1-1-2011

Dynamic routing on stochastic time-dependent networks using real-time information

Ali R. Guner
Wayne State University,

DYNAMIC ROUTING ON STOCHASTIC TIME-DEPENDENT NETWORKS USING REAL-TIME INFORMATION

by

ALI RIZA GÜNER

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2011

MAJOR: INDUSTRIAL ENGINEERING

Approved by:

Advisor Date

DEDICATION

To my wife, Hatice

To my daughter, Fatma Betul

To my parents,

En içten sevgilerimle, ...

ACKNOWLEDGMENTS

I need to thank many people who directly or indirectly helped me with the research presented in this dissertation. I offer my sincere appreciation and gratitude to all of those who I mention here or not. These words are not enough to represent their contributions but an attempt to thank them all.

First of all, I cannot enough express my gratitude to my advisors Dr Ratna Babu Chinnam and Dr. Alper Murat. Their original ideas, perfectionism, enthusiasm and the confidence they have shown in me, have not only directly contributed to the work presented in this dissertation, but have also meant a lot to me personally. I am forever grateful to both of them and feel very lucky to have such great mentors. I will never forget the nice atmosphere that Dr. Chinnam creates in his group and Dr. Murat's intellectually stimulating questions in our team meetings.

I express my sincere appreciation to the committee members, Dr. R. Darin Ellis, Dr. Kyoung-Yun Kim, and Dr. Peter T. Savolainen. I sincerely believe that their valuable inputs to this research make it diverse and successful.

Special thanks go to Dr. Leslie Monplaisir, Chair of the Industrial & Systems Engineering Department at Wayne State University, for his support throughout my doctoral studies and also being an excellent example as a supervisor during my teaching assistantship to him.

I would also like to thank all the members of the Department. In particular, Dr. Ibrahim Dogan, Dr. Akhilesh Kumar, and Nezir Aydin for making study a pleasure in such friendly work environment.

I would like to extend my sincerest thanks to Patricia Martinico, Asst. Director of MIOH University Transportation Center (UTC) for encouraging me to attend regional transportation meetings and conferences. I am grateful MIOH UTC for their sponsorship. This work was supported by funds from the US Department of Transportation and the Michigan Department of Transportation through the MIOH University Transportation Center. I'd like to acknowledge also Michigan ITS center, traffic.com, and Caltrans PEMS for providing their ITS data.

I am more than grateful to my family and I would especially like to thank my parents for always giving me their unconditional support. I can only hope that I made them proud to complete this far away from home yet rewarding journey. I hope that I can manage to become such a good parent as them.

I could not have finished this dissertation without the support from my 'dearest' and 'nearest colleague' Hatice who has been with me all the time during these years of research. No words could thank her enough for her support and constant encouragements.

Thank you!

TABLE OF CONTENTS

LIST OF TABLES

Chapter II:

Chapter III:

LIST OF FIGURES

Chapter II:

Chapter IV:

CHAPTER I: INTRODUCTION

In just-in-time (JIT) manufacturing environments, on-time delivery is one of the key performance measures for dispatching and routing of freight vehicles. Both the travel time delay and its variability impact the efficiency of JIT logistics operations, that are becoming more and more common in many industries, and in particular, the automotive industry. In this dissertation, we first propose a framework for dynamic routing of a single vehicle on a stochastic time dependent transportation network using real-time congestion information from Intelligent Transportation Systems (ITS). Then, we consider milk-run deliveries with multiple pickup and delivery destinations subject to time windows using real-time congestion information. Finally, we extend our dynamic routing model to account for arc interactions on the network and investigate its benefits.

Recurrent and non-recurrent congestion are the two primary reasons for travel time delay and variability, and their impact on urban transportation networks is growing in recent decades. Hence, our routing methods explicitly account for both recurrent and non-recurrent congestion in the network. In our modeling framework, we develop alternative delay models for both congestion types based on historical data (e.g., velocity, volume, and parameters for incident events) and then integrate these models with the forward-looking routing models. The dynamic nature of our routing decisions exploits the real-time information available from various ITS sources, such as loop sensors.

The forward-looking traffic dynamic models for individual arcs are based on congestion states and state transitions driven by time-dependent Markov chains. We propose effective methods for estimation of the parameters of these Markov chains. Based on vehicle location, time of day, and current and projected network congestion states, we generate dynamic routing policies using stochastic dynamic programming formulations.

We have tested the models and algorithms in the networks of Southeast-Michigan and Los Angeles, CA freeways and highways using historical traffic data from the Michigan ITS Center, Traffic.com, and Caltrans PEMS.

1. Motivation

Just-in-time supply chains require reliable deliveries. Hence, JIT supply chains with frequent deliveries in urban areas have to constantly cope with the delivery time unreliability, which, in most part, is caused by traffic congestion. An important characteristic of road networks in many urban-areas is the stochastic and time dependent travel times. The sources of uncertainty are multiple. One of the most significant factors is the high volume of traffic due to commuting. This form of traffic congestion is called recurrent congestion as it occurs during certain time periods and days. One of the most common coping strategies for recurrent congestion is building 'buffer time' into the trip, i.e. starting the trip earlier. However, building 'buffer time' may increase driver and equipment idle time or, sometimes, the buffer may not be enough.

Incidents, such as accidents, vehicle breakdowns, bad weather conditions, work zones, lane closures, special events, etc. are other important disturbances to traffic networks. These disturbances are collectively referred as non-recurrent congestion (or incidents) as the frequency of this kind of congestion is unpredictable. Non-recurrent congestion is a significant part of the total congestion, as described in the *Traffic congestion and reliability report* [1]. Changing the route of the trip in response to an incident is common response by most drivers. However, re-routing around the congested road segments, without an accurate analysis and the use of the real-time information, can cause to travel out-of-the-way and potentially result in more expensive routing outcome in terms of duration or the length of the trip.

There are several ways to improve the traffic operations and hence reduce the transportation times associated with deliveries. Options include developing new infrastructure or expanding existing infrastructure, deploying advanced information technologies, and improving operational management systems. It should be noted that the option of developing new infrastructures, if not infeasible, is significantly more costly. On the other hand, the exploitation of advanced technologies such as the use of information technologies can reduce the level of uncertainty to a manageable level such that the dynamic routing becomes a viable alternative. Accordingly, the dynamic routing allows vehicles use less congested road alternatives and thereby reduce the load on the network chokepoints. ITS infrastructure is now available in most urban areas and provides real-time traffic data and traffic monitoring systems are beginning provide real-time information regarding incidents. In-vehicle communication technologies enable to communicate with vehicles en-route to re-route with real-time information.

The goal of this dissertation is to develop dynamic routing models integrated with congestion delay estimation models that can be easily implemented using available computer and information technologies. With the aid of these technologies, our models will help drivers avoid both recurrent and non-recurrent congestion by dynamically routing the vehicle from an origin to several destinations in traffic networks within given time windows.

2. Research Setting

Our most general model is *a non-stationary stochastic time dependent traveling salesman problem with time windows* (STD-TSP-TW). The Traveling Salesman Problem (TSP) is concerned with finding optimal trip (e.g. with the least travel time, distance, or other performance measure) in which the vehicle starts from the depot, visits every customer in a given set, and returns to the depot. If the travel time between two customers or between a customer and the depot depends on not only the distance/travel time between the customers, but also the time of day of departure then it is called *time-dependent TSP* (TD-TSP). The service time at each customer may also depend on the time of day. If the travel times and/or service times are also random values then this lead to another variant of TSP namely, *stochastic TD-TSP* (STD-TSP). Finally, each of the customers may also have imposed time window constraints on delivery time. In literature this is called *STD-TSP with time windows* (STD-TSP-TW). Hence, in the STD-TSP-TW, a vehicle is initially located at the depot, and must serve a number of geographically dispersed customers in a network where travel times are stochastic and time dependent and each customer must be served within a specified time window. The objective is to find the optimum route with minimum total cost of travel and service time in networks with random arc travel times. The randomness of travel times on arcs may be because of several reasons. The recurrent and/or non-recurrent congestion are the two prime reasons [1] and hence we develop delay estimation models for both of these congestion types. In addition, we assume that there might be interaction among network arcs. For instance, an incident on an arc may affect its upstream arcs because of vehicle queue spillback.

3. Research Scope

The forward-looking traffic dynamic models for individual arcs are based on congestion states and state transitions driven by time-dependent Markov chains. Namely, the state of the next time period depends on only the state of the previous time period. Then our problem may be modeled based on Markov decision process (MDP). We assume state set of the MDP is based on the position of the vehicle, the time of the day and the (recurrent and non-recurrent) congestion states of the arcs. We propose effective methods for estimation of the parameters of these Markov chains. Based on vehicle location, time of day, and current and projected network congestion states, we generate dynamic routing policies using stochastic dynamic programming formulations.

We assume recurrent (peak-time) congestion states are based on the average speed of the vehicles, time of the day, and a cut-off speed. The congestion state classes (i.e.: congested, uncongested, etc.) of the roads may be determined with historic traffic data from ITS center based on Gaussian Mixture Model (GMM). Since, not all of the network information affects an optimal decision, we assume the arc set of a state such that only the arcs those are close the vehicle affects the decision. We also assume that the traffic data for some of the arcs may be unavailable.

Non-recurrent (incident induced) congestion model for a vehicle routing problem has to address incident-induced arc travel time delay and incident clearance time. To address these we assume incident-induced arc travel time delay may be estimated with a stochastic queuing model where the incident duration is stochastic. The time passes until the clearance of incident scene and restoring of the road capacity to normal operation capacity is usually defined as the incident duration. An incident has typically four stages (report/detect response, clearance, and recovery) and the incident duration refers the total of first three stages of the incident. To model incident duration we assume every incident eventually will be cleared and the likelihood of ending increases as more time passes (hazard rate property).

We also assume stochastic dependency among arcs. Although arcs in most traffic networks are statistically dependent (arc-wise and time-wise), there are only a few studies on modeling dependency between arcs.

4. Objectives and Contributions

The objective of our study is to develop methods for routing vehicles in stochastic road network environments representative of real-world conditions. The specific objective is developing dynamic routing algorithms for stochastic time-dependent shortest path problem and stochastic time-dependent TSP with time-windows in networks where vehicles may encounter recurrent and/or non-recurrent congestion during the trip. The prerequisites for this objective are:

- Ability to project future recurrent congestion states based on historical and ITS data
- Ability to anticipate future evolution of non-recurrent congestion throughout the network while accounting for stochastic dependency and interactions between arcs
- Ability to account the traffic interactions among network arcs on our routing policy.

In the literature some aspects of this problem have been studied at some level but there does not exist any study that takes into account all aspects of our dynamic routing problem. Our contributions may be listed as:

1) Methods for accurate and efficient representation of recurrent congestion, in particular, identification of multiple congestion states and their transition patterns.

2) Extension of modeling recurrent congestion and estimation of transition probabilities methods to the case where there is arc interactions among arcs.

3) Integrated modeling and treatment of recurrent and non-recurrent congestion for vehicle routing and demonstrating the need and value of such integration.

4) An integrated methodology for identifying the traveling salesman problem (TSP) tours in stochastic time dependent (STD) networks where the stochastic path travel times between pairs of pickup and delivery sites are estimated through optimal dynamic routing.

5) An approach for dynamic routing between pairs of sites in STD networks using the real-time congestion information available from ITS sensor networks.

6) Transportation cost and delivery service level improvement based on optimal dynamic routing between sites and demonstrating this fact with using real network traffic data.

Summarizing, the original contributions of this dissertation can be quantified in terms of the following technical publications:

4.1 Publications

- A.R. Guner, A. Murat, R.B. Chinnam: *Dynamic Routing Under Recurrent and Non-Recurrent Congestion Using Real-time ITS Information*. Computers & Operations Research, Article in Press, 39(2), 2012. [2]
- A.R. Guner, R.B. Chinnam, and A. Murat: Dynamic Routing Using Real-time ITS Information. Proceedings of Third International Workshop on Intelligent Vehicle Controls & Intelligent Transportation Systems. Milan, Italy, July 2009. [3]

- A.R. Guner, A. Murat, R.B. Chinnam: *Dynamic Routing in Stochastic Time-Dependent Networks for Milk-run tours with time windows*. (Under Review, Transportation Research Part E, 2010)
- A.R. Guner, R.B. Chinnam, A. Murat: Dynamic Routing in Stochastic Time-Dependent Networks under Arc Interactions. (To be submitted to IEEE Transactions on ITS, 2011)

5. Organization of the Dissertation

The dissertation is organized as follows. In Chapter 2 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. We consider dynamic routing under milk-run tours with time windows in congested transportation networks in Chapter 3. The proposed method integrates TSP with dynamic routing to find a static yet robust recurring tour of a given set of sites (i.e., DC and suppliers) while dynamically routing the vehicle between site visits. Chapter 4 proposes methods for minimizing expected travel time from an origin to a destination in a stochastic time-dependent network with arc interactions to improve delivery efficiency.

Since 2nd, 3rd, and 4th chapters are stand-alone manuscripts submitted (to be submitted) to journals, some of the sub-sections might be repeated.

CHAPTER II: DYNAMIC ROUTING UNDER RECURRENT AND NON-RECURRENT CONGESTION USING REAL-TIME ITS INFORMATION*

Abstract – 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

* This chapter resulted in the following publications:

- A.R. Guner, A. Murat, R.B. Chinnam: Dynamic Routing Under Recurrent and Non-Recurrent Congestion Using Real-time ITS Information. Computers & Operations Research, Article in Press, 2009

- A.R. Guner, R.B. Chinnam, and A. Murat: Dynamic Routing Using Real-time ITS Information. Proceedings of Third International Workshop on Intelligent Vehicle Controls & Intelligent Transportation Systems. Milan, Italy, July 2009 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 Southeast-Michigan freeways using historical data from the MITS Center and Traffic.com.

Keywords – JIT logistics, transportation, congestion, incidents, dynamic routing, ITS

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 [4]. 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 [5]. These requirements have now become the norm in the US as well. For example, many automotive final assembly plants in Southeast 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 [6]. 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 [7]. 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.

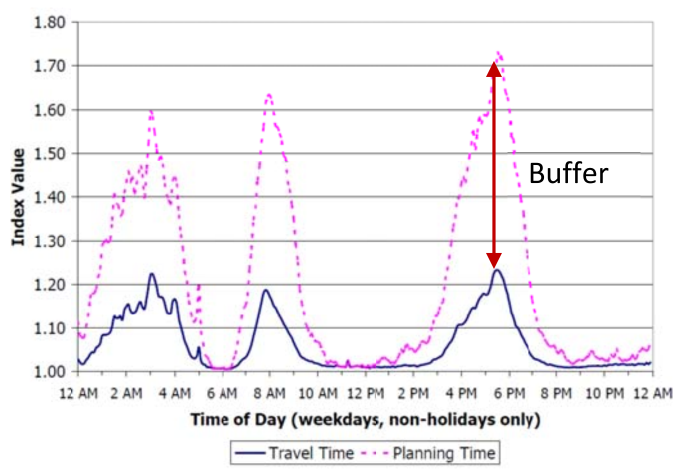

Fig. 1. Extra buffer time needed for on-time delivery with 95% confidence in Detroit [7].

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 [1] 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 Southeast 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 suboptimal 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 queuing 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 [8]. To find the path with minimum total expected cost, Frank [9] suggested replacing arc costs with their expected values and subsequently solving as a deterministic SP. Loui [10] showed that this approach could lead to sub-optimal paths and proposed using utility functions instead of the expected arc costs. Eiger et al. [11] showed that Dijkstra's algorithm [12] 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 [13] 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 [13] employed the dynamic programming (DP) approach to derive the optimal policy. Bertsekas and Tsitsiklis [14] proved the existence of optimal policies for STD-SP. Later, Fu and Rilett [15] modified the method of Hall [13] 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 [16] 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 [17] 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 [18] suggested polynomial algorithms to find optimal policies for stochastic shortest path problems with one-step arc and limited temporal dependencies. Gao and Chabini [19] 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 [20] proposed a modified label-correcting algorithm. Miller-Hooks and Mahmassani [21] extends [20] 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 [18] and Gao and Chabini [19]. 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. Polychronopoulos and Tsitsiklis [22] 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 Tsitsiklis [23] 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 [24] extended [22] 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. [25] studied a similar problem as in [22] 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. [26] proposed state space reduction methods. A limitation of Kim et al. [26], 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.

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 [27-29], queuing theory [30-35], and statistical (regression) models [36-38]. 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 [39] derived probability distributions of delay under flow stopping. Truck-involved incident duration is studied by Golob et al. [40] and employs lognormal distribution. Analysis of variance is examined by Giuliano [41] and a truncated regression model to estimate incident duration is proposed by Khattak et al. [42] for incident durations in Chicago area. Gamma and exponential distributions are also suggested as good representations of incident duration distribution [43]. 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 [44]. Nam and Mannering [45] applied hazard-based duration models to model distribution of detect/report, respond and clear durations 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 Ruszczynski [46] 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 [47] consider the incident clearance process and adopt the models in Kim et al. [25] 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 $a \equiv (n,n') \in A$, if and only if, there is a road that permits traffic flow from node n to n'. Given an origin-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_0 and n_d, 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, $a \equiv (n,n') \in A$ 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 $r+1 \in Z^{+}$ 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 $c_a^{i-1}(t)$ and $c_a^{i}(t)$ for *i=1,2,...,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 $s_a^i(t)$ as the *i^th^* traffic congestion state of arc a at time t, i.e. $s_a^1(t)=\{1\}$ and $s_a^r(t)=\{\text{Congested at level r}\}=\{r\}$. 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., $s_a^0(t)=\{\text{Uncongested}\}=\{0\}$ and $s_a^1(t)=\{\text{Congested}\}=\{1\}$. Congestion state, $s_a^i(t)$ of the arc a at time t can then be determined as:

$$s_a(t)=\{i\text{, if } c_a^{i-1}(t)\le v_a(t)<c_a^i(t)\} \tag{1}$$

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., $S(t)=\{s_1(t),s_2(t),\dots,s_{|A|}(t)\}$ 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 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 $i=s_a(t)$ to state $j=s_a(t+1)$ 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:

$$\alpha_a^{ij}(t)=\frac{\left|c_a^{i-1}(t)\le V_a(t)<c_a^i(t)\cap c_a^{j-1}(t+1)<V_a(t+1)<c_a^j(t+1)\right|}{\left|c_a^{i-1}(t)\le V_a(t)<c_a^i(t)\right|} \tag{2}$$

Let $T_a(t,t+1)$ denote the matrix of state transition probabilities from time t to time $t+1$, then we have $T_a(t,t+1)=\left[\alpha_a^{ij}(t)\right]_{ij}$. We further assume that arc *a*'s congestion state is independent of other arcs' states, i.e. $P\{s_a(t+1)\,|\,s_{a'}(t+1),s_a(t)\}=P\{s_a(t+1)\,|\,s_a(t)\}=\alpha_a^{ij}(t)$ for $\forall a'\in A$. 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 [48]. 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 δ is a positive integer number. Given the independence assumption of arcs' congestion states, this can be formulated as follows:

$$P\left(S(t+\delta)\,|\,S(t)\right)=\prod_{a=1}^{|A|}P\left(s_a(t+\delta)\,|\,s_a(t)\right) \tag{3}$$

Then the congestion state transition probability matrix for each arc in δ periods can be found by the Kolmogorov's equation [49]:

$$T_a(t,t+\delta)=\left[\alpha_a^{ij}(t)\right]_{ij}\times\left[\alpha_a^{ij}(t+1)\right]_{ij}\times\ldots\times\left[\alpha_a^{ij}(t+\delta)\right]_{ij} \tag{4}$$

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:

$$\delta(t,a,s_a)\sim N\left(\mu(t,a,s_a),\sigma^2(t,a,s_a)\right) \tag{5}$$

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

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 minute, given that it has lasted 14 minutes, is greater than the probability of it being cleared in the 14th minute given that it has lasted 13 minutes. 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 $\lambda(t)=f(t)/(1-F(t))$ 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_{inc}^{0}), current status (i.e. cleared/not cleared), expected duration (μ), and standard deviation (σ) are available through ITS's incident management and incident database systems. Hence, we can estimate the parameters of the Weibull distribution ($\varphi(a,b)$) of the incident clearance duration [49]. 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 [31]. Here incident-induced delay function, $\Theta(\cdot)$, is based on the incident duration φ, road non-incident capacity denoted with c (vehicle per hour, or *vph* in short), road capacity during the incident denoted with ρ(*vph*) and arrival rate of vehicles to the incident arc denoted with q(*vph*). Given these parameters for an incident started at t_{inc}^{0}, the vehicle arriving to the incident arc at time (t) experiences the following expected incident-induced delay:

$$E\left(\Theta(.)\right)=\left(\frac{c-\rho}{c}\right)\left(D_{12}-D_1P_3\right)+P_2d_m \tag{6}$$

where $D_{12}=\int_{D_1}^{D_2}x.\varphi(x)\,dx$, $D_1=\left(\frac{c-\rho}{c-q}\right)\left(t-t_{inc}^0\right)$, $D_2=\left(\frac{q}{\rho}\right)\left(t-t_{inc}^0\right)$,

$d_m=\left(\frac{q-\rho}{\rho}\right)\left(t-t_{inc}^0\right)$, $P_1=\int_0^{D_2}x.\varphi(x)\,dx$, $P_2=\int_{D_2}^{\infty}x.\varphi(x)\,dx$, and $P_3=1-\left(P_1+P_2\right)$.

