t_k, s, t_k, 0; w)) = E(\delta_d(t_k, s, l = 0) + \Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0))$$

$$= \mu_d(t_k, s) + E(\Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0),$$

and, $\mu_d(t_k, s)$ is the mean travel time on arc $a$ at time $t_k$ with congestion state $s$. The expression $E(\Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0))$ can be expressed in two alternative closed-form expressions. In the first case, we assume that the vehicle experiences the maximum delay (i.e. fixed-delay regime in Fu and Rilett [29]), e.g., $E(\Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0)) = -q/c A$. The other alternative is the variable-delay regime in which the vehicle experiences a delay somewhere between the no-delay and the maximum delay [29].

$$E\{\Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0)\} = \frac{c-p}{c} \mu_{inc} - \frac{c-q}{c} (t_k - t_{inc}^0).$$

Note that the waiting decision at the incident node is reasonable only in the case of incident queue dissipation, i.e. either the incident is cleared but the queue is not fully dissipated or the incident is not cleared but the vehicle will exit the link before the clearance. This corresponds to the variable-delay regime and we will show that this holds true by comparing the conditions derived for each case. We first express the no node waiting condition under incident for variable-delay regime as:

$$E(\delta_d(t_k, A, s, t_k - t_{inc}^0, 0)) - E(\delta_d(t_k, s, t_k - t_{inc}^0, 0)) \geq -A,$$

$$\mu_d(t_k + A, s) + E(\Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0)) \geq -D,$$

$$\mu_d(t_k + A, s) - \mu_q(t_k, s) \geq -q/c A.$$ 

When we take the limit $A \to 0$, we have, $(d\mu_q(t,s)/dt)_{t=0} \geq -q/c$. In the maximum delay case, the no node waiting condition can be expressed as

$$E(\delta_d(t_k, A, s, t_k - t_{inc}^0, 0)) - E(\delta_d(t_k, s, t_k - t_{inc}^0, 0)) \geq -A,$$

$$\mu_d(t_k + A, s) + E(\Theta_d(0, c, s, p, q, l = t_k - t_{inc}^0)) \geq -D,$$

$$\mu_d(t_k + A, s) - \mu_q(t_k, s) \geq \frac{q}{p} A.$$ 

When we take the limit $A \to 0$, we have, $(d\mu_q(t,s)/dt)_{t=0} \geq -q/p$. Note that since the capacity under incident is less than regular capacity, i.e. $c > p$, we have the condition for variable-delay regime more strict than the fixed-delay regime, i.e., $-q/c > q/p$. Hence, for arbitrary waiting time $A$, no node waiting condition under incident is

$$\mu_d(t_k + A, s) - \mu_q(t_k, s) \geq \frac{q}{p} A.$$  

References