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, $I(t)$, i.e. $I(t)=\left\{i_1(t),i_2(t),\dots,i_{|A|}(t)\right\}$. Note that if an arc *a* is not an incident arc, then $i_a(t)$=0, otherwise $i_a(t)=t-t_{inc}^0(a)$ and $0<i_a(t)<\infty$, where $t_{inc}^0(a)$ 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. $t_{inc}^0=t_{inc}^0(a)$, 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 same 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, $\delta_a(t,s,i=0)$, and the incident parameters, φ,c,ρ,q,i, we can express the arc travel time with incident as:

$$\delta_a(t,s,i)=\delta_a(t,s,i=0)+\Theta_a\left(\varphi,c,\rho,q,i=t-t_{inc}^0\right) \tag{7}$$

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 Δ (smallest discrete time interval), ensures that waiting at the incident node does not reduce the expected travel time.

$$\mu_a(t_k + \Delta, s) - \mu_a(t_k, s) \geq -\frac{q}{c}\Delta \tag{8}$$

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_d. Let's assume that there is a feasible path between (n_0, n_d) where a path $p = (n_0, .., n_k, .., n_{K-1})$ is defined as sequence of nodes such that $a_k \equiv (n_k, n_{k+1}) \in A$, $k = 0, .., K-1$ 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 arcs* 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^{th} decision stage, t_k is the time at k^{th} decision stage where $t_k \in \{1, ..., T\}$, $T > t_{K-1}$. 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. [26]. 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 final destination, the dynamic route selection problem can be modeled as a dynamic programming model. The state of the system at kth decision stage is denoted by $\Omega\left(n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k}, I_k\right)$. 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 $\{a_{k+2} : a_{k+2} \in PScAS(a_k)\}$, and incident durations (I_k) of the network at stage k, i.e. $I_k \equiv I(t_k)$. The action space for the state $\Omega\left(n_k, t_k, s_{a_{k+1}\cup a_{k+2},k}, I_k\right)$ is the set of current arcs of node n_k, denoted with $CrAS(n_k)$.

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 $\pi = \{\pi_0, \pi_1, ..., \pi_{K-1}\}$ be the policy of the trip and is composed of policies for each of the *K-1* decision stages. For a given state $\Omega_k = \left(n_k, t_k, s_{a_{k+1}\cup a_{k+2},k}, I_k\right)$, 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 $\pi = \{\pi_0, \pi_1, ..., \pi_{K-1}\}$ is as follows:

$$F_0(n_0, t_0, S_0, I_0) = \underset{\delta_k}{E}\left\{g_{K-1}(\Omega_{K-1}) + \sum_{k=0}^{K-2} g_k\left(\Omega_k, \pi_k(\Omega_k), \delta_k\right)\right\} \tag{9}$$

where (n_0, t_0, S_0, I_0) is the starting state of the system. δ_k is the random travel time at decision stage *k*, i.e., $\delta_k \equiv \delta\left(t_k, \pi_k(\Omega_k), s_a(t_k), i_a(t_k)\right) + \Theta(\varphi, c, \rho, q, i)$ and $\Theta(\varphi, c, \rho, q, i = 0) = 0$, i.e. the incident delay of an arc without incident. $g_a(\Omega_k, \delta_k)$ is cost of travel on arc $a = \pi_k(\Omega_k) \in CrAS(n_k)$ at stage k, i.e., if travel cost is a function (ϕ) of the travel time, then $g(\Omega_k, \pi_k(\Omega_k), \delta_k) \equiv \phi(\delta_k)$. Then the minimum expected travel time can be found by minimizing $F\left(n_0,t_0,S_0,I_0\right)$ over the policy vector $\pi=\left\{\pi_0,\pi_1,...,\pi_{K-1}\right\}$ as follows:

$$F^*\left(\Omega_0\right)=\min_{\pi=\left\{\pi_0,\pi_1,...,\pi_{K-1}\right\}} F\left(\Omega_0\right) \tag{10}$$

The corresponding optimal policy is then $\pi^*=\underset{\pi=\left\{\pi_0,\pi_1,...,\pi_{K-1}\right\}}{\arg\min} F\left(\Omega_0\right)$. Hence, the Bellman's cost-to-go equation for the dynamic programming model can be expressed as follows [48]:

$$F^*\left(\Omega_k\right)=\min_{\pi_k} \underset{\delta_k}{E}\left\{g(\Omega_k,\pi_k\left(\Omega_k\right),\delta_k)+F^*\left(\Omega_{k+1}\right)\right\} \tag{11}$$

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

$$\begin{aligned} F\left(\Omega_k\,|\,a_k\right)=\sum_{\delta_k}P\left(\delta_k\,|\,\Omega_k,a_k\right)\Bigg[\, g\left(\Omega_k,a_k,\delta_k\right)+\sum_{s_{a_{k+1},k+1}}P\Big(s_{a_{k+1},k+1}\left(t_{k+1}\right)|\,s_{a_{k+1},k}\left(t_k\right)\Big) \\ \sum_{s_{a_{k+2},k+1}}P\Big(s_{a_{k+2},k+1}\left(t_{k+1}\right)\Big)\sum_{I_{k+1}}P\left(I_{k+1}\left(t_{k+1}\right)|\,I_k\left(t_k\right)\right)F\left(\Omega_{k+1}\right)\Bigg] \end{aligned} \tag{12}$$

where $P\left(\delta_k\,|\,n_k,t_k,S_k,I_k\right)$ is the probability of travelling arc a_k in δ_k periods. $P\Big(s_{a_{k+2},k+1}\left(t_{k+1}\right)\Big)$ is the long run probability of arc $a_{k+2}:a_{k+2}\in PScAS\left(a_k\right)$ being in state $s_{a_{k+2},k+1}$ 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 $F_k^*\left(\Omega_k\right)$, $k=K-1,K-2,..,0$. In the backward induction, we initialize the final decision epoch such that, $\Omega_{K-1} = \Omega(n_{K-1}, t_{K-1})$, 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., $t_{K-1} > T$.

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 OD pairs. Section 5.4 presents the experimental setup that involves an incident 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 hours 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 [7] to improve quality and reduce data errors.

(a) (b)

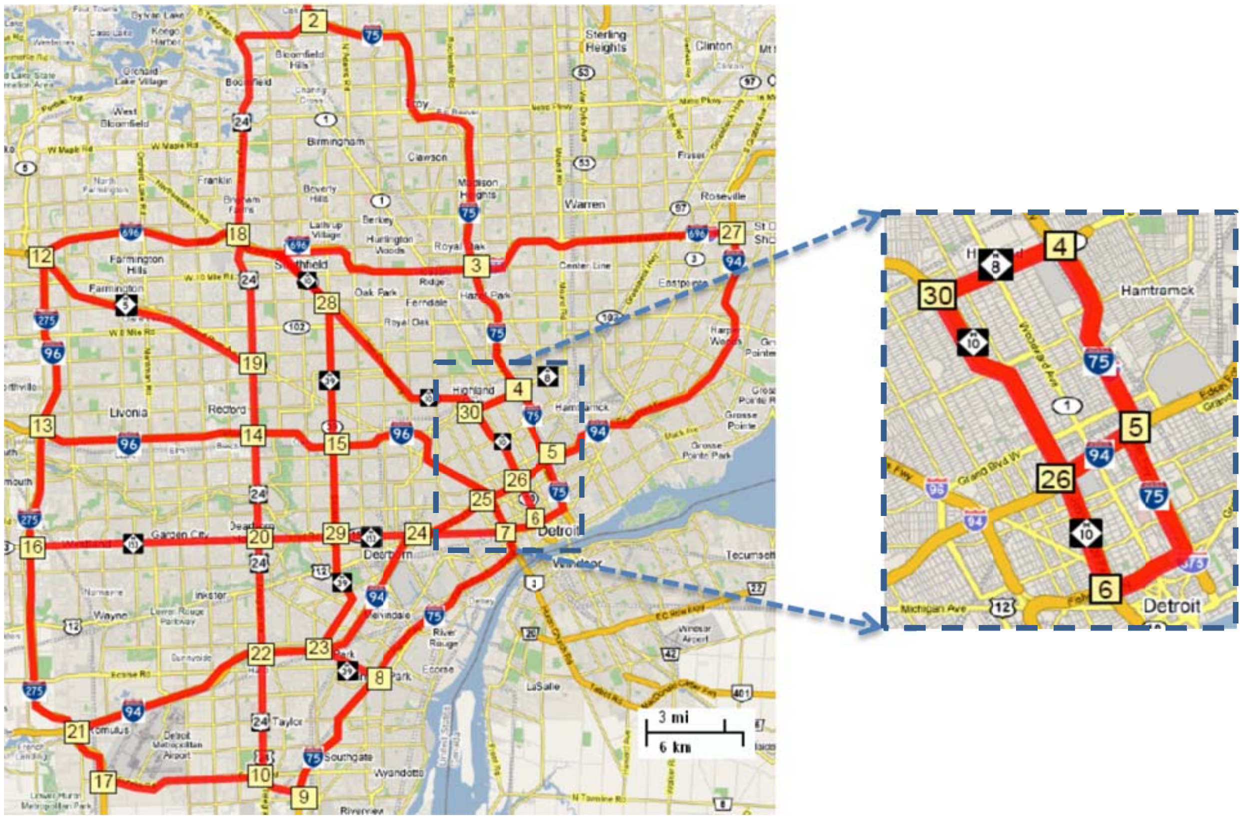

Fig. 2. (a) South-East Michigan road network considered for experimental study. (b) Sub-network from South-East Wayne County.

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.† 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

† Historical data for arcs missing ITS monitoring are obtained from interpolation of floating car data.

these same arcs are shown in Fig. 4, clearly revealing the non-stationary nature of traffic.

Table 1 : Information regarding sub-network nodes and arcs.

			FROM		TO	
Arc ID	Freeway	Length (miles)	Node #	Description (Exit #)	Node #	Description (Exit #)
1	I-94	1.32	5	216	26	215
2	M-8	1.75	4	56A (I-75)	30	7C (M-10)
3	I-75	3.13	4	56A	5	53B
4	I-75	2.81	5	53B	6	50
5	M-10	3.26	30	7C	26	4B
6	M-10	1.42	26	4B	6	2A

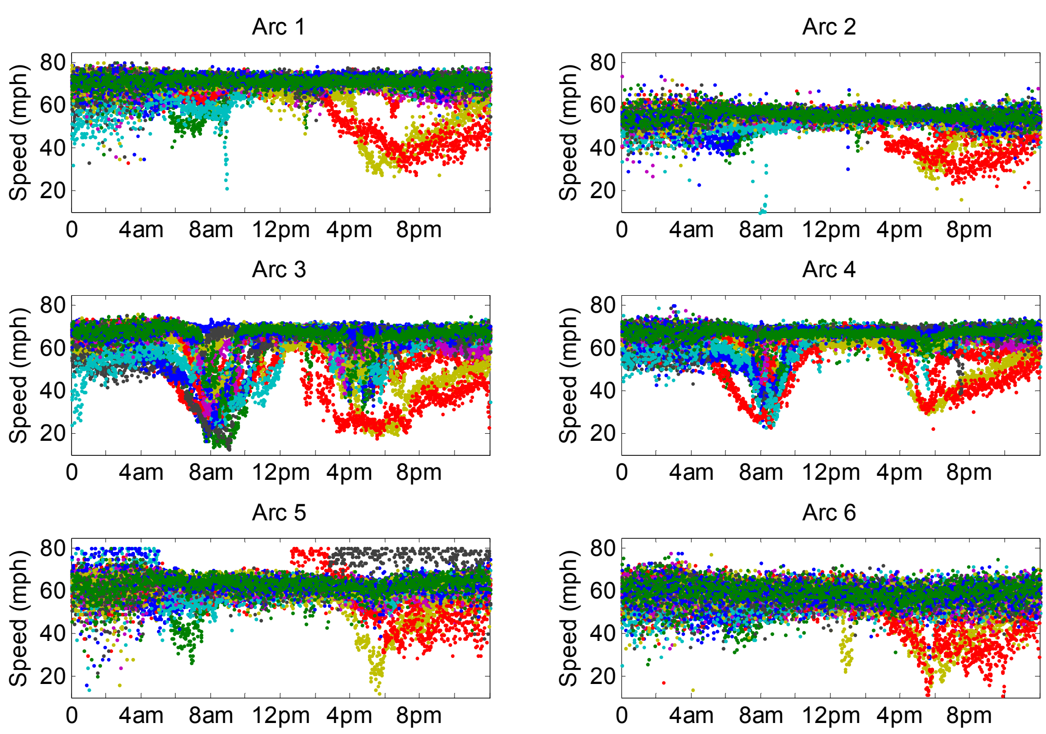

Fig. 3. Raw traffic speeds for arcs on sub-network (mph) at different times of the day. (Data: Weekday traffic from January 21 to February 20. Each color represents a distinct day of 23 days.)

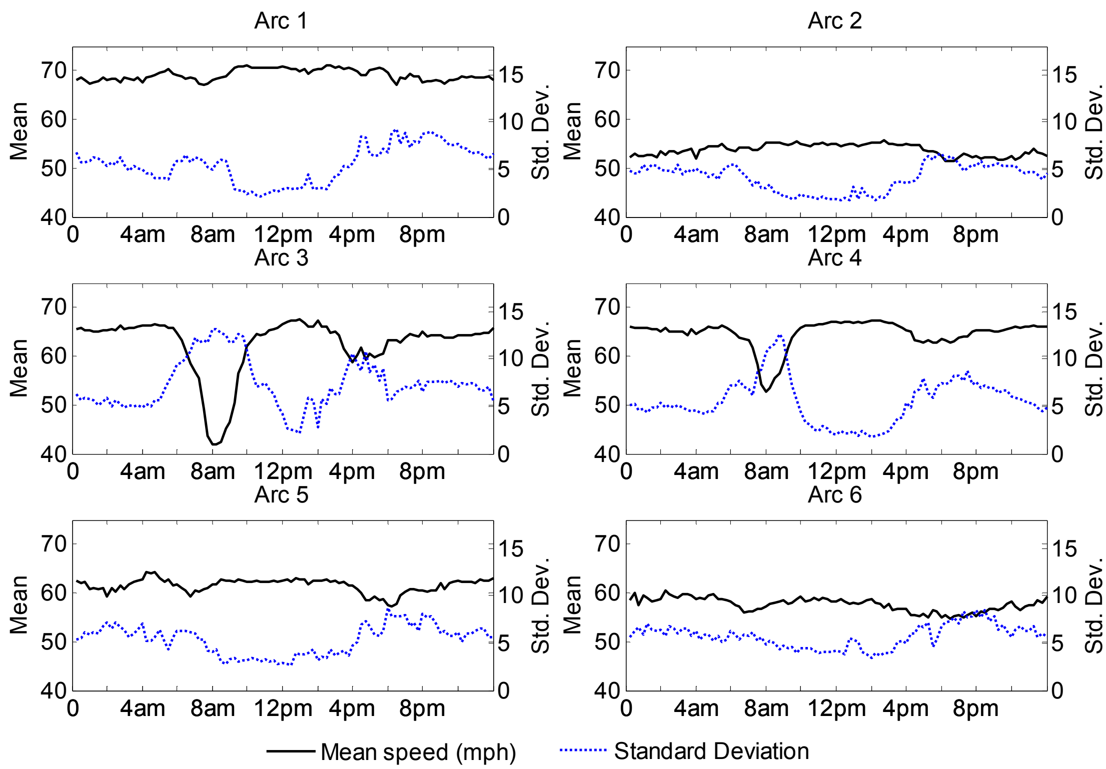

Fig. 4. Traffic mean speeds (mph) and standard deviations by time of the day for arcs on sub-network. (15 minute time interval resolution)

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 [50] for its computational efficiency and performance. To estimate the number of congestion 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 ($\boldsymbol{\mu}^i_{t,t+1};\boldsymbol{\Sigma}^i_{t,t+1}$), where *i* denotes the i^{th} cluster. The Gaussian distribution assumption has been employed by others in the literature (see Kim et al. [25]). 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 intervals, as illustrated in Fig. 5. Formally, the cut-off speed between congestion state-pair $(i, i+1)$ for arc a at time t is denoted by $c_a^i(t)$ and is calculated as follows: $c_a^i(t) = x, x : f_{i_t}(x) = f_{(i+1)_t}(x)$ where $f(\cdot)$ is the projected probability density function for state *i*. Unlike most clustering methods, the GMM clustering procedure employed does not call for specification of number of clusters (i.e., congestion states) in advance and can determine the optimal number of clusters based on the maximum likelihood and model complexity measures. However, we did limit the number of clusters to two, considered quite adequate for modeling recurrent-congestion, and to limit estimation errors attributable to data sparsity.

As expected, the GMM procedure generally yielded mostly two states, even without the constraint, as in Fig. 5 (resulting in states denoted 'congested' and 'uncongested' states with c_1^1(8:30)= 64.9 mph), and rarely a single state during periods of low traffic (as in Fig. 6). Following these observations, we have adopted two congestion states in representing arc congestion dynamics. Note that this does not compromise from the accuracy of congestion modeling, rather provides uniformity in the algorithmic data structures across all arcs in the network.

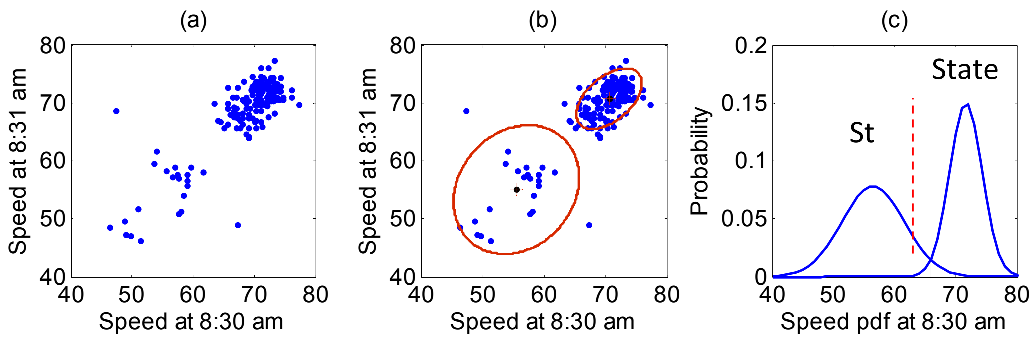

Fig. 5. (a) Joint plots of traffic speeds in consecutive periods for modeling state-transitions at 8:30 am, for arc 1; (b) Cluster joint distributions of speed at 8:30am generated by GMM; (c) Partitioned traffic states based on projections.

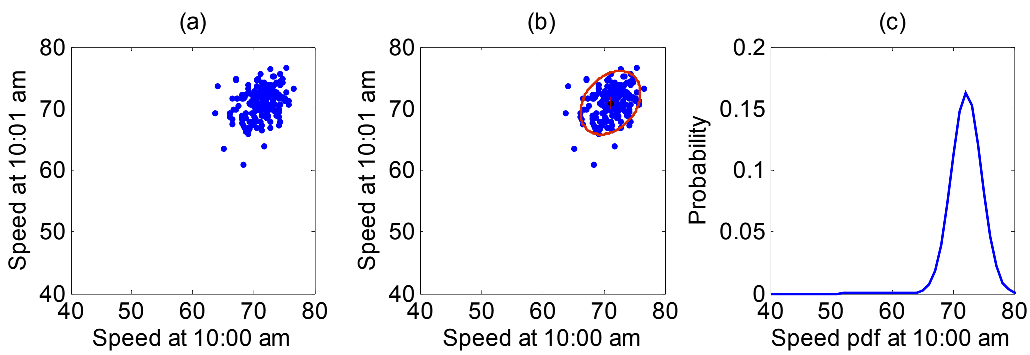

Fig. 6. (a) Joint plots of traffic speeds in consecutive periods for modeling state-transitions at 10:00 am, for arc 1; (b) Single cluster joint distribution of speed at 10:00am generated by GMM; (c) Partitioned traffic states based on projections.

The parameters of the traffic state joint Gaussian distributions (i.e., $\boldsymbol{\mu}^{i}_{t,t+1};\boldsymbol{\Sigma}^{i}_{t,t+1}$) 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, α_t denotes the probability of state transition from congested state to congested state where as β_t denotes the probability of state transition from uncongested state to uncongested state. 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.

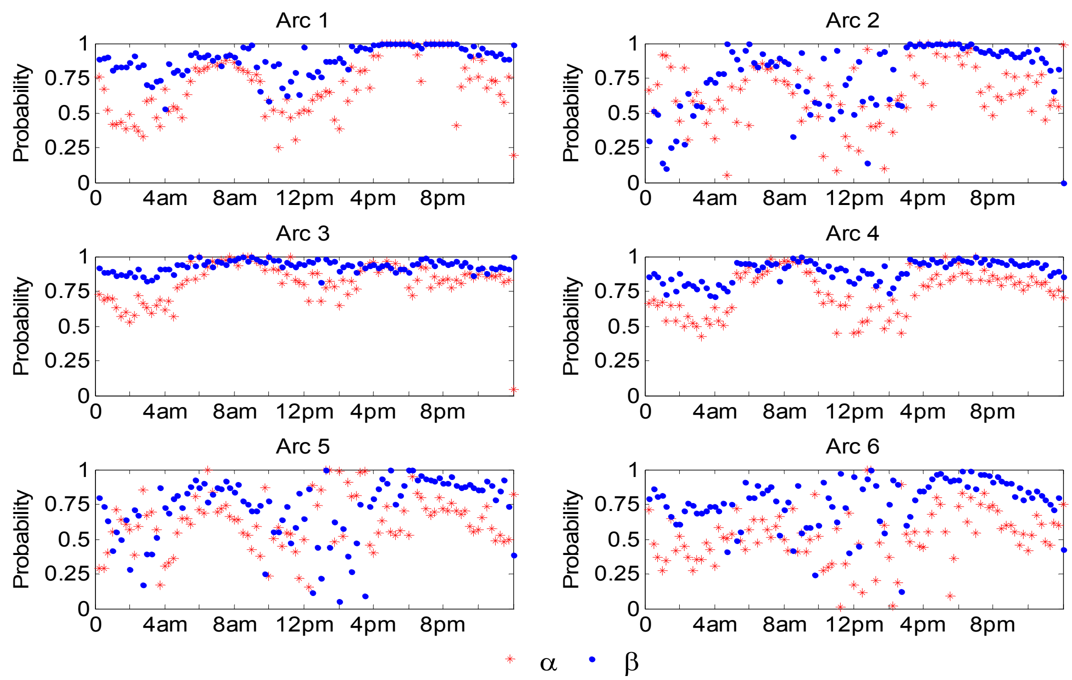

Fig. 7. Recurrent congestion state-transition probabilities for arcs on sub-network. α: congested to congested transition; β: uncongested to uncongested transition probability (plotted with 15 minute time interval resolution).

For the sub-network, the mean and standard deviation of arc travel times are illustrated in Fig. 8 and Fig. 9, respectively, by traffic state and time of day.

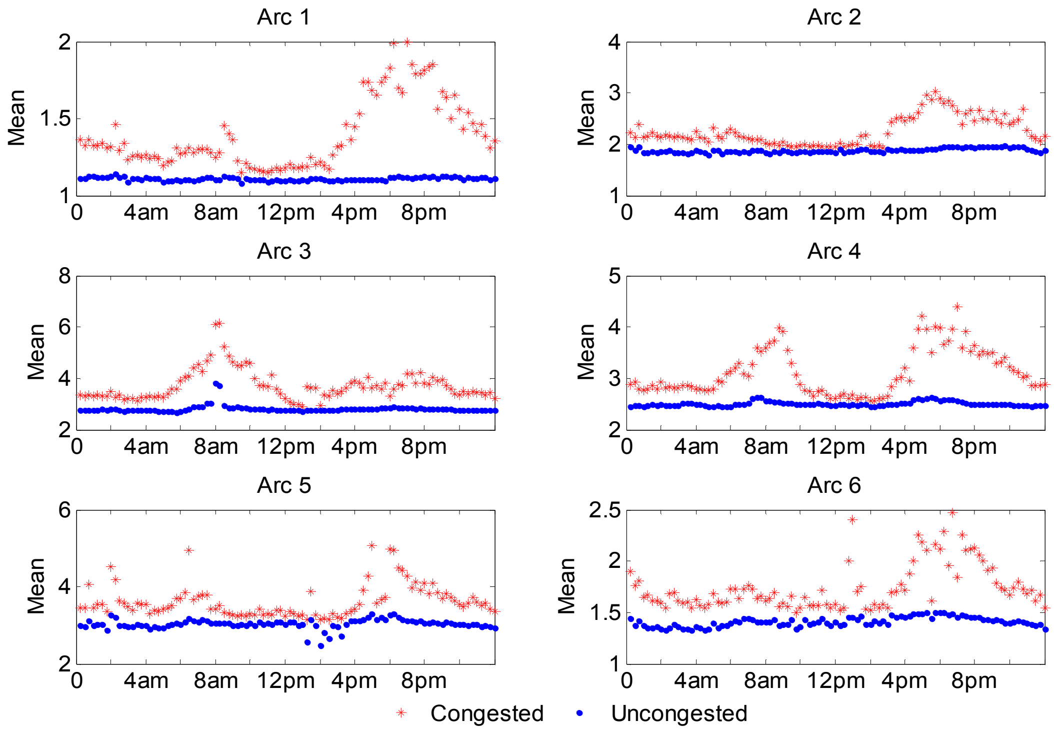

Fig. 8. Sub-network arc travel time means in minutes (plotted with 15 minute time interval resolution).

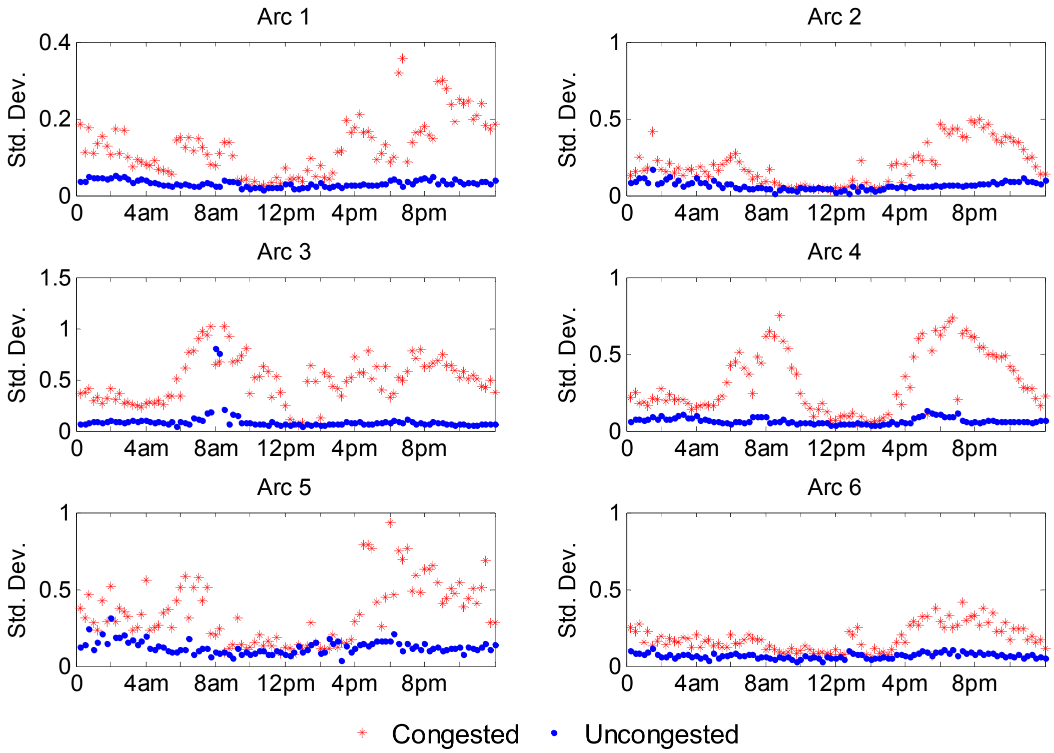

Fig. 9. Sub-network arc travel time standard deviations in minutes (plotted with 15 minute time interval resolution).

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 aims 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, *path1: 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 minute interval starting times (yielding (24×60)/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.

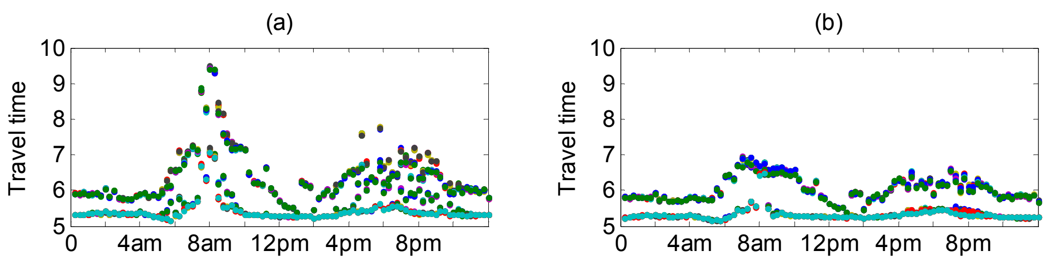

Fig. 10. Mean travel times for all state combinations of the sub-network (each color represents a different state combination): (a) Baseline path. (b) Dynamic vehicle routing policy.

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.

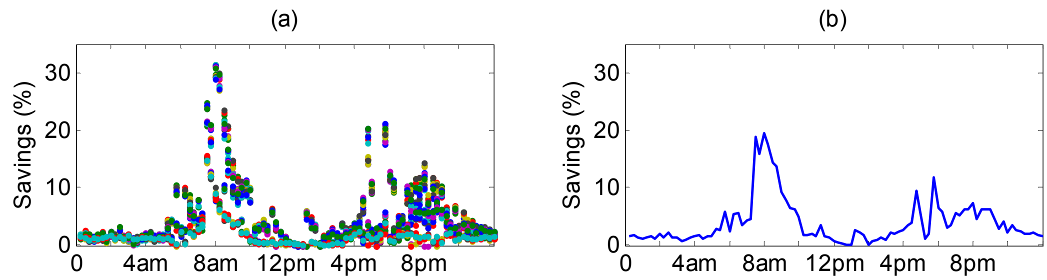

Fig. 11. Savings from employing dynamic vehicle routing policy over baseline path: (a) Savings for each of the 32 network state combinations. (b) Average savings across all state combinations.

Besides the sub-network (Fig. 2b), as listed in Table 2, we have also identified 5 other origin and destination (OD) pairs in Southeast 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.

Table 2: Origin-Destination pairs selected from South-East Michigan road network.

	ORIGIN		DESTINATION	
OD Pair	Node #	Description (Intersection of)	Node #	Description (Intersection of)
1	2	I-75 & US-24	21	I-275 & I-94
2	12	I-96 & I-696	25	I-96 & I-94
3	19	M-5 & US-24	27	I-696 & I-94
4	23	I-94 & M-39	13	I-96 & I-275
5	3	I-75 & I-696	15	I-96 & M-39

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.

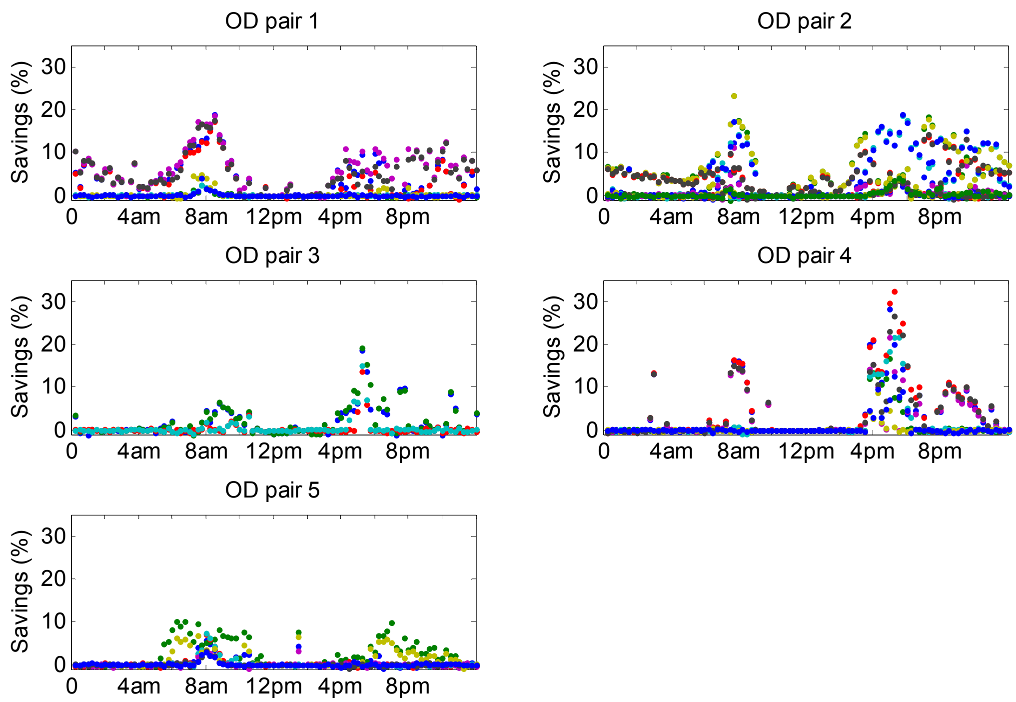

Fig. 12. Savings of dynamic policy over baseline path during the day for all starting states of given OD pairs (with 15 minute time interval resolution).

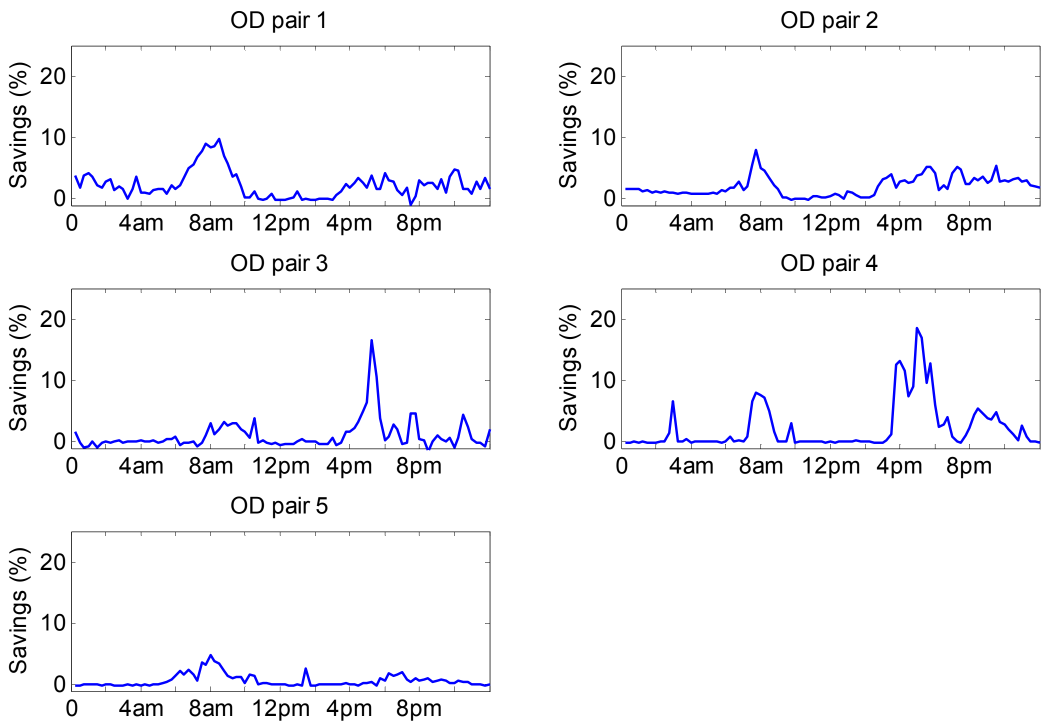

Fig. 13. Average savings of dynamic policy over baseline path during the day for all starting states of given OD pairs (with 15 minute time interval resolution).

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 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): 6:30am, 9:00am, 10:30am, 4:00pm, 5:30pm and 7:00pm. 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 minutes and standard deviation of 5 minutes, 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 in-flow 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 the condition derived in Lemma 1.

The percentage savings from the explicit modeling of non-recurrent congesting along with recurrent congestion during dynamic vehicle routing are illustrated in Fig. 14. The results are very compelling and pertain to three different scenarios. In the first scenario, the incident occurs 10 minutes before vehicle's departure from the starting node. In the second and third scenarios, the incident occurs 20 minutes and 30 minutes before vehicle's departure from the starting node, respectively. For example, if the vehicle departs the origin node at 6:30am, incident is simulated to occur at 6:20am or 6:10am or 6:00am, and incident has not yet been cleared in all three cases.

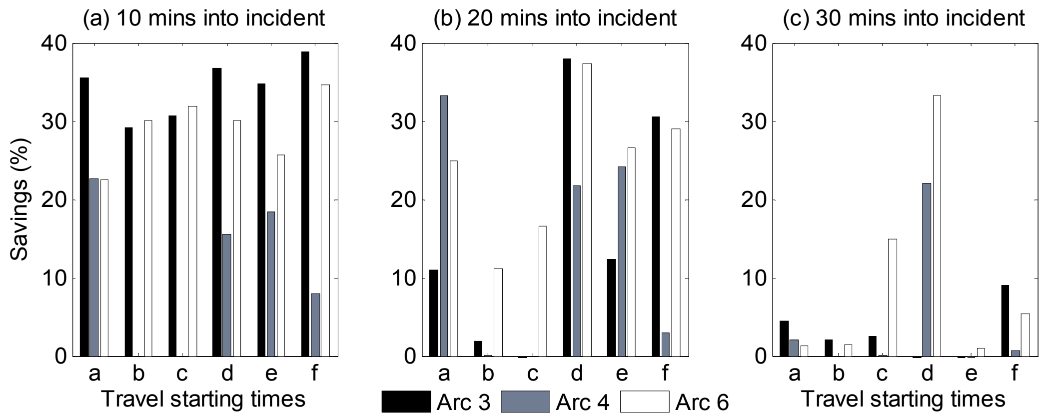

Fig. 14. Savings realized by dynamic routing based on modeling both recurrent and non-recurrent congestion compared to the dynamic routing with only recurrent congestion modeling: a: 6:00, b: 7:30, c: 9:00, d: 16:00, e: 17:30, and f: 19:00. Incident is either on arc 3, or 4, or 6. Trip starts (a) 10 minutes (b) 20 minutes (c) 30 minutes after incident has occurred.

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 mins 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 (because of space limits, we are not showing the other results). 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 minutes before trip start time. Fig. 15b, c, and d report path distributions under explicit modeling of incidents and the resulting non-recurrent congestion, with trip start times of 10, 20, and 30 minutes 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 minutes 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. 15d and d).

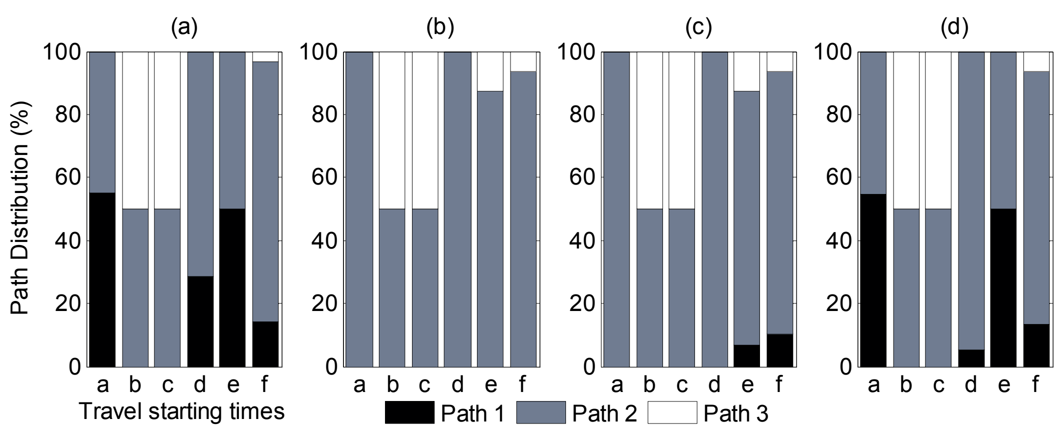

Fig. 15. Path distribution from dynamic routing under an incident on arc 4 for different trip start times: a: 6:00, b: 7:30, c: 9:00, d: 16:00, e: 17:30, f: 19:00. (a) Results without modeling incident and trip starts 10 minutes into incident. (b), (c), and (d) report path distributions under explicit modeling of incidents, with trip start times of 10, 20, and 30 minutes into the incident, respectively.

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 clearence. The computational perfomance 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 $G' = (N', A')$ which is smaller than the entire road network $G = (N, A)$ where $N' \subseteq N$ and $A' \subseteq A$. 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 Martins and Pascoal (2003) which is an improved version of the algorithm introduced in Yen (1971). 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.

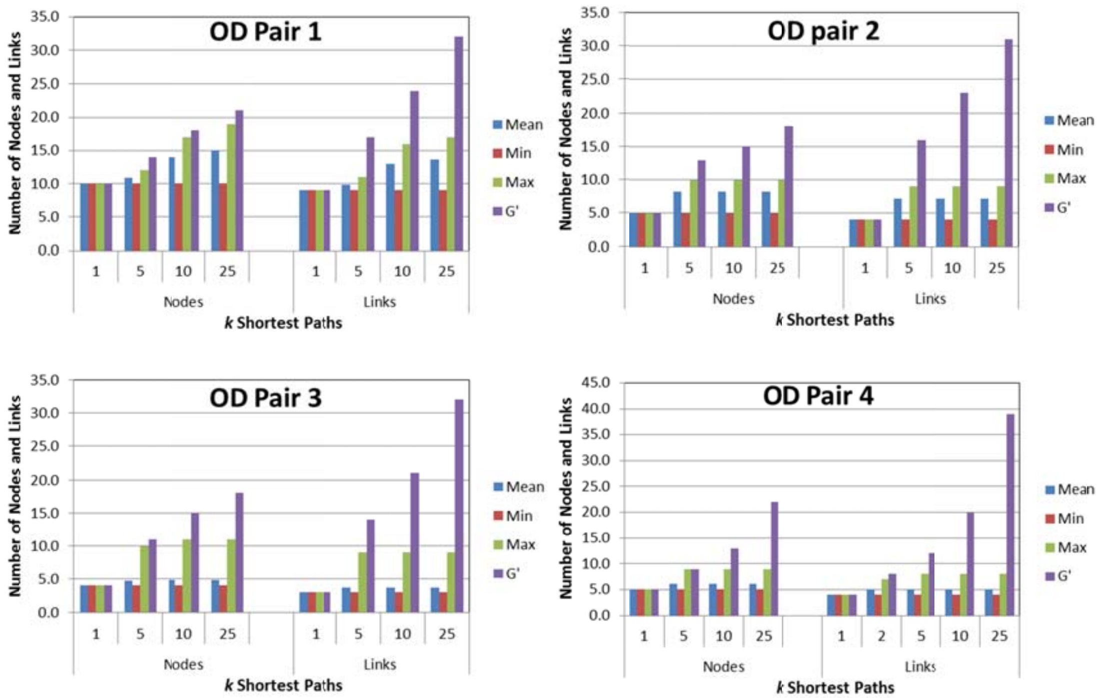

Fig. 16. Number of links and nodes in the sub-network G' and the mean, minimum and maximum number of links and nodes selected in the optimal policy using network G'.

Figure 16 illustrates the impact of *k* on the size of the restricted network (denoted by G') 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).

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' 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 perforamnce 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 3: CPU times for calculating optimal policy for OD pairs 1,2,3, and 4 in Table 2 for k=1,2,10 and 25. Table also reports the number of nodes and arcs that are part of G' for different k.

	OD 1					OD 2					OD 3					OD 4				
	G'		CPU Time (sec)			G'		CPU Time (sec)			G'		CPU Time (sec)			G'		CPU Time (sec)		
# k SP	N'	A'	Mean	Min	Max	N'	A'	Mean	Min	Max	N'	A'	Mean	Min	Max	N'	A'	Mean	Min	Max
1	10	9	0.4	0.3	1.4	5	4	0.3	0.1	1.3	4	3	0.1	0.0	1.1	5	4	0.2	0.1	1.0
5	14	17	0.5	0.4	0.7	13	16	0.8	0.6	1.3	11	14	0.4	0.2	0.9	9	12	0.2	0.1	0.4
10	18	24	0.8	0.5	2.9	15	23	1.3	0.9	2.0	15	21	1.5	0.8	4.0	13	20	0.5	0.4	0.9
25	21	32	8.5	3.9	37.5	18	31	2.5	1.7	4.1	18	32	160.6	58.6	447.7	22	39	4.6	2.9	9.7

Figure 17 further illustrates that the path distribution of the optimal dynamic routing policy remains steady above a certain *k* which depends on the OD 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 *k*=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 *k*=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 can be determined offline for each OD pair and trip start time combination.

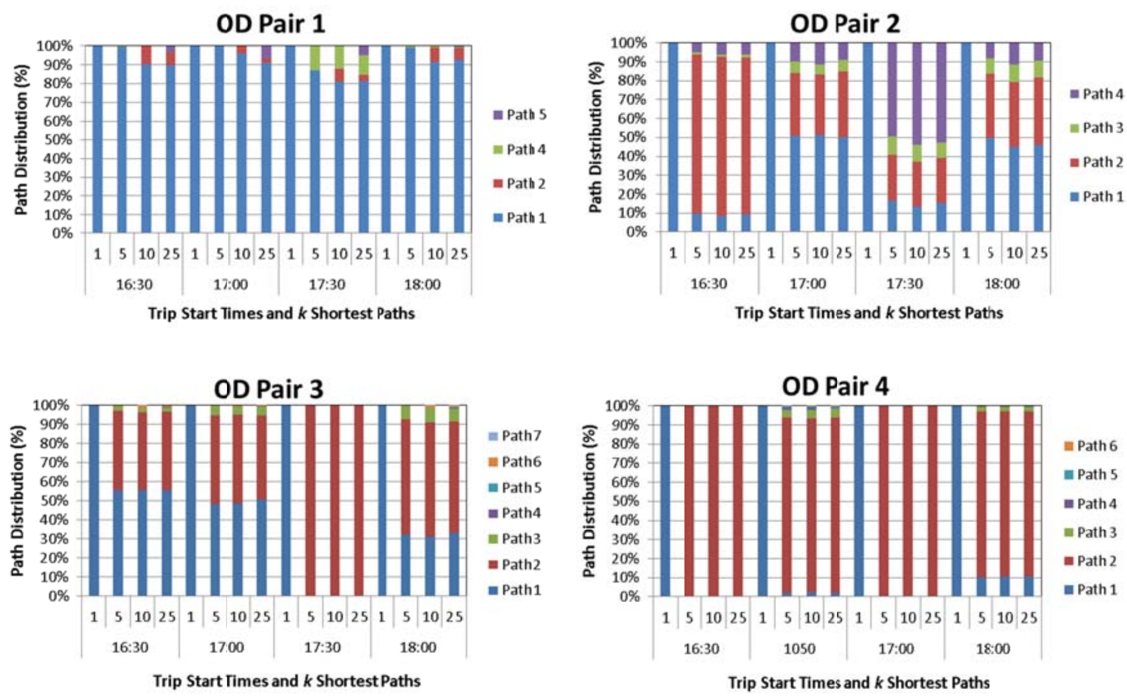

Fig. 17. Path distribution from dynamic routing for different start times and k-shortest paths.

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 stemming from incidents 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.

CHAPTER III: DYNAMIC ROUTING IN STOCHASTIC TIME-DEPENDENT NETWORKS FOR MILK-RUN TOURS WITH TIME WINDOWS UNDER ITS*

Abstract–We consider dynamic vehicle routing under milk-run tours with time windows in congested transportation networks for just-in-time (JIT) production. The arc travel times are considered stochastic and time-dependent. The problem integrates TSP with dynamic routing to find a static yet robust recurring tour of a given set of sites (i.e., DC and suppliers) while dynamically routing the vehicle between site visits. The static tour is motivated by the fact that tours cannot be changed on a regular basis (e.g., daily or even weekly) for milk-run pickup and delivery in routine JIT production. We allow network arcs to experience recurrent congestion, leading to stochastic and time-dependent travel times and requiring dynamic routing decisions. While the tour cannot be changed, we dynamically route the vehicle between pair of sites using real-time traffic information (e.g. speeds) from Intelligent Transportation System (ITS) sources to improve delivery performance. Traffic dynamics for individual arcs are modeled with congestion states and state transitions based on time-dependent Markov chains. Based on vehicle location, time of day, and current and projected network congestion states, we generate dynamic

* This chapter resulted in the following publication:
A.R. Guner, A. Murat, R.B. Chinnam: Dynamic Routing in Stochastic Time-Dependent Networks for Milk-Run Tours with Time Windows Under ITS. (Transportation Research Part E, Under Review, 2010)

routing policies for every pair of sites using a stochastic dynamic programming formulation. The dynamic routing policies are then simulated to find travel time distributions for each pair of sites. These time-dependent stochastic travel time distributions are used to build the robust recurring tour using an efficient stochastic forward dynamic programming formulation. Results are very promising when the algorithms are tested in a simulated network of Southeast-Michigan freeways using historical traffic data from the Michigan ITS Center and Traffic.com.

Keywords – Milk-run, transportation, congestion, dynamic routing, ITS, TSP with hard time windows

1. Introduction

Just-in-time (JIT) production requires frequent small-batch pickups and deliveries subject to fixed time windows. Since the shipments are usually less than truck load, the freight carrier planners develop milk-run tours (e.g., a visiting sequence of pickup and delivery sites). In a milk-run tour, for example, the vehicle departs from a distribution center (DC), picks up goods from several supplier sites, and returns to the DC for delivery. In planning milk-run tours, managers also consider *heijunka* (production smoothing or workload leveling) and *muda* (waste) philosophies of JIT production. Whereas the former can be achieved by equally spacing the delivery time windows over the suppliers' operating hours, the latter can be achieved by visiting the supplier sites at an optimal frequency, balancing transportation and inventory costs. The recurrent and non-recurrent congestion on road networks increase the travel time variability thus rendering it difficult to make delivery and pickup visits within the established time windows, which can be as narrow as 15 to 30 minutes [51, 52]. For carriers, as congestion worsens the costs related to travel time (e.g. labor and overtime costs) may outweigh other operating costs (e.g. vehicle miles traveled) [53].

For example, a survey in California found that 85% of trucking companies miss their time window schedules due to road network congestion. Furthermore, 78% of the managers surveyed stated that the time-window schedules for pickup and deliveries force their drivers to operate under congested road network conditions [54]. Some industries allow early or tardy delivery and/or pickups with a penalty (*soft* time windows). However, there are many practical settings (e.g., JIT production) with *hard* time windows where vehicles may pick up or deliver only during fixed times without exception [55].

In this paper, we address the problem of planning milk-run tours for JIT production subject to hard time windows in congested road networks. We model the milk-run tours as a Traveling Salesman Problem (TSP) with hard time windows. The road network congestion is represented through random network arc travel times and time-dependent congestion states.

The classical TSP is concerned with finding the least cost tour that visits each site exactly once given the set of sites. The travel between any pair of sites is a *path* which can be static (e.g., a fixed sequence of arcs) or can be determined through a dynamic policy. The cost of travel between pairs of sites can be measured in time, distance or a function of both, be deterministic or probabilistic, and be time-dependent or independent. In our problem setting, we consider a TSP with hard time windows under stochastic time-dependent (STD) arc travel times. All of the preceding TSP literature assumes that the path travel cost between pairs of sites is either deterministic or stochastic with a known probability distribution. In our network setting, the path travel times are both stochastic and time-dependent. We determine the distributions of these path travel times through optimal dynamic routing on network arcs using the real-time traffic information (e.g., speed data) available from the Intelligent Transportation System (ITS) sensor network.† In optimal dynamic routing between pairs of sites, we consider only the recurrent congestion (e.g., rush hour) and exclude the non-recurrent (e.g., traffic incidents and inclement weather). This is necessary since the milk-run TSP tours are established for longer periods where the recurrent congestion is more dominant. We model the recurrent congestion by defining congestion states of arcs based on historical ITS traffic data using Gaussian Mixture Model (GMM) based clustering [50]. The changes in arc congestion states represent the traffic dynamics and are modeled as Markov processes. Accordingly, the optimal dynamic routing problem is then cast as a Markov decision process (MDP) where the states space consists of the position of the vehicle, the time of the day, and the current and projected congestion states of arcs with limited look ahead (examining the state of the full network is

† According to Research and Innovative Technology Administration (RITA) of U.S. Department of Transportation (US DOT), "Intelligent transportation systems (ITS) encompass a broad range of wireless and wire line communications-based information and electronics technologies. When integrated into the transportation system's infrastructure, and in vehicles themselves, these technologies relieve congestion, improve safety and enhance productivity." ITS technology and coverage is expanding quickly in the U.S. and is widely used in many developed and developing nations around the world. For more information about U.S. ITS, see: http://www.its.dot.gov.

computationally prohibitive and even unnecessary, see [26]). We identify the paths' optimal dynamic routing policies (DRP) by solving a stochastic dynamic programming formulation for each pair of sites.

By simulating the optimal DRPs, we estimate the travel time distributions between every pair of sites. We then use these distributions to determine the optimal TSP tour by solving a stochastic dynamic programming formulation for TSP. Since the travel times are STD, we employ the convolution approach in Chang et al. [4] to estimate the distribution of site arrival times for pickup and delivery. Whereas the routes between pairs of sites are dynamic, the TSP tour is static. This is because, in JIT production systems, the tours for pickups and deliveries support such objectives as production smoothing and workload leveling and remain fixed for extended periods (e.g., months). The optimal TSP tour can be obtained by minimizing the mean criteria combination (e.g., travel time, mileage, and truck utilization) or a mean-variance objective which also accounts for the variability of criteria. Although our methodology could have accommodated a wide range of these objectives, we selected a mean-variance objective based on the trip time which accounts for the transportation cost and service level (i.e., on-time performance) trade-offs in JIT production systems. We defined the most robust TSP tour as the tour with minimum trip time mean-variance objective.

The contribution of this study is three-fold. First, we developed an integrated methodology for identifying the TSP tours of sites in STD networks where the stochastic path travel times between pairs of pickup and delivery sites are estimated through optimal dynamic routing. Second, we proposed an approach for dynamic routing between pairs of sites in STD networks using the real-time congestion information available from ITS sensor networks. Third, using a real network and data, we simulated the results of the proposed integrated approach and demonstrate the transportation cost and delivery service level improvement based on optimal dynamic routing between sites.

The rest of the paper is organized as follows. A selective survey of the related literature is given in Section 2. In Section 3, the modeling of the stochastic time-dependent TSP is described. Section 4 presents the experimental results of a case study application to show the effectiveness of the proposed approach. Section 5 concludes the study and suggests directions for future research.

2. Literature Survey

In JIT production systems, the pickup and delivery tours are constructed while accounting for logistics drivers such as leveling the workload and decreasing inventory levels. One approach for determining pickup and delivery tours in JIT systems is the common frequency routing (CFR) method, where the suppliers are grouped into subsets and each subset of suppliers is served in a single tour [56]. The CFR method considers scheduling and routing decisions jointly while accounting for transportation and inventory costs. For computational tractability, the CFR method assumes fixed routes and identical visit frequency for suppliers in the same subset. Another approach is the generalized frequency routing (GFR) where a supplier's visit frequency is not required to be the same as other suppliers in the subset [57]. One of the goals in scheduling and routing decisions is to achieve production smoothing through uniformly spaced pickup and delivery visits. These “lean” routing studies consider a more general problem (e.g., VRP) than the TSP studied in this paper but assume that the travel times on the transportation network are deterministic and time-independent. Accordingly, our focus was on selecting robust tours for a given subset of suppliers with uniformly spaced hard time windows.

The body of literature to which this study is related is the stochastic time-dependent traveling salesman problem (TSP) with time windows. In the classical TSP, given a set of sites and the cost matrix relating pairs of sites, the goal was to find the shortest tour starting from the origin site, visiting each site exactly once, and returning to the origin site. TSP and its generalization VRP have been studied for more than five decades and a wide variety of exact and heuristic algorithms have been developed [58-61]. There are many variants of the classical TSP but we restricted our review to those studies with time-dependent and stochastic travel times. Malandraki and Dial [62] presented a dynamic programming (DP) procedure and a “restricted” DP procedure that uses the nearest-neighbor heuristic approach to solve the time-dependent TSP (TD-TSP). They modeled the time dependency by discrete step functions such that the planning horizon had a number of different time zones and the travel times differed only at different time zones. [63] recognized the limitation of using such step functions which violates the first-in-first-out (FIFO) principle by causing a later departure time leading to an earlier arrival time if steep speed increases occur. Accordingly, they emphasized the need to explicitly model time-dependent travel times and proposed a model to determine TSP tours in compliance with the FIFO principle.

Another variant of the classical TSP is the TSP with stochastic travel times between sites. This variant is most studied in the more general form of the vehicle routing problem [64, 65]. [4, 66] studied the stochastic time-dependent TSP with time windows (STD-TSP-TW). [66] solved the TSP through a dynamic programming approach applied to a reduced state space. They employed two-state space reduction strategies to reduce the computational complexity. Initially they estimated the mean and variance of the arrival time of the vehicle at each site based on the first (or second) order Taylor approximation. In the first strategy, they defined a service level based on the arrival times to sites and eliminated routes that did not satisfy those service levels. The other strategy eliminates states based on expected travel times. [4] developed a convolution–propagation approach (CPA) to estimate the mean and variance of arrival times at sites assuming the arc travel times are normally distributed. They proposed a heuristic algorithm that uses the *n*-path relaxation of deterministic TSP in [67] to solve the problem. Although the TSP problem we considered is similar to those in [4, 66], the travel time distributions between pairs of sites were endogenous in our study. In particular, we integrated the construction of a TSP tour among sites with the road network routing between pairs of sites in the TSP tour. The dynamic routing between sites accounts for the time-dependent stochastic congestion states by using real-time traffic information and by anticipating congestion states with limited look ahead. To the best of our knowledge, there is no prior study proposing and integrating dynamic routing between sites for the stochastic time-dependent TSP problem. In addition, whereas [4, 66] identified tour(s) with least expected tour times, we selected tour(s) with minimum mean-variance objective of the trip times.

Dynamic routing and modeling real time information has mostly been studied in shortest path problem literature. [23] conducted the first study to consider the stochastic temporal dependence of arc costs and suggested using online information en route. They defined the environmental 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 dynamic programming procedure to determine the optimal DRP. [25] studied a similar problem as did [22] except that the information of all of the arcs was available in real-time. They proposed a dynamic programming (DP) formulation where the state space included the states of all arcs, time, and the current node. They noted that the state space of the proposed formulation became quite large making the problem intractable. They reported substantial cost savings in a computational study based on a Southeast-Michigan road network. To address the intractable state-space issue, [26] proposed state space reduction methods. A limitation of [25] is the modeling and partitioning of travel speeds for the determination of arc congestion states. They assumed that the joint distribution of velocities from any two consecutive periods followed a single unimodal Gaussian distribution, which did not adequately represent arc travel velocities for arcs that routinely experience multiple congestion states. Moreover, they also employed a fixed velocity threshold *(50 mph)* for all arcs and for all times in partitioning the Gaussian distribution to estimate state-transition probabilities (i.e., transitions between congested and uncongested states). As a result, the value of real-time information was compromised rendering the loss of performance of the DRP. Our dynamic routing approach addressed all of these limitations. The detailed steps of our model are described in the Section 3.1.

3. STD-TSP with Dynamic Routing

The *STD-TSP with dynamic routing problem* is to find a tour of a given set of sites (i.e., DC and supplier) while dynamically routing between sites' visits on a STD network to meet the time windows requirements. It differs from the TSP with stochastic travel times in that the travel time distributions are obtained through dynamic routing on the road network and thus are dependent on the site departure times. We selected the tours based on a robust tour objective. This robust tour objective captured the tradeoff between transportation efficiency and on-time delivery service level.

We used a sequential method to select the robust tour. First, we first determined the travel time distributions between every pair of sites. Second, we found and selected the tour minimizing the mean-variance objective of the trip time. The travel time distributions between sites were estimated through the following steps (See Section 3.1.):

- Develop a dynamic routing policy between every pair of sites.
- Estimate the travel time distribution through simulation for every possible departure times.

Once the travel time distributions were estimated for every pair of sites at different departure times, we then employed a stochastic time-dependent dynamic programming (STD-DP) to select the robust tour (Section 3.2.).

3.1. Dynamic Routing with Real-time Traffic Information

Let $G=(N,A)$ be a directed graph in which N is the set of nodes and $A\subseteq N\times N$ is the set of directed arcs. The (decision) node $n\in N$ represents an intersection where the driver can decide which arc to select next. A directed arc is represented by an ordered pair of nodes $(n,n')\in A$ in which n is called the origin and n' is called the destination of the arc. Given an origin-destination (OD) node pair of sites (DC, supplier), the dynamic routing problem is to decide which arc to choose at each decision node such that the expected total OD travel time is minimized. We denote the origin and destination nodes with n_0 and n_d, 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. We first describe the modeling of recurrent congestion and then present the stochastic dynamic programming formulation and solution approach.

3.1.1. Congestion Modeling

A directed arc $(n,n')\in A$ is labeled as observed if its real-time traffic data (e.g., velocity) is available through the ITS. An observed arc can be in $r+1\in Z^{+}$ different states that represent the arc's traffic congestion level at a given time. Let $s_a(t)$ be the congestion state of arc a at time period t, i.e. $s_a(t) = \{\text{Congested at level } i\} = \{i\}$ for $i = 1, 2, ..., r+1$ and be determined as follows:

$$s_a(t) = \left\{ i, \text{if } c_a^{i-1}(t) \le v_a(t) < c_a^i(t) \right\} \tag{1}$$

where $c_a^i(t)$ denote the cut-off velocity at level *i*. 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 $s_a(t) = \{\text{Congested}\} = \{1\}$.

We assume that 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., $S(t) = \left\{ s_1(t), s_2(t), ..., s_{|A|}(t) \right\}$ 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 $s(t)$. We assume that arc states are independent from each other and have the single-stage Markovian property. To estimate the state transitions for each arc, we jointly model the velocities of two consecutive periods Accordingly, the time-dependent single-period state transition probability from state $s_a(t) = i$ to state $s_a(t+1) = j$ is denoted by $P\left\{ s_a(t+1) = j \mid s_a(t) = i \right\} = \alpha_a^{ij}(t)$. We estimate the transition probability for arc *a*, $\alpha_a^{ij}(t)$ from the joint velocity distribution as follows:

$$\alpha_a^{ij}(t) = \frac{\left| c_a^{i-1}(t) \le V_a(t) < c_a^i(t) \cap c_a^{j-1}(t+1) < V_a(t+1) < c_a^j(t+1) \right|}{\left| c_a^{i-1}(t) \le V_a(t) < c_a^i(t) \right|} \tag{2}$$

where the |*e*| operator corresponds to the frequency count of event *e*. Let $TP_a(t,t+1)$ denote the matrix of state transition probabilities from time t to time $t+1$, then, we have $TP_a(t,t+1)=\left[\alpha_a^{ij}(t)\right]_{ij}$. Note that the single-stage Markovian assumption is not restrictive in our approach as we could extend our methods to the multi-stage case by expanding the state space [48]. Let the network be in state $S(t)$ at time t, and we want to find the probability of the network state $S(t+\delta)$, where δ is a positive integer number. Given the independence assumption of the arcs' congestion states, this can be formulated as follows:

$$P\left(S(t+\delta)\,|\,S(t)\right)=\prod_{a=1}^{|A|}P\left(s_a(t+\delta)\,|\,s_a(t)\right) \tag{3}$$

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

$$TP_a(t,t+\delta)=\left[\alpha_a^{ij}(t)\right]_{ij}\times\left[\alpha_a^{ij}(t+1)\right]_{ij}\times\ldots\times\left[\alpha_a^{ij}(t+\delta)\right]_{ij} \tag{4}$$

We assume that the distribution of an arc travel time is Gaussian. We further assume that the arc 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:

$$\delta(t,a,s_a)\sim N\left(\mu(t,a,s_a),\sigma^2(t,a,s_a)\right) \tag{5}$$

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 $s_a(t)$. For clarity of notation, we hereafter suppress the arc label from the parameter space wherever it is obvious, i.e. $\delta(t,a,s_a)$ will be referred as $\delta_a(t,s)$.

3.1.2. DP Formulation for Dynamic Routing

The objective of the dynamic routing algorithm is to minimize the expected travel time based on real-time information such as the path originates at node n_0 and ends at node n_d. Let us assume that there is a feasible path between (n_0,n_d) where a path $p=(n_0,..,n_k,..,n_{K-1})$ is defined as the sequence of (decision) nodes such that $a_k \equiv (n_k,n_{k+1}) \in A$, $k=0,..,K-1$ 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 arcs set of node n_k, 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^{th} decision stage, t_k is the time at k^{th} decision stage where $t_k \in \{1,...,T\}$ $T > t_{K-1}$. T is an arbitrarily large number and is used to limit the planning horizon for modeling purposes. Note that we are discretizing the planning horizon.

While the optimal dynamic routing policy requires real-time consideration and projection of the traffic states of the complete network, this approach renders 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 from the long run average steady state probabilities of the arc congestion states. Hence, an efficient but practical approach would trade off the degree of look-ahead (e.g., the number of arcs to monitor) with the resulting projection accuracy and routing performance. This has been very well illustrated in [26]. Thus, we limit our look-ahead to a 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, the value of real-time information, and the congestion state transition characteristics of the arcs. 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 of the arcs in the network. $ScAS(a_k)$, the successor arc set of arcs 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 nodes (n_{k+2}) of arcs a_{k+1}.

Since the total path 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 $\left(n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k}\right)$ of the system at the k^{th} decision stage is denoted by Ω_k. This state vector is composed of the state of the vehicle and network and thus is 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 Ω_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 based on the remaining expected travel time. The expected travel time at a given node with the selection of an outgoing arc is the summation of expected arc travel time on the arc chosen and the expected travel time of the next node. Let $\boldsymbol{\pi}_{n_0 n_d} = \{\pi_0, \pi_1, ..., \pi_{K-1}\}$ be the dynamic routing policy (DRP) of the trip that is composed of policies for each of the *K*-1 decision stages. For a given state $\Omega_k = \left(n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k}\right)$, 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 π is as follows:

$$F^{\pi}(\Omega_0) = \underset{\delta_k}{E}\left\{\sum_{k=0}^{K-2} g(\Omega_k, \pi_k(\Omega_k), \delta_k) + \bar{g}(\Omega_{K-1})\right\} \tag{6}$$

where $\Omega_0 = (n_0, t_0, S_0)$ is the starting state of the system. δ_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 (ϕ) 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 Ω_{K-1}. Then, the minimum expected travel time can be found by minimizing $F(\Omega_0)$ over the policy vector π as follows:

$$F^*(\Omega_0) = \min_{\boldsymbol{\pi}_{n_0 n_d} = \{\pi_0, \pi_1, ..., \pi_{K-1}\}} F(\Omega_0) \tag{7}$$

The corresponding optimal policy is then:

$$\boldsymbol{\pi}^*_{n_0 n_d} = \underset{\boldsymbol{\pi}_{n_0 n_d} = \{\pi_0, \pi_1, ..., \pi_{K-1}\}}{\arg\min} F(\Omega_0) \tag{8}$$

Hence, the Bellman's cost-to-go equation for the dynamic programming model can be expressed as follows (Bertsekas, 2001):

$$F^*(\Omega_k) = \min_{\pi_k} \mathop{E}_{\delta_k} \left\{ g(\Omega_k, \pi_k(\Omega_k), \delta_k) + F^*(\Omega_{k+1}) \right\} \tag{9}$$

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:

$$F(\Omega_k \mid a_k) = \sum_{\delta_k} P(\delta_k \mid \Omega_k, a_k) \left[g(\Omega_k, a_k, \delta_k) + \sum_{s_{a_{k+1},k+1}} P\left(s_{a_{k+1},k+1}(t_{k+1}) \mid s_{a_{k+1},k}(t_k)\right) \right. \\ \left. \sum_{s_{a_{k+2},k+1}} P\left(s_{a_{k+2},k+1}(t_{k+1})\right) F(\Omega_{k+1}) \right] \tag{10}$$

where $P(\delta_k \mid \Omega_k, a_k)$ is the probability of travelling arc a_k in δ_k periods. $P\left(s_{a_{k+2},k+1}(t_{k+1})\right)$ 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 used the backward dynamic programming algorithm to solve $F^*(\Omega_k)$, $k = K-1, K-2, .., 0$. 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 the destination node, and $F(\Omega_{K-1}) = 0$ if $t_{K-1} \le T$. Accordingly, a penalty cost is accrued whenever there is delivery tardiness, e.g., $t_{K-1} > T$. Note that $s_{K-1} = \varnothing$, since the destination node does not have any current and successor arc states, e.g. the travel terminates at the destination node.

3.1.3. Estimating Travel Time Distributions between Sites

Given a pair of sites (DC, supplier), origin $j \in M$ and destination $k \in M$, we solve the dynamic programming formulation in preceding section for all feasible departure times from *j* and obtain the optimal routing policy, $\boldsymbol{\pi}_{jk}$, for each departure time alternative. Next, for each departure time alternative (t_j), we sample a congestion state $s(t_j)$ for current and successor arcs of *j*, and simulate the policy corresponding to the sample state $\Omega = \left(j, t_j, s(t_j)\right)$. Note that the sampling probabilities of the congestion state $s(t_j)$ are based on the steady-state probabilities of the states of current and successor arcs of *j*. Following sufficient sampling for t_j, we estimate the distribution of the mean travel times obtained by simulating corresponding policies for each sampled state Ω. We then calculate the expectation and variance of travel time from *j* to *k* at time t_j and respectively denote them with $E\left[\delta_{jk}\left(t_j\right)\right]$ and $Var\left(\delta_{jk}\left(t_j\right)\right)$. Note that, with slight abuse of notation, $\delta_{jk}\left(t_j\right)$ corresponds to the random travel time between *j* and *k* departing at t_j.

3.2. Dynamic Programming for STD-TSP

In this section, we describe the stochastic time-dependent dynamic programming (STD-DP) approach for selecting a robust tour of a given set of sites (i.e., DC and supplier) while dynamically routing between sites' visits to meet the time windows requirements. The time window requirements are strict (e.g., hard time windows) and each site has a deterministic service time for loading/unloading. This STD-DP approach integrates and builds on the results of earlier studies. Specifically it integrates the stochastic tour search procedure from [62] [66] and the convolution idea from [4]. However, the proposed STD-DP approach uses the travel time distributions obtained in the preceding section by dynamically routing on the road network. Further, the approach selects the most robust tour by trading off the expected duration of the tour with its variability as follows:

$$TC_{00} = E\left[T\left(M,\tau,0\right)\right] + b\sqrt{Var\left(T\left(M,\tau,0\right)\right)} \qquad (11)$$

where, τ is the TSP tour, $E\left[T\left(M,\tau,0\right)\right]$ and $Var\left(T\left(M,\tau,0\right)\right)$ are the expected and variance of the round trip duration departing from site *0* (DC) at time t_0, visiting all sites in M once, and returning back to site *0* (DC); b is a user defined risk-parameter for balancing the transportation efficiency with on-time delivery performance.

We first describe the STD-DP approach without the time-windows and present its extension to time window case in Section 3.2.1. There are *m*-1 sites (other than the DC, assuming the vehicle at the DC) to be visited, represented by nodes $1,\ldots,m-1 \in M$. Let $\left(C,k\right) \subseteq M/\{0\}$ be an unordered set of visited sites where $k \in C$ is the last visited site. Define *partial tour* τ as a tour that starts from the DC, visits all sites in $\left(C,k\right)$ only once and ends the tour at site k. Note that there may be more than one partial tour corresponding to set $\left(C,k\right)$ and we denote the set of partial tours with $\tau \in \Gamma\left(C,k\right)$. For brevity, we do not repeat the membership of partial tours in the remainder and assume $\left(C,\tau,k\right)$ implies $\tau \in \Gamma\left(C,k\right)$. Let $T\left(C,\tau,k\right)$ be the random variable of arrival time at site k taking the partial tour τ of set $\left(C,k\right)$ after departing site 0 at time t_0. Let also $E\left[T(C,\tau,k)\right]$ and $Var\left(T(C,\tau,k)\right)$ be the mean and variance of arrival time to site *k*, $T(C,k)$ after taking the partial tour τ, respectively.

Step 1. Initialize: For all $|(C,k)|=1$ where $(C,k)=\{k\}, k\in M/\{0\}$, we initialize $E\left[T(C,\tau,k)\right]=T(0)+s_0+E\left[\delta_{0k}(t_0)\right]$ and $Var\left(T(C,\tau,k)\right)=Var\left(\delta_{0k}(t_0)\right)$, where $T(0)$ is the arrival time to the site *0* (DC), s_0 is the service (e.g., loading/unloading) time at the site 0, and $E\left[\delta_{0k}(t_0)\right]$ is the expected travel time from site 0 to site k as a function of the departure time, t_0. Note that the expectation $E\left[\delta_{0k}(t_0)\right]$ is over the congestion states of current and successor arcs of site *0.*

Step 2. Main: For all $|(C,k)|>1$, there are partial tours of set (C,k), where we visit k, $k\in M/\{0,j\}$ immediately after j $\left(\text{for all } j\in C/\{k\}\right)$. The mean and variance $T(C,\tau,k)$ for the partial tour τ is calculated through the following convolution propagation approach adapted from [4] :

$$E\left[T(C,\tau,k)\right]=E\left[T(C,\tau,j)\right]+s_j+\sum_{t_j}E\left[\delta_{jk}(t_j)\right]p_{t_j} \tag{12}$$

$$\begin{aligned} Var\left(T(C,\tau,k)\right)=&Var\left(T(C,\tau,j)\right)+\sum_{t_j}p_{t_j}\sigma_{t_j}^2+\sum_{t_j}p_{t_j}E\left[\delta_{jk}(t_j)\right]^2-\left[\sum_{t_j}p_{t_j}E\left[\delta_{jk}(t_j)\right]\right]^2 \\ &-2\sum_{t_j}E\left[\delta_{jk}(t_j)\right]\sqrt{Var\left(T(C,\tau,j)\right)}\left(\varphi_{z_{t_j}}-\varphi_{z_{t_j-1}}\right), \end{aligned} \tag{13}$$

where s_j is the deterministic service time at site j; $\delta_{jk}(t_j)$ is the travel time from site j to site k at the departure time $t_j=T(C,\tau,j)+s_j$; p_{t_j} is the probability of departing at time t_j from node j. Note that the expectation $E\left[\delta_{jk}(t_j)\right]$ is over the congestion states of current and successor arcs of site *j*. Let $z_{t_j} = \frac{t_j - E[T(C,\tau,j)] - s_j}{\sqrt{Var(T(C,\tau,j))}}$, we calculate p_{t_j} as $p_{t_j} = \Phi(z_{t_j}) - \Phi(z_{t_j - 1})$, where $\varphi(\cdot)$ and $\Phi(\cdot)$ are the density and cumulative distribution functions of the standard normal distribution, respectively. Once $T(C,\tau,k)$ is calculated for all $|(C,k)| > 1$, we decrease the number of partial tours under investigation by performing the following partial tour elimination test adapted from [66].

Dominancy test: There may be more than one *partial tour* for a set (C,k). Let us assume (C,τ_1,k) and (C,τ_2,k) are two partial tours of set (C,k) that cover same sites. We eliminate the partial tour (C,τ_1,k) if $T(C,\tau_2,k)$ dominates $T(C,\tau_1,k)$, e.g., $E[T(C,\tau_2,k)] \le E[T(C,\tau_1,k)]$ and $Var(T(C,\tau_2,k)) \le Var(T(C,\tau_1,k))$.

We note that additional partial tour elimination tests based on time windows are described in the next section. After testing all pairs of partial tours, we repeat the main step until $C = M - \{0\}$.

Step 3. Termination: To complete the tour at the site *0* (DC), we set *k*=0 and perform the main step one last time and obtain the expectation and variance of the total tour time $T(C,\tau,0)$ for all remaining tours τ of $(C,0)$ where *C*=*M*. We calculate the total tour cost as $TC_{00} = E[T(C,\tau,0)] + b\sqrt{Var(T(C,\tau,0))}$ for each of the remaining tours. We select the tour with minimum cost as the robust tour solution.

3.2.1. STD-TSP with Time Windows

In the preceding section, we presented STD-DP for solving the STD-TSP without time windows. This section extends it to cases with hard time windows. When there is a time window requirement at a site, there are three possible arrival scenarios to that site with regard to the time window: early, late, and on-time arrival. In our model, we allow early arrivals, if earliness is not greater than a pre-specified value, by requiring the vehicle to wait until the beginning of time window. In comparison, we do not allow late arrivals by eliminating those partial tours with the possibility of tardiness greater than a pre-specified probability.

Let us assume the vehicle arrives at site j with a random arrival time of $T(C,\tau,j)$ with partial tour τ and does not violate any time window requirement. Let (e_j,l_j) be the time window at site j, where e_j is the earliest time and l_j is the latest time to start service at site j.

- *Early Arrival:* The vehicle arrival is assumed to be early if probability of arriving later than e_j is less than the early arrival probability $\underline{\gamma}$: $P\left(T(C,\tau,j)\geq e_j\right)\leq\underline{\gamma}$. The vehicle can wait only if $T(C,\tau,j)\geq\left(e_j-\varepsilon\right)$, where ε is maximum allowable waiting time at the site; otherwise the vehicle is assumed to be *too early* and the partial tour is then discarded. Note that if a particular vehicle arrival is accepted, then, the start time to service is $\max\left(T(C,\tau,j),e_j\right)$.
- *Late Arrival:* The vehicle arrival is assumed to be late and the partial tour is discarded if probability of arriving later than l_j is greater than the maximum allowable tardiness probability $\overline{\gamma}$: $P\left(T(C,\tau,j)\geq l_j\right)>\overline{\gamma}$.

- *On-time Arrival:* The vehicle arrival is assumed to be on-time and is accepted if both $P\left(T(C,\tau,j)\geq e_j\right)>\underline{\gamma}$ and $P\left(T(C,\tau,j)\geq l_j\right)\leq\overline{\gamma}$.

Given these definitions, $E\left[T(C,\tau,j)\right]$ and $Var\left(T(C,\tau,j)\right)$ in equation (12) and (13) can be calculated with the following formulas:

$$E\left[T(C,\tau,j)\right]=E\left[\max\left(T(C,\tau,j),e_j\right)\right]+s_j \tag{14}$$

$$Var\left(T(C,\tau,j)\right)=E\left[\max\left(T(C,\tau,j),e_j\right)^2\right]-E^2\left[\max\left(T(C,\tau,j),e_j\right)\right] \tag{15}$$

Note that the maximization operator is due to the waiting upon early arrival. For late arrivals, the maximum operator in (14) and (15) does not exist since there is no waiting with late arrivals. In both early and late arrival cases, we eliminate those partial tours according to the corresponding pre-defined parameters ($\underline{\gamma},\varepsilon,\overline{\gamma}$). Note that, different than the stochastic dominance elimination, time window eliminations are used in the initialization step and at the termination step if there are also DC time windows applicable to the tour completion time.

3.2.1.1. Determining Time Windows for a Given Tour

In the preceding section, we described how the STD-DP approach is extended for problems with hard-time windows. In most JIT production systems, the time window requirements affect different parties differently. For instance, the carriers are penalized for late deliveries either by charges associated with contracted service levels or by their reduced ranking as a transportation service supplier. In comparison, early arrivals correspond to lower utilization of assets and drivers. The suppliers (pickup sites), on the other hand, need to stock more safety inventory and allocate more material handling resources if time windows are relaxed (e.g., width of the window is increased). The width of the time windows and their positioning constitute two features of most logistics contracts and are often re-adjusted due to changing production volumes and routes. The time window setting process differs from industry to industry. In JIT environments, it is common that the time windows are set by trucking and/or manufacturer companies according to JIT principles and are usually accepted by the suppliers as part of the sourcing contract. In such a setting, the trucks visit the supplier sites several times per day subject to the tight time windows spaced as much evenly as possible within the supplier's operating hours (even spacing is generally key to supplier efficiency; reduces finished goods inventory levels).

We now describe a procedure for carriers to position the time windows such that the on-time delivery performance is improved. We assume that the width of time windows (w) is determined beforehand by the supplier and manufacturer and they are indifferent to the positioning of the time windows as long as they are uniformly distributed during delivery horizon. The procedure uses the result that the site arrival times follow Gaussian distribution when the arc travel times are also Gaussian (Chang et al., 2010). Therefore, centering the time windows at the expected site arrival times maximizes the on-time delivery performance, if, there is no waiting allowed at the site for early deliveries. This is indeed the case practiced by carriers even if there is some flexibility in early arrival acceptance. Let τ be the selected ordered tour that starts from DC, visits all sites once, and ends at DC. Further let τ_k be the partial tour of τ ending at site *k*. Accordingly, $T(C,\tau_k,k)$ is the random variable of arrival time at site k by following the partial tour τ_k. Let also $E\left[T\left(C,\tau_k,k\right)\right]$ and $Var\left(T\left(C,\tau_k,k\right)\right)$ be the mean and variance of arrival time $T\left(C,\tau_k,k\right)$, respectively.

Procedure for Setting Time Windows:

For $k=1,\ldots,m-1$, Repeat:

If k=1,

$E\left[T\left(C,\tau_k,k\right)\right]=T(0)+s_0+\delta_{0k}\left(t_0\right)$ and $Var\left(T\left(C,\tau_k,k\right)\right)=Var\left(\delta_{0k}\left(t_0\right)\right)$ where $T(0)$ is the arrival time to the site 0 (DC), s_0 is the service time at the site 0, and $\delta_{0k}\left(t_0\right)$ is the random travel time from site 0 to site k as a function of the departure time, t_0.

Else,

Assume visiting k immediately after j and look up the updated $E\left[T\left(C,\tau_j,j\right)\right]$ from the previous step. Calculate $E\left[T\left(C,\tau_k,k\right)\right]$ from (11).

End.

Set $e_k=E\left[T\left(C,\tau_k,k\right)\right]-w/2$ and $l_k=E\left[T\left(C,\tau_k,k\right)\right]+w/2$.

Update $E\left[T\left(C,\tau_k,k\right)\right]$ and $Var\left(T\left(C,\tau_k,k\right)\right)$ according to equations (14) and (15).

Return.

The above procedure is an iterative procedure where we visit sites according to the tour τ and set time windows for each site one at a time. At each site, we calculate the expected arrival time to that site based on the time windows set at the previously visited sites. We account for the previously set time windows because they affect the site arrival time of the subsequent visited sites through the waiting at early arrivals. Note that the centered placement of time windows is an assumption. It is possible to shift the time windows to the right of the center (expected site arrival time) such that the likelihood of late arrivals decreases. Clearly, this modification is contingent upon the maximum allowable waiting time imposed for early arrivals. In the case of unrestricted waiting, it can be shown that, by shifting the time window to right, one can turn time window constraints into redundant constraints.

4. Experimental Study

In this section, we test the proposed methodology on a real case study application using the road network from Southeast Michigan, U.S.A. (Fig. 1). We consider an automotive JIT production system where an OEM's DC is replenished by milk-run pickup and deliveries from multiple suppliers. The case study road network covers major freeways and highways in and around the Detroit metropolitan area. The network has 140 nodes and a total of 492 arcs with 140 observed arcs and 352 unobserved arcs. Real-time traffic data for the observed arcs is collected by the Michigan ITS Center and Traffic.com. In this application, we used data from 66 weekdays of May, June, and July 2009, for the full 24 hours of each day. The raw speed data was aggregated at a resolution of 5 minute intervals. For the experimentation, we increased the resolution of data to one data-point per minute through linear interpolation (see [25]). Since the collected speed data is averaged across different vehicle classes (i.e., automobile, trucks) and no data was available for individual classes of vehicles, we assumed that the truck being routed could also cruise at the collected average speeds. We implemented all of our algorithms and methods in Matlab 7 and executed them on a Pentium IV machine (with CPU 1.6 GHz and 1024 MB RAM) running Microsoft Windows XP operating system.

Our experimental study is outlined as follows: Section 4.1 describes the estimation and modeling process for recurrent congestion and illustrates through a sample arc of the network. Section 4.2 explains the steps of generating DRPs and estimating travel time distributions between sites. Section 4.3 presents experimental results of identifying and selecting robust STD-TSP tours without time windows and reports savings from employing the dynamic routing policy over the static routing policy between pair of sites. Section 4.4 evaluates the performance of routing policies identified in Section 4.3 after setting the sites' time windows as described in Section 3.2.1.1.

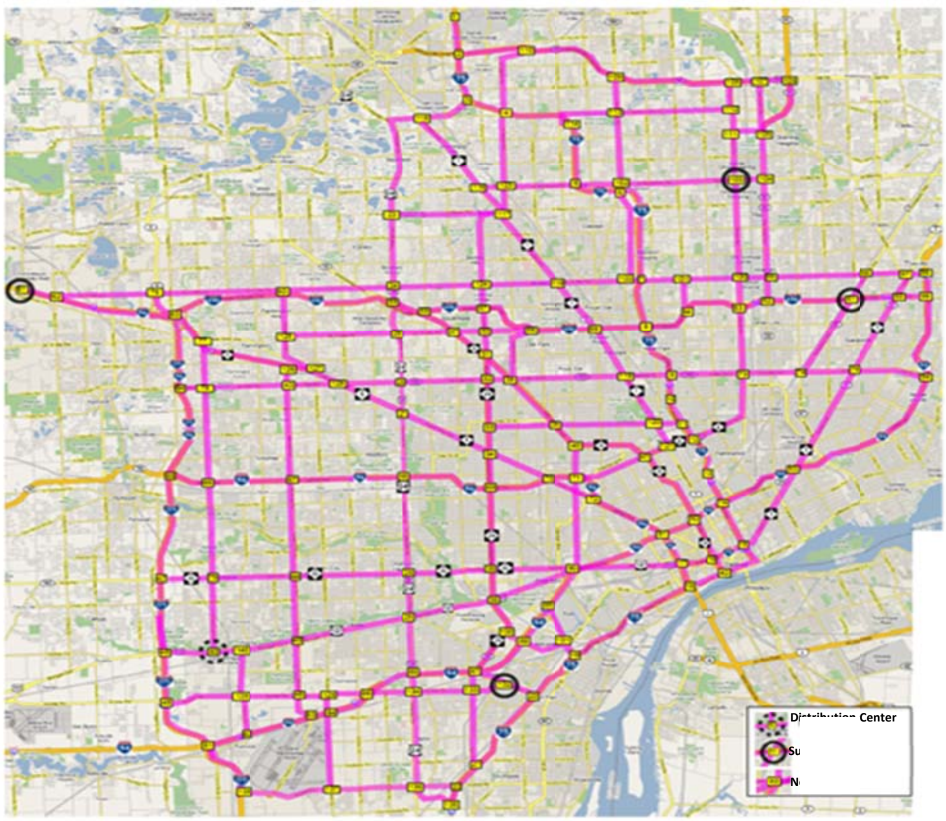

Fig. 1 Southeast Michigan road network considered for experimental study.

4.1. Estimating Congestion States

The proposed dynamic routing algorithm calls for identification of different congestion states, estimation of their state transition rates, and estimation of arc traverse times by time of the day. To better illustrate the modeling of congestion states, we present the data and congestion state identification and separation procedures for an example arc (7, 8). The speed data for arc (7, 8) for the weekdays is illustrated in Fig. 2a. The mean and standard deviations of speed for the arc (7, 8) are plotted in (Fig. 2b). From Fig. 2a and Fig. 2b, it can be clearly seen that the traffic speeds follow a non-stationary distribution that vary highly with 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 the day. In particular, we used the greedy learning GMM clustering method of [50] for its computational efficiency and performance. After obtaining the state clusters for each time interval *t*, we first estimate the time-dependent cut-off speeds if GMM yields more than one congestion state at *t.* Next, given cut-off speeds, we then estimate the parameters of the Gaussian distributions for state transitions for congestion state *i* from *t* to *t+1* for all *t,* i.e., $(\boldsymbol{\mu}^{i}_{t,t+1};\boldsymbol{\Sigma}^{i}_{t,t+1})$. Applying GMM for arc (7,8), for instance, recommended two clusters of congestion states for almost all time intervals except few. Fig. 3a illustrates the transition rates for arc (7, 8) with a 15 minute time interval resolution during the day. Note that, we are using two clusters for arc (7, 8) in all time intervals for presentation purpose (other than increasing computational burden, there are no other consequences). In Fig. 3a, the α_t denotes the probability of state transition from congested state to congested state and β_t denotes the probability of state transition from uncongested state to uncongested state. The mean travel time of arc (7, 8) for congested and uncongested traffic states is given in Fig. 3b.

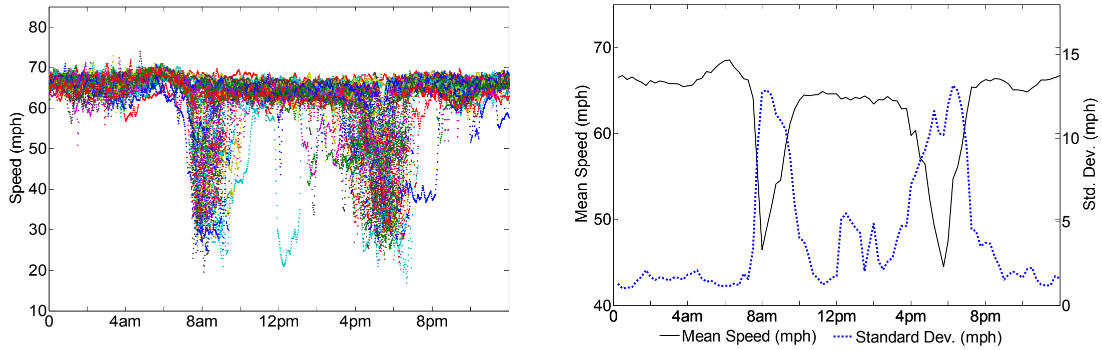

Fig. 2 For arc (7,8) (a) raw traffic speeds for May, June, and July 2009 weekdays (b) mean (mph) and standard deviations (mph) of speeds by time of the day with time interval resolution of 15 minutes.

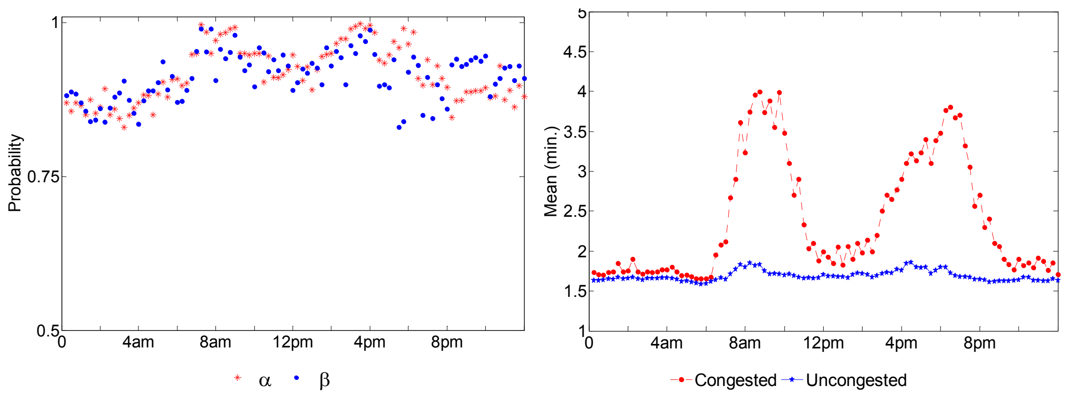

Fig. 3 For arc (7, 8) (a) congestion state-transition probabilities: α, congested to congested transition; β, uncongested to uncongested transition probability (b) mean travel time(min.) for congested and uncongested congestion states.

4.2. Estimating Travel Time Distributions between Sites

Using the previous section's results, e.g., time and congestion state dependent distribution of arc travel times and congestion state transition probabilities, we employed the dynamic routing algorithm in Section 3.1.2 to determine the dynamic routing policy $\boldsymbol{\pi}_{jk}$ between every pair of customer sites (j,k) at different departure times. Next, we estimate the travel time distribution between every pair of sites. This can be achieved by simulating the optimal dynamic policies in two different ways: using estimated arc travel time distributions as described in Section 3.1.2. or using the available historical data for 66 weekdays. We choose to use the historical data because of the link interactions and dependencies not captured through the estimation of arc travel time distributions.

In most real transportation networks, the congestion states among the arcs are highly correlated. As a result, independent simulation of each arc's congestion states leads to uncorrelated arc states and might cause incorrect travel time distributions. To avoid such problems, we simulated the network with historical data one day at a time. Specifically, we routed the vehicle from origin site to the destination site; at each decision epoch (e.g. node), the historic arc speed data was used to identify the congestion state and determine which arc to traverse next. We ran the simulations for 66 weekdays of May, June, and July 2009 and obtained 66 samples for all pairs of sites at different departure times. Although the number of runs was small, we believe it captured the dependency of arc congestion states better and accurately predicts the routing scenario's outcome. In addition, due to weather patterns/seasonality, traffic dynamics do change over extended periods. Hence, it is generally inappropriate to use data from extended periods (e.g., a year) to establish the tours and the dynamic routing policies. For these reasons, it might be best to re-optimize the tour and the dynamic routing policies at regular intervals (e.g., monthly or quarterly).

4.3. Building STD-TSP Tours

In this section, we construct the robust STD-TSP tours using the effective travel time distribution resulting from dynamic routing between every pair of sites (as explained in section 4.2). To quantify the benefits of using a dynamic routing policy, we also identify and select the robust STD-TSP tours with a static routing policy between each pair of sites.

In milk-run tours, the number of tour stops in urban areas is generally equal or greater than 5 stops per tour: approximately 5.6 in Denver [68], 6 in Calgary [69], and 6.2 in Amsterdam [70]. Our case study application also conforms to these estimates as there are 5 stops (i.e., one DC and four supplier sites). Although there are hundreds of suppliers replenishing the same DC, we only consider the subset of suppliers that were part of the same TSP tour. The determination of such supplier clusters is beyond the scope of this study and is assumed to be performed a priori based various factors (e.g., geographical supplier locations, nature of cargo) as in CFR. There were no pre-established requirements on the sequence of site visits and the truck had enough capacity to visit all sites in a single tour. As in most JIT environments, the time windows in this case study were set by trucking and OEM's logistics division and accepted by the suppliers as part of the sourcing contract. Therefore, we herein consider the case without time windows and then set the time windows for on-time performance in Section 4.4.

In the STD-TSP of the case study application, we have node 80 as the DC (origin site) and nodes 61, 103, 51, and 132 as the supplier sites (Fig. 1). Accordingly, there are (5-1)!=24 possible dominated and non-dominated tours. To capture the effect of traffic congestion, we consider 48 trip start times evenly spaced every half an hour and determine tours for each of them separately (Fig. 4). We assume all the sites' service times are 15 minutes. Since there are 4 sites other than the DC, the total service time is 60 minutes for each trip. To compare the results we define STD-TSP tours with following two site-to-site routing policies:

1. *STD-TSP tour with static routing policies (**Static policy**):* In practice, almost all commercial logistics software aims to identify TSP tours based on a static path between a pair of sites. First, for a given site pair and departure time, all paths are identified and then their expected path travel times are calculated according to the travel time distributions of paths' arcs. Next, the path with the least expected cost is selected as the static path to be used in the TSP tour. Then, for every trip start time, we select a robust TSP tour by solving STD-TSP using travel time distributions between pairs of sites estimated through the static paths.
2. *STD-TSP tour with dynamic routing policies (**Dynamic policy**):* In this policy, the paths between pairs of customers are dynamic routing policies (DRP). Based on the arc travel time distributions, congestion states and transition probabilities, we first generate DRPs between every pair of sites as described in Section 3.1. Then, these DRPs are simulated to find the site-to-site travel time distributions as described in Section 4.2. Finally, for every trip starting time, the robust TSP tour is selected using the DP algorithm for STD-TSP based on the simulated travel time distributions between pair of sites.

In identifying and selecting the robust tour, we set standard deviation coefficient in the cost function $b = 1.65$ such that the robust tour's trip duration is less than the mean-variance objective 97.5% of the time. We calculated the mean and standard deviations of trip times for all static and dynamic policy tours for evenly spaced 48 trip starting times beginning at 00:00am. The results revealed that 4 out of the 24 possible tours dominate the other tours for all 48 trip starting times for both static and dynamic policies. These *dominant* tours are: *tour 1*: 80→132→103→51→61→80; *tour 2*: 80→132→51→103→61→80; *tour 3*: 80→61→103→51→ 132→80; and *tour 4*: 80→61→51→103→132→80. Among these four tours, *tour 1* is the most selected tour by both static (40 times out of 48) and dynamic (41 times out of 48) policies. We report *tour 1* mean travel time and standard deviations in Fig. 4 for every starting time during the day. Note that these results are obtained by simulating the *tour 1* using the historic data (66 weekdays of May, June, and July 2009).

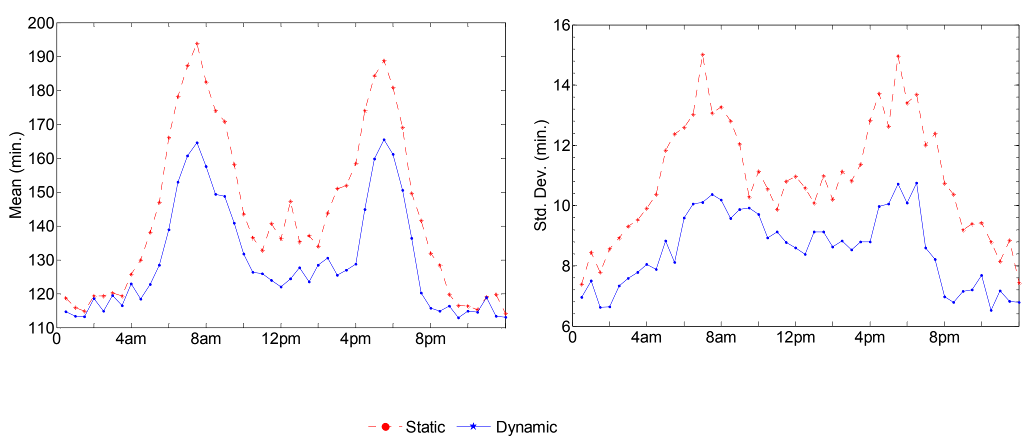

Fig. 4 The tour 1's (a) mean tour travel time (trip time - service times), (b) standard deviation for 48 starting times during the day for static and dynamic policies.

As expected, the savings are higher and rather significant during peak traffic times (e.g., around 8:00 and 17:00) and insignificant during uncongested periods. These results clearly illustrate the importance of using dynamic routing between pairs of sites. To further illustrate the savings, we present the selected robust tours and their mean and standard deviation of travel times identified by the two policies for two particular departure times in Table 1.

Table 1: Tours, tours mean travel times and standard deviations at two departure times for the static and dynamic policies.

Policy	Robust Tour	Departure Time	Mean Trip Time (min.)	Mean Tour Travel Time (min.)	Std. Dev. of Tour Travel Time (min.)
Static	*tour 1*	7:00	253.8	193.8	13.08
Dynamic	*tour 1*	7:00	224.5	164.5	10.37
Static	*tour 1*	7:30	242.4	182.4	13.27
Dynamic	*tour 2*	7:30	216.1	156.1	10.19

4.4. Evaluation of STD-TSP Tours with Time Windows

In the previous section, we selected the robust tours associated with static and dynamic routing policies across 48 starting times. We originally assumed no time windows. In this case study application, the determination of the TSP tour and the setting of time windows are sequential tasks. Specifically, the carrier first determines the tours for transportation efficiency and then the carrier and OEM's logistics division jointly set the spacing of time windows so as to maximize the on-time delivery performance. Next, we present and compare the trip duration results of using static and dynamic routing policies in a scenario where there are 4 DC replenishment shifts in each day and the shift starting times (ST) are $\mathrm{ST} = \{0:00;\ 6:00;\ 12:00;\ 18:00\}$ We then present the results after setting time windows.

According to the results in the preceding section, *tour 1* is the most selected tour by both static and dynamic policies across different trip start times. The other robust tours identified are *tours 2, 3,* and *4* in decreasing order of selection frequency. In Table 2 and Table 3, we provide the mean and standard deviation of *trip times* (tour travel time + service times) of these four dominant tours and their associated standard deviations at shift starting times when following *static* and *dynamic* policies between pair of sites, respectively. These results are obtained by simulating the corresponding tours using the historic data (66 weekdays of May, June, and July 2009).

Table 2**:** Mean of tour trip times at the beginning of shifts based on *static* and *dynamic* policies (without time windows).

Mean Tour Trip Times													
ST		**0:00**			**6:00**			**12:00**			**18:00**		
Policy		Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.
Tour	**1**	178.7	174.5	2.4%	238.2	212.9	10.6%	207.2	184.4	11.0%	229.1	210.5	8.1%
	2	177.2	174.0	1.8%	241.6	219.3	9.2%	207.8	185.7	10.6%	233.5	207.8	11.0%
	3	181.2	179.0	1.2%	236.4	220.0	6.9%	209.2	189.6	9.4%	237.9	220.1	7.5%
	4	183.6	181.1	1.4%	248.3	224.9	9.4%	205.1	193.5	5.7%	242.6	222.5	8.3%

Table 3 **:** Standard deviations of tour trip times at the beginning of shifts based on *static* and *dynamic* policies (without time windows).

Standard Deviation of Tour Trip Times													
ST		**0:00**			**6:00**			**12:00**			**18:00**		
Policy		Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.
Tour	**1**	7.8	7.0	10.3%	13.0	10.1	22.8%	10.9	8.4	23.4%	14.1	9.7	31.1%
	2	8.3	7.5	9.5%	13.6	11.2	17.8%	12.4	9.8	20.6%	14.3	10.8	25.0%
	3	7.8	7.7	1.0%	14.5	11.6	20.3%	11.9	10.5	11.9%	14.3	11.1	22.4%
	4	9.8	8.6	12.0%	15.2	12.2	20.3%	12.4	9.6	23.0%	14.8	13.0	12.6%

Table 2 results indicate that the mean tour trip time savings associated with dynamic routing are most in the two congested start times, namely 6:00 and 18:00, which are close to the urban area peak traffic times. We further note that the savings with start time at 12:00 is also as high as the congested periods (i.e., 6:00 and 18:00). The results in Table 3 for the standard deviation of tour trip times demonstrate the savings in variability similar to those in mean trip times.

The robust tour for each starting time is selected according to the mean-variance objective using the results in Table 2 and Table 3. These mean-variance objectives for the four dominant tours are presented in Table 4 along with that of the selected robust tour in the last row. The selected robust tours corresponding to static and dynamic policies are highlighted in bold for each start time. The dynamic policy's robust tour achieves the most savings over that of the static policy for trips starting at 12:00 and the mean-variance objective savings range from 2.6% to 12.0% with an average of 9.2%. The mean tour trip time savings based on the robust tours range from 1.5% to 11.0% with an average of 8.1% as can be calculated from Table 2. These tour trip duration savings correspond to the improvement in transportation efficiency. Similarly, the savings in the standard deviation of tour trip times based on the robust tours range from 16.5% to 23.7% with an average of 21.6% as can be calculated from Table 3. These savings correspond to the improvement in tour trip time reliability affecting the on-time delivery performance.

Table 4 results indicate that *tours 1* and *2* are dominant tours for the four start times. In the remainder of section, we assume that *tour 1* is selected for both static and dynamic policies. In fact, *tour 1* is indeed the selected robust tour for start times 6:00 and 12:00 and its performance difference from the selected robust tour is small for starting times of 0:00 and 18:00.

Table 4 **:** Mean-variance objectives of tour trip times at the beginning of shifts based on *static* and *dynamic* policies (without time windows).

		Mean-Variance Tour Trip Time Objectives											
ST		**0:00**			**6:00**			**12:00**			**18:00**		
Policy		Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.	Sta.	Dyn.	Improv.
Tour	**1**	191.5	**186.0**	2.9%	**259.7**	**229.5**	11.6%	**225.3**	**198.2**	12.0%	**252.3**	226.5	10.2%
	2	**190.9**	186.4	2.4%	264.1	237.8	10.0%	228.2	201.9	11.5%	257.2	**225.5**	12.3%
	3	194.1	191.8	1.2%	260.3	239.1	8.2%	228.8	206.9	9.6%	261.5	238.4	8.8%
	4	199.7	195.3	2.2%	273.4	244.9	10.4%	225.6	209.3	7.2%	267.1	243.9	8.7%
Robust Tour		190.9	186.0	2.6%	259.7	229.5	11.6%	225.3	198.2	12.0%	252.3	225.5	10.6%

Next, we set the time windows according to the procedure described in Section 3.2.1.1. Here, we assume the width of the time windows is 30 minutes for all supplier sites. Further, we allow unrestricted waiting for early arrivals at all sites. We illustrate the time windows through their centers (mean site arrival times) and deviations around centers (standard deviation of site arrival times) in Table 5 for the selected robust *tour 1*.

Table 5 **:** Simulated mean arrival times (with time windows) to the sites in the sequence of *tour 1* based on *static* and *dynamic* policies.

		Mean Site Arrival Times								Std. Dev. of Site Arrival Times							
ST		**0:00**		**6:00**		**12:00**		**18:00**		**0:00**		**6:00**		**12:00**		**18:00**	
Policy		Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.
Site	**132**	18.7	18.5	26.0	20.3	21.6	19.9	23.8	23.6	1.2	1.0	1.9	1.6	1.7	1.7	1.8	1.8
	103	67.3	66.6	87.9	80.7	79.2	74.1	102.7	93.5	3.3	2.8	5.2	4.4	4.7	3.9	6.2	4.8
	51	98.7	97.9	131.6	113.9	116.8	108.8	137.9	128.6	4.6	3.8	7.3	6.0	6.4	5.3	9.1	6.3
	61	147.0	143.9	197.2	172.5	169.8	154.3	192.2	180.2	6.3	5.5	10.3	8.3	8.7	7.0	11.8	8.0
	80	179.2	175.1	240.1	214.2	208.4	185.6	231.8	212.7	7.8	7.0	13.1	10.1	11.0	8.4	14.2	9.8

The mean and standard deviation of return times to DC (node #80) corresponds to the mean and standard deviation of the *tour 1* trip times. Note that the means and standard deviations of DC return times in Table 5 are different than those of tour trip times without time windows reported in Table 2. These differences are due to the waiting at the sites upon early arrival. The waiting due to early arrival increases (decreases) the mean (standard deviation) of the tour trip time. Table 6 presents the service level performance (on-time deliver) of static and dynamic policies for *tour 1* at different start times. These results are based on simulating *tour 1* using dynamic and static policies between sites subject to the time windows set for each policy in Table 5. Table 6 results show that as congestion increases, the dynamic policy taking real-time traffic information into account becomes increasingly superior to the static policy planning methods. The on-time delivery performance can be increased up to 8% for a site and up to 4% for a tour (starting at 18:00). We conclude that the dynamic policy not only decreases transportation cost (measured by trip time), but also increases the delivery service level performance (measured by on-time delivery).

Table 6 : On-time delivery performances (in percentages) of the policies with time windows.

		On-time delivery performances (in percentages)							
ST		**0:00**		**6:00**		**12:00**		**18:00**	
Policy		Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.	Sta.	Dyn.
Site	**132**	100	100	100	100	100	100	100	100
	103	100	100	100	100	100	100	100	100
	51	100	100	96	100	98	100	95	100
	61	98	100	91	98	96	98	91	97
	80	97	98	88	94	92	96	86	94

The results in Table 6 are obtained with the assumption that there is unrestricted waiting for early arrivals at all sites. Further, the time windows are centered on the mean site arrival times depending on whether static or dynamic routing policy is used between pairs of sites. As explained in Section 3.2.1.1, one could shift the time windows to the right of the center (expected site arrival time) to reduce the late arrival occurrences. However, the effectiveness of this modification relies on the maximum allowable waiting time imposed for early arrivals. To understand the effect of shifting time windows, we adapted time windows of the static policy as the time windows of the dynamic policy. This allows us to retain the assumption of unrestricted waiting for early arrivals and compare the on-time delivery results of dynamic policy with those in Table 6. The results of on-time delivery with dynamic policy using the time windows of the static policy are presented in Table 7. With this setting, the on-time delivery performance of the truck following the dynamic policy is 100 percent for all starting times and for all sites based on historic data (66 weekdays of May, June, and July 2009). Clearly, this improvement in on-time performance is attained with increased waiting at sites. Table 7 also presents the mean waiting times at sites.

Table 7 : On-time delivery performances (in percentages) and average waiting times (in minutes) for dynamic policy when setting time windows of dynamic policy as the time windows of static policy in Table 6.

		On-time delivery performances (in percentages)				**Waiting times (in minutes)**			
ST		0:00	6:00	12:00	18:00	0:00	6:00	12:00	18:00
Site	**132**	100	100	100	100	0	0	0	0
	103	100	100	100	100	0	0.06	0	0
	51	100	100	100	100	0	3.49	0.01	0
	61	100	100	100	100	0	10.12	2.21	0
	80	100	100	100	100	0	11.91	9.55	0

5. Conclusions

In this work, we studied the STD-TSP with dynamic routing problem. It is an extension of stochastic TSP and aims to find a robust milk-run tour of a given set of sites (i.e., DC and suppliers) while dynamically routing on a stochastic time-dependent road network between sites' visits to meet the time windows requirements. The solution is comprised of static TSP tour of sites that remains fixed for extended periods (e.g., months) and a dynamic routing policy between pairs of sites. The static tour is motivated by the fact that tours cannot be changed on a regular basis (e.g., daily) for milk-run pickup and delivery in routine JIT production. The objective trades off the expected duration of the tour with its variability, capturing the tradeoff between transportation efficiency and on-time delivery service level.

We proposed a sequential solution approach. We first determined the travel time distributions between each pair of sites by formulating and solving a stochastic dynamic programming formulation for the dynamic routing problem on a stochastic time-dependent road network. The dynamic routing model exploits the real-time traffic information available from ITS. We proposed effective data driven methods for accurate modeling and estimation of recurrent congestion states and their state transitions. Whereas we assumed arcs are independent in generating dynamic routing policies, we simulated dynamic routing policies using historic data to capture the arc dependencies in all our experiments. Using simulation results, we estimated the site-to-site travel time distributions. Once the travel time distributions were estimated for every pair of sites at different departure times, we employed a stochastic time-dependent dynamic programming (STD-DP) to solve the problem and select the robust tour minimizing the mean-variance objective of the trip time. We also provided a time window setting procedure to increase on-time delivery performance and support workload leveling.

We tested the proposed methodology on a real case study application using the road network from Southeast Michigan. This study corresponded to an automotive JIT production system where an OEM's DC is replenished by milk-run pickup and deliveries from multiple suppliers. The study road network covered major freeways and highways in and around the Detroit metropolitan area. To quantify the benefits of using dynamic policy, we compared the selected robust STD-TSP tours with those of the static routing policy between pair of sites. We first experimented without time windows for both static and dynamic policies. The results showed that the dynamic policy saves 8.1% in trip duration on the average and reduces standard deviation of trip duration by 21.6% on the average. After setting the time windows according to the expected site arrival times, we showed that the on-time delivery performance can be increased up to 8% for a site and up to 4% for a tour by using dynamic routing policy. Lastly, we showed that it is possible to further increase the on-time performance by setting the time windows of dynamic routing policy according to those of the static policy. We concluded that the dynamic policy not only decreases transportation cost (measured by trip time), but also increases the delivery service level performance (measured by on-time delivery).

There are several promising extensions of this research. The dynamic routing policies are generated by assuming arc independence. While we have partly compensated for this by simulating the policies using actual historical data from the

ITS network, the policies themselves are not guaranteed to be optimal if there are significant arc interactions. Hence, a future study is to account for the link interactions in modeling congestion and generating dynamic routing policies. Another future study is to integrate the proposed approach within the more general problem of VRP, where the supplier-route assignment decisions are made in addition to the routing of individual vehicles.

CHAPTER IV: DYNAMIC ROUTING IN STOCHASTIC TIME-DEPENDENT NETWORKS UNDER ARC INTERACTIONS[*]

Abstract – Just-in-time (JIT) production, an increasingly popular option in automotive and other industries, requires frequent and on-time pickups and deliveries. However, growing travel time delays and variability, attributable to increasing congestion in transportation networks, are greatly impacting reliability of transportation operations. In this study, our objective is to minimize the expected travel time from an origin to a destination in a stochastic time-dependent network with interactions between arcs (in terms of traffic flow/congestion conditions) to improve delivery efficiency.

We model the evolution of arc congestion "states" using Markov Chains, where state transitions for individual arcs are allowed to be dependent on the traffic states of successor arcs. We model the problem as a Markov decision process (MDP) and propose a stochastic dynamic programming formulation to solve the problem. MDP states are defined based on time of day, the physical state (decision point-vehicle location), and the information state (traffic network congestion states within the vicinity of the vehicle). The solution is a dynamic routing policy consisting

---

[*] This chapter resulted in the following publications:
- A.R. Guner, R.B. Chinnam, A. Murat: Dynamic Routing in Stochastic Time-Dependent Networks under Arc Interactions, (To be submitted to IEEE Transactions on ITS)

of decisions (which arc to take next) for every state. The dynamic nature of our routing policy exploits the real-time information available from various ITS (Intelligent Transportation Systems) sources. Results are very promising when the algorithms are tested in a simulated network of Los Angeles, CA freeways using historical data from the Caltrans PEMS.

Keywords— JIT logistics; dynamic routing; intelligent transportation systems; arc interactions; congestion

1. Introduction

Growing travel time delays and variability, attributable to increasing congestion in transportation networks, are greatly impacting reliability of transportation operations. 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. For example, many automotive final assembly plants in Southeast Michigan of U.S. 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, the congestion on road networks increases the travel time variability, thus, rendering it difficult to make delivery and pickups on-time. Golob and Regan [54] report that up to 85% of trucking companies miss their delivery time-window schedules due to road network congestion in California. In addition, 78% of the managers surveyed stated that the time-window schedules for pickup and deliveries force their drivers to operate under congested road network conditions. Congestion is forcing logistics solution providers to add significant travel time buffers to improve on-time delivery performance, causing idling of vehicles due to overly early arrivals. Given the levels of congestion, these travel time buffers can be significant. For example, in the Detroit metro area of Southeast Michigan, the buffers required for a typical OD pair can exceed 65% of free-flow travel time during peak congestion periods of the day to achieve 95% on-time delivery performance[7]. Given that automotive plants are heavily relying on JIT deliveries throughout a day, 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. However, these coping strategies (extra buffer of time or inventory) increase the costs and inefficiency.

In this study, our objective is to minimize expected travel time for a vehicle from an origin to a destination in a stochastic time-dependent network with correlated arc costs (resulting from interaction/dependence of arc traffic conditions of downstream arcs) to improve delivery efficiency. The reason for only modeling interaction from downstream arcs to upstream arcs is that congestion typically propagates backwards. For example, traffic buildup from incidents such as traffic accidents on freeways may cause queue spillbacks of several miles on its upstream and may affect vicinity streets; an inclement snowstorm may simultaneously affect a wide region of a road network, etc. Although these arc traffic interactions are an important characteristic of real world road networks, it is not considered in most routing studies.

We assume that an arc's state transition probability is dependent to the states of its adjacent successor arcs and have the single-stage Markovian property. To estimate the state transitions for each arc, we model the arc speeds for every pair of consecutive periods conditioned on the states of successor arcs. We employ clustering methods to identify the number of states and their transition probabilities. The changes in arc congestion states represent the traffic dynamics and are modeled as Markov processes. Efficient stochastic dynamic programming algorithms are proposed for developing optimal dynamic routing policies that consist of decisions (which arc to take next) for every state. The dynamic nature of our routing policy exploits the real-time information available from various ITS (Intelligent Transportation Systems) sources. The algorithms are tested in a simulated network of Los Angeles, CA freeways using historical data from the Caltrans PEMS2.

The contribution of this article is three-fold: First, we propose methods for accurate and efficient representation of recurrent congestion, in particular, identification of multiple congestion states and their transition patterns under arc interactions. Second, we propose a dynamic routing algorithm for stochastic time-dependent networks with arc interactions. Third, using data from a real transportation network, we test the proposed approach and demonstrate its value in improving travel times when compared to two methods from the literature.

The rest of the paper is organized as follows. Related literature is discussed in Section 2. Section 3 discusses the modeling of recurrent congestion under arc interactions in stochastic time-dependent networks. Section 4 presents the experimental results from a case study application. Section 5 concludes the study and proposes directions for future research.

2. Literature Survey

A variety of stochastic and time-dependent routing problems have been studied in the literature, of which many assume that arc (road segment) travel times are independent from each other. In contrary, an arc traffic state usually dependent to its (at least) nearby arcs. For instance, an incident on an arc or a bottleneck due to peak demand may cause congestion to spill on many upstream arcs. Because of the stochastic and time dependent features of the problem the optimal solution has to be a dynamic routing policy (DRP) rather than a single path [13]. In a DRP, the node to visit next depends on both the node and the time of arrival at that node.

Polychronopoulos and Tsitsiklis [23] study the shortest path problem with "recourse" (The case which a vehicle that starts moving towards the destination along a priori path has also recourse options of choosing a new path whenever new information is obtained.) They assume travel time on an arc becomes known and fixed upon the arrival of its starting node and they treated correlation of arc travel times by a discrete joint distribution. Waller and Ziliaskopoulos [18] studied a simplified recourse problem known as recourse with reset [71] problem with limited forms of spatial and temporal arc cost dependencies. They assume, given the cost of predecessor arcs, no further information is obtained through spatial dependence and the travel time distribution of an arc is conditional on the state of the preceding link. They define temporal dependency as learning the cost of an arc once the origin node of that arc is reached and this cost may change at different time visits. They proposed polynomial recourse algorithms for acyclic problems and developed complexity bounds for cyclic problems. In this paper, however, temporal dependency is defined in the sense of Hall [13], i.e., the traversal time distribution is conditional on time. Fan et al. [72] address the dynamic routing problem in static and stochastic networks with a limited correlation structure which is similar to [18]. They restricted arc states to be either congested or uncongested. They assume travel time distribution of a downstream arc is conditional on its upstream arcs states and these distributions are available. They also show that the label-correcting algorithm in [18] can also be derived from the dynamic programming point of view. Boyles [73] studied a similar problem to that in [18] in which conditional probabilities of adjacent link travel costs are utilized and travelers are assumed to remember only the travel time on the last link they traverse. Gao and Chabini [19] studied dynamic routing problem on stochastic and time dependent networks. They made different assumptions on travel time (e.g. no, partial, and full) information access and studied the effects of them on routing. They assume there is complete dependency where all travel times on all links at all time periods are correlated. They employed a joint distribution of travel time random variables to model this dependency. However, they didn't give any insight on how to get this joint matrix. In Nie and Wu [74], travel time correlations are restricted only to successor arcs states. They consider finding a priori paths which maximize arrival time reliability.

All of the above studies used dynamic programming (DP) or algorithms that can be derived from DP. However, Sivakumar and Batta [75] and Sen et. al. [76] used nonlinear and integer programming formulations as a solution method since they approached modeling of correlation with using covariance matrices. And DP is shown generally inapplicable to this kind of models.

In this research, travel time correlations are restricted only to adjacent arcs, similar to the work of Waller and Ziliaskopoulos [18], Nie and Wu [74], and Fan et al. [72], arc travel time distributions are allowed to vary over time, along the line of Hall [13], Fu and Rilett [15], Fu [17] and Miller-Hooks and Mahmassani [77] and dynamic programming is employed to find optimal policies.

3. Modeling Recurrent Congestion with Correlated Arcs

Let be a directed graph in which is the set of nodes and is the set of directed arcs. The (decision) node represents an intersection where the driver can decide which arc to select next. A directed arc is represented by an ordered pair of nodes in which is called the origin and is called the destination of the arc. Given an origin, and destination, (OD) node pair, the dynamic routing problem is to decide which arc to choose at each decision node such that the expected total OD 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. We first describe the modeling of recurrent congestion and then present the stochastic dynamic programming formulation and our solution approach.

3.1. Congestion Modeling with Arc Interactions

A directed arc $(n,n') \in A$ is labeled as observed if its real-time traffic data (e.g., velocity) is available through the ITS. An observed arc can be in $r+1 \in Z^{+}$ different states that represent the arc's traffic congestion level at a given time. Let

$s_a(t)$ be the congestion state of arc a at time period t, i.e. $s_a(t) = \{\text{Congested at level } i\} = \{i\}$ for $i = 1,2,\ldots,r+1$ and be determined as follows:

$$s_a(t) = \left\{i\,,\text{if } c_a^{i-1}(t) \le v_a(t) < c_a^i(t)\right\} \tag{1}$$

where $c_a^i(t)$ denotes the cut-off velocity at level i on arc a at time period t and $v_a(t)$ denotes the velocity on arc a at time period t. For instance, if there are two congestion levels (e.g., $r+1=2$), then the states will be $s_a(t) = \{\text{Uncongested}\} = \{0\}$ and $s_a(t) = \{\text{Congested}\} = \{1\}$.

We assume that 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., $S(t) = \left\{s_1(t), s_2(t), \ldots, s_{|A|}(t)\right\}$ 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 $s(t)$. We assume that an arc state and its transition to other states are dependent to its immediate successor (downstream) arcs states and state transitions have the single-stage Markovian property. We denote the successor arc set of arc a with $ScAS(a)$ or $\boldsymbol{a'}$ for compact representation where $ScAS(a) \equiv \left\{a' : a = (n,n') \in A\,,\, a' \equiv (n',n'') \in A\right\}$, i.e., the set of outgoing arcs from the destination node (n') of arc a. To estimate the state transition probability for each arc, we jointly model the speeds of two consecutive periods and condition this joint model to the successor arcs states. Accordingly, the time-dependent single-period state transition probability from state $s_a(t) = i$ to state $s_a(t+1) = j$ given successor arc set $\boldsymbol{a'}$ in state $s_{\boldsymbol{a'}}(t)$ is denoted by $P\{s_a(t+1) = j \mid s_a(t) = i, s_{\boldsymbol{a'}}(t)\} = \alpha_a^{ij,s_{\boldsymbol{a'}}}(t)$. We estimate the transition probability for arc *a*, $\alpha_a^{ij}(t)$ from the joint velocity distribution as follows:

$$\alpha_a^{ij,s_{\boldsymbol{a'}}}(t) = \frac{\left\| \left[V_a^i(t) \mid V_{\boldsymbol{a'}}^{s_{\boldsymbol{a'}}}(t) \right] \cap \left[V_a^j(t+1) \mid V_{\boldsymbol{a'}}^{s_{\boldsymbol{a'}}}(t) \right] \right\|}{\left| V_a^i(t) \mid V_{\boldsymbol{a'}}^{s_{\boldsymbol{a'}}}(t) \right|} \tag{2}$$

where the |*e*| operator corresponds to the frequency count of event *e*, $V_a^i(t)$ denotes the velocity vector on arc a at time t in state i for the sampled days, and $V_{\boldsymbol{a'}}^{s_{\boldsymbol{a'}}}$ denotes the velocity vector on arc set $\boldsymbol{a}$' at time t in state set $s_{\boldsymbol{a'}}$ for the sampled days. $V_a^i(t)$ is calculated as follows: $V_a^i(t) = c_a^{i-1}(t) \le V_a(t) < c_a^i(t)$ where $V_a(t)$ denotes the velocity vector on arc a at time t for the sampled days without conditioning on the state. Let $TP_a(t, t+1)$ denote the matrix of state transition probabilities from time t to time $t+1$, then, we have $TP_a(t, t+1) = \left[\alpha_a^{ij,s_{\boldsymbol{a'}}}(t) \right]_{ij}$. Note that the single-stage Markovian assumption is not restrictive in our approach as we could extend our methods to the multi-stage case by expanding the state space [48]. Let the network be in state $S(t)$ at time t, and we want to find the probability of the network state $S(t+\delta)$, where δ is a positive integer number. Given the independence assumption of the arcs' congestion states, this can be formulated as follows:

$$P\left(S(t+\delta) \mid S(t)\right) = \prod_{a=1}^{|A|} P\left(s_a(t+\delta) \mid s_a(t), s_{\boldsymbol{a'}}(t)\right) \tag{3}$$

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

$$TP_a\left(t,t+\delta\right)=\left[\alpha_a^{ij,s_{a'}}\left(t\right)\right]_{ij}\times\left[\alpha_a^{ij,s_{a'}}\left(t+1\right)\right]_{ij}\times\ldots\times\left[\alpha_a^{ij,s_{a'}}\left(t+\delta\right)\right]_{ij} \tag{4}$$

We assume that the distribution of an arc travel time is Gaussian. We further assume that the arc 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:

$$\delta\left(t,a,s_a\right)\sim N\left(\mu\left(t,a,s_a\right),\sigma^2\left(t,a,s_a\right)\right) \tag{5}$$

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

3.2. DP Formulation for Dynamic Routing

The objective of the dynamic routing algorithm is to minimize the expected travel time based on real-time information such as the path originates at node n_0 and ends at node n_d. Let us assume that there is a feasible path between $\left(n_0,n_d\right)$ where a path $p=\left(n_0,..,n_k,..,n_{K-1}\right)$ is defined as the sequence of (decision) nodes such that $a_k\equiv(n_k,n_{k+1})\in A$, $k=0,..,K-1$ 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 arcs set of node n_k, 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^{th} decision stage, t_k is the time at k^{th} decision stage where $t_k \in \{1,...,T\}$ $T > t_{K-1}$. T is an arbitrarily large number and is used to limit the planning horizon for modeling purposes. Note that we are discretizing the planning horizon.

While the optimal dynamic routing policy requires real-time consideration and projection of the traffic states of the complete network, this approach renders 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 from the long run average steady state probabilities of the arc congestion states. Hence, an efficient but practical approach would trade off the degree of look-ahead (e.g., the number of arcs to monitor) with the resulting projection accuracy and routing performance. This has been very well illustrated in [26]. Thus, we limit our look-ahead to a 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, the value of real-time information, and the congestion state transition characteristics of the arcs. 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 of the arcs in the network.

$ScAS(a_k)$, the successor arc set of arcs 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 nodes (n_{k+2}) of arcs a_{k+1}.

Since the total path 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 $\left(n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k}\right)$ of the system at the k^{th} decision stage is denoted by Ω_k. This state vector is composed of the state of the vehicle and network and thus is 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 Ω_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 based on the remaining expected travel time. The expected travel time at a given node with the selection of an outgoing arc is the summation of expected arc travel time on the arc chosen and the expected travel time of the next node. Let $\boldsymbol{\pi}_{n_0 n_d} = \{\pi_0, \pi_1, \ldots, \pi_{K-1}\}$ be the dynamic routing policy (DRP) of the trip that is composed of policies for each of the *K*-1 decision stages. For a given state $\Omega_k = \left(n_k, t_k, s_{a_{k+1} \cup a_{k+2}, k}\right)$, 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 π is as follows:

$$F^{\pi}(\Omega_0) = \underset{\delta_k}{E}\left\{\sum_{k=0}^{K-2} g(\Omega_k, \pi_k(\Omega_k), \delta_k) + \bar{g}(\Omega_{K-1})\right\} \tag{6}$$

where $\Omega_0 = (n_0, t_0, S_0)$ is the starting state of the system. δ_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 (ϕ) 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 Ω_{K-1}. Then, the minimum expected travel time can be found by minimizing $F(\Omega_0)$ over the policy vector π as follows:

$$F^*(\Omega_0) = \min_{\boldsymbol{\pi}_{n_0 n_d} = \{\pi_0, \pi_1, \ldots, \pi_{K-1}\}} F(\Omega_0) \tag{7}$$

The corresponding optimal policy is then:

$$\boldsymbol{\pi}^*_{n_0 n_d} = \underset{\boldsymbol{\pi}_{n_0 n_d} = \{\pi_0, \pi_1, \ldots, \pi_{K-1}\}}{\arg\min} F(\Omega_0) \tag{8}$$

Hence, the Bellman's cost-to-go equation for the dynamic programming model can be expressed as follows (Bertsekas, 2001):

$$F^*(\Omega_k) = \min_{\pi_k} \underset{\delta_k}{E}\left\{g(\Omega_k, \pi_k(\Omega_k), \delta_k) + F^*(\Omega_{k+1})\right\} \tag{9}$$

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:

$$F\left(\Omega_k \mid a_k\right)=\sum_{\delta_k} P\left(\delta_k \mid \Omega_k, a_k\right)\left[g\left(\Omega_k, a_k, \delta_k\right)+\sum_{s_{a_{k+1},k+1}} P\left(s_{a_{k+1},k+1}\left(t_{k+1}\right) \middle| s_{a_{k+1},k}\left(t_k\right), s_{a_{k+2},k}\left(t_k\right)\right)\right.$$
$$\left.\sum_{s_{a_{k+2},k+1}} P\left(s_{a_{k+2},k+1}\left(t_{k+1}\right) \middle| s_{a_{k+1},k}\left(t_k\right)\right) F\left(\Omega_{k+1}\right)\right] \tag{10}$$

where $P\left(\delta_k \mid \Omega_k, a_k\right)$ is the probability of travelling arc a_k in δ_k periods. $P\left(s_{a_{k+2},k+1}\left(t_{k+1}\right) \middle| s_{a_{k+2},k}\left(t_k\right)\right)$ is the transition probability of arc $a_{k+2} : a_{k+2} \in PScAS\left(a_k\right)$ to state $s_{a_{k+2},k+1}$ at stage $k+1$ given it was in state $s_{a_{k+1},k+1}$ at stage k. This probability can be calculated from the historical frequency of a state for a given arc and time.

We used the backward dynamic programming algorithm to solve $F^*\left(\Omega_k\right)$, $k = K-1, K-2, .., 0$. In the backward induction, we initialize the final decision epoch such that, $\Omega_{K-1} = \left(n_{K-1}, t_{K-1}, s_{K-1}\right)$, n_{K-1} is the destination node, and $F\left(\Omega_{K-1}\right) = 0$ if $t_{K-1} \le T$. Accordingly, a penalty cost is accrued whenever there is delivery tardiness, e.g., $t_{K-1} > T$. Note that $s_{K-1} = \varnothing$, since the destination node does not have any current and successor arc states, e.g. the travel terminates at the destination node.

4 Experimental Studies

In this section, we test the proposed methodology on a real case study application using the road network from Los Angeles, California (Fig. 1). Real-time traffic data for the observed arcs is collected and archived by Caltrans Performance Measurement System (PeMS) and available online at pems.dot.ca.gov. PeMS collects data from automatic sensors (mostly loop detectors) that are installed at thousands of California freeway locations, across all lanes, including over 3000 locations in Los Angeles Metropolitan area [78].We implemented all of our algorithms and methods in Matlab R2010a and executed them on a Pentium IV machine (with CPU 1.6 GHz and 1024 MB RAM) running Microsoft Windows XP operating system.

Our experimental study is outlined as follows: Section 4.1 introduces the road network used for demonstrating the performance of the proposed algorithms along with a description of its general traffic conditions. Section 4.2 describes the estimation and modeling process for recurrent congestion with correlated arcs and illustrates through some sample arcs of the network. Section 4.3 presents the results of the proposed approach and compare them with two other methods.

4.1 Case Study Network and Traffic Data

The case study road network covers some major freeways and highways of Los Angeles metropolitan area (Fig. 1a). The network has 32 nodes and a total of 78 arcs with 67 observed arcs and 21 unobserved arcs. In this application, we used data from 66 weekdays of June, July and, August 2010 for the full 24 hours of each day. The raw speed data was aggregated at a resolution of 5 minute intervals. For the experimentation, we increased the resolution of data to one data-point per minute through linear interpolation. Since the collected speed data is averaged across different vehicle classes (i.e., automobile, trucks), we assumed that the truck being routed could also cruise at the collected average speeds.

(a) (b)

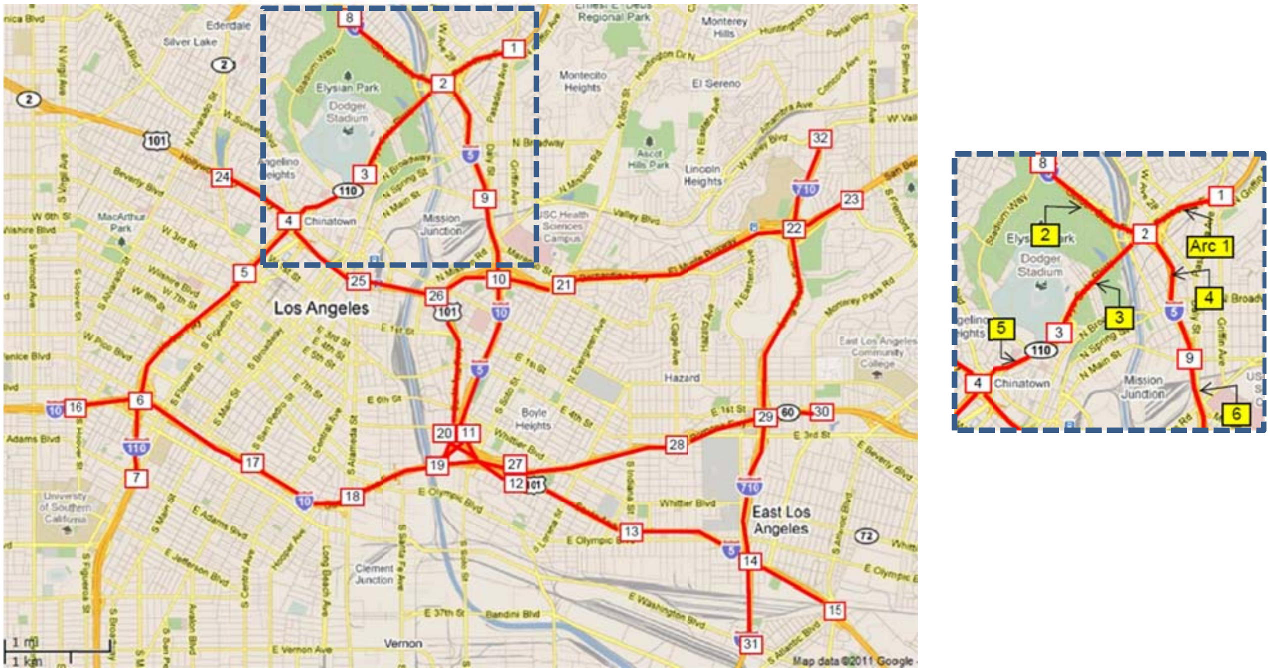

Fig. 1 Los Angeles, CA road network considered for experimental study and the sub-network used for illustration.

To better illustrate the need and modeling of congestion states, we present the data and congestion state identification and separation procedures in a sub-network of the Los Angeles, CA road network illustrated in Fig. 1b.

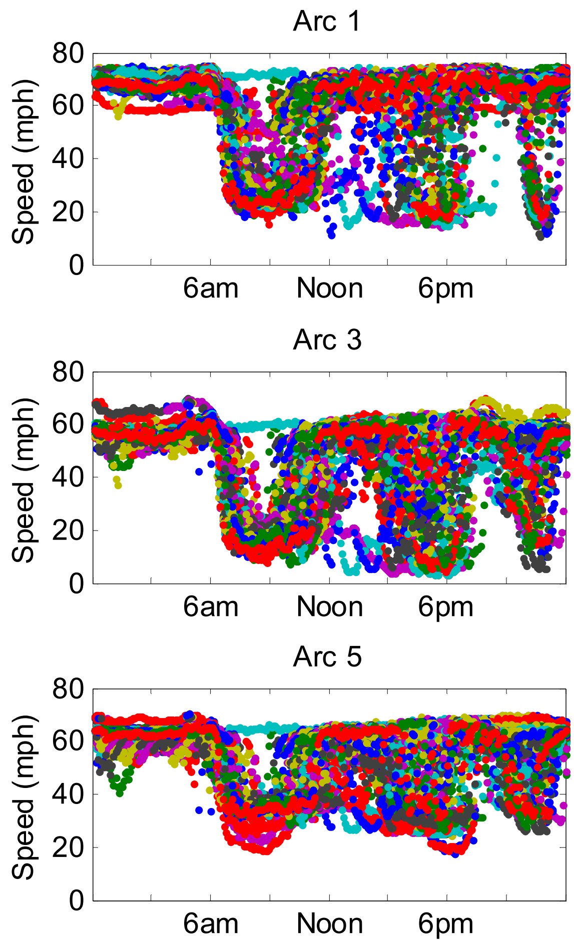

Fig. 2 Traffic speeds of sub-network arcs for June, July and August 2010 weekdays. Each color represents a day.

The speed data of arcs for the weekdays is illustrated in Fig. 2. The mean and standard deviations of speed are plotted in Fig. 3. From Fig. 2 and Fig. 3, it can be seen clearly that the traffic speeds follow a stochastic non-stationary distribution that vary with the time of the day.

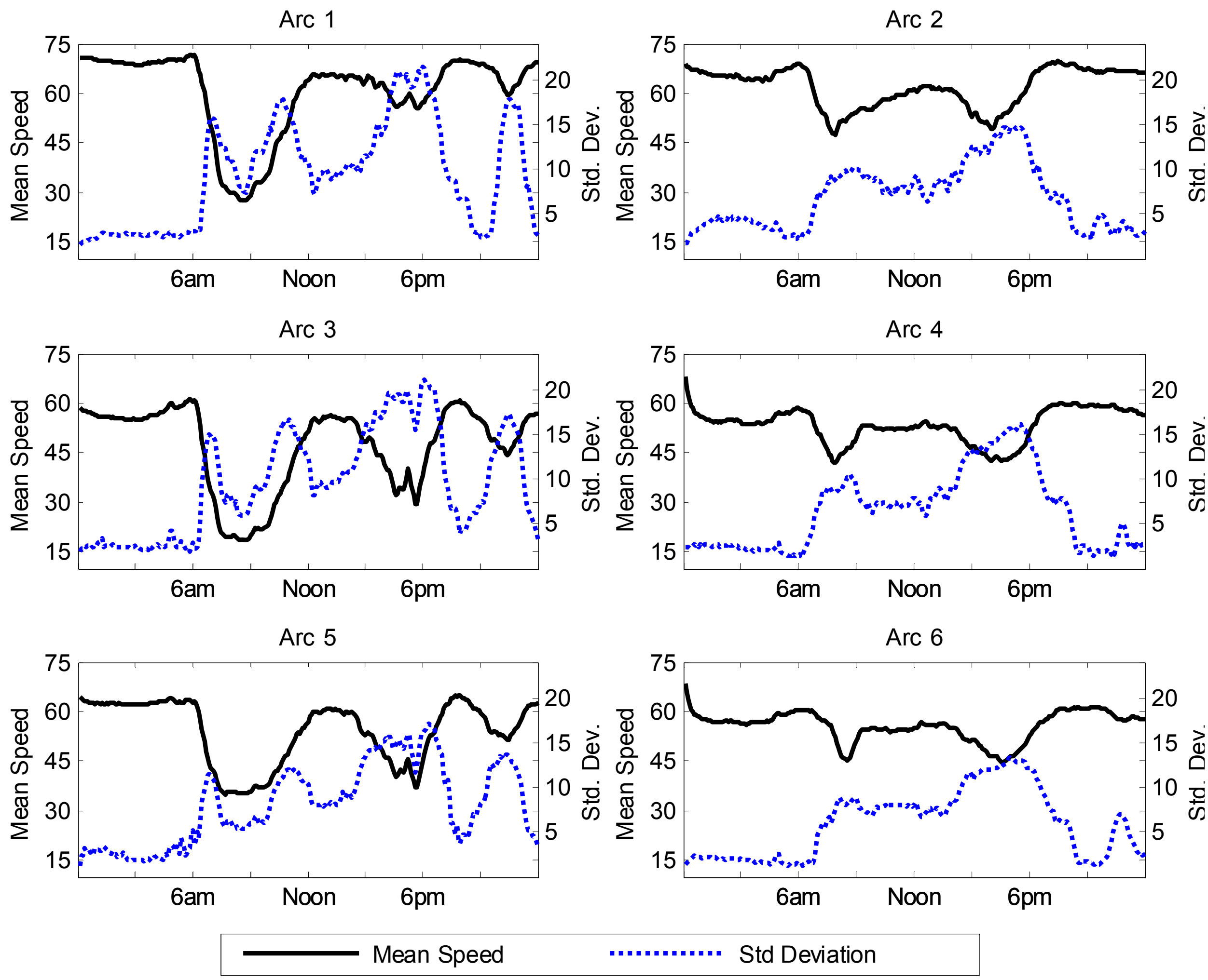

Fig. 3 Mean and standard deviations of sub-network arcs speeds (mph) by time of the day with time interval resolution of 5 minutes.

4.2 State Parameters Estimation Procedure

The proposed dynamic routing algorithm calls for identification of different congestion states, estimation of their state transition rates, and estimation of arc traverse times for congestion states by time of the day. Given the traffic speed data we employed a three-step procedure given below and summarized in Fig. 4 to calculate these parameters:

Step 1: GMM Clustering Find cut off speeds and number of clusters **Step 2:** Post-Processing Smooth number of clusters (Heuristic method) Recalculate cut off speeds with imposing 'new' number of clusters (GMM) Smooth cut off speeds (Window averaging) **Step 3:** Estimate State-Transition Probabilities Calculate state transition probabilities with 'updated' cut off speeds Smooth state transition probabilities (Window averaging)

Fig. 4 Summary of 'State Parameters Estimation Procedure'

Step 1: GMM Clustering

We estimate the number of congestion states and cut-off speeds for each arc by time of the day with the following procedure. We first cluster traffic speed data for every pair of two consecutive time periods. With assuming congestion clusters normally distributed, we partition this cluster using a Gaussian mixture model (GMM) where each partition models congestion states. In particular, we exploited the '*gmdistribution'* function built-in Statistics Toolbox V7.3 of Matlab R2010a [79] for partitioning. The product of GMM is bi-variate joint Gaussian distributions. We limited the maximum number of clusters to two for the experiments. Thus the cut off speed is the mean of the distribution in the case of a one partition and the intersection of two distributions probability density function in the case of a two partitions. We describe the procedure in detail below.

Let $v_a(t)$ is the sampled speed data of arc a at time t. We form a cluster of traffic speed data from every pair of two consecutive time periods such as t and t+1 and denote this matrix with $V_a(t,t+1)$. With employing *gmdistribution* function we decide an initial number of clusters (one or two) which minimize Akaike Information

Criterion (AIC). If the AIC value is minimized when there is one cluster we assign the number of clusters as ‘1’ and the cut-off speed as the distribution mean at this time point for this arc. If the AIC value is minimized when there is two clusters we first look at the Mahalanobis distance to understand if clusters are overlapping or not. If the Mahalanobis distance is less than a given value (e.g. 6) than we say the clusters are overlapping (e.g. bigger cluster surrounds most or all of the smaller cluster data points) and conclude that there is one cluster in fact, and we assign the number of clusters as ‘1’ and the cut-off speed as the whole distribution mean at this time point for this arc. If the Mahalanobis distance is greater than the given value we look at the size of each cluster. If the size of smaller cluster is less than a given portion (e.g. 10%) of the whole data points than we assume the smaller cluster as outliers and conclude that there is one cluster. We assign the number of clusters as ‘1’ and the cut-off speed as the whole distribution mean at this time point for this arc. If the size of smaller cluster is greater than the given value, then we assign the number of clusters as ‘2’. The cut-off speed between congested (left cluster) and uncongested states (right cluster) for arc a at time t is denoted by $c_a(t)$ and is calculated as follows: $c_a(t) = x, x : f_t^1(x) = f_t^2(x)$ where $f(\cdot)$ is the projected probability density function for state $i = \{1: congested, 2: uncongested\}$.

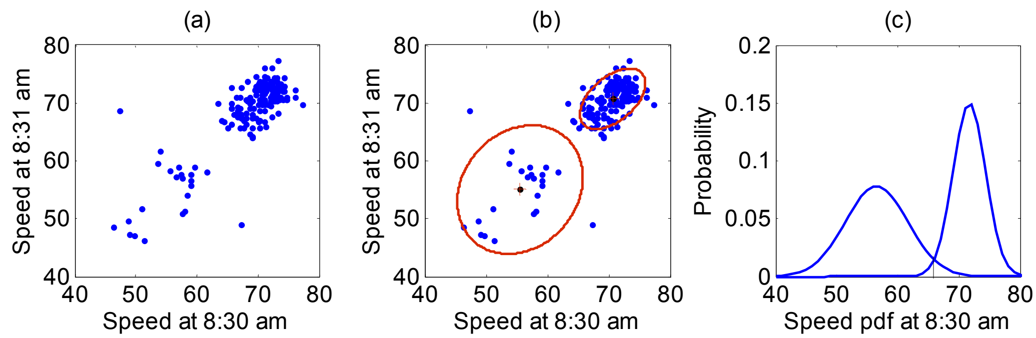

Fig. 5 (a) Example joint plots of traffic speeds in consecutive periods for modeling state at 8:30 am; (b) Cluster joint distributions of speed at 8:30am generated by GMM; (c) Partitioned traffic states based on projections.

As expected, the GMM procedure yielded two states generally, as in Fig. 5 (resulting in states denoted 'congested' and 'uncongested' states with c_1 (8:30)= 64.9 mph), and rarely a single state during periods of low traffic (as in Fig. 6). The number of clusters determined by the GMM procedure during the day for a sample arc is given in Fig. 8a.

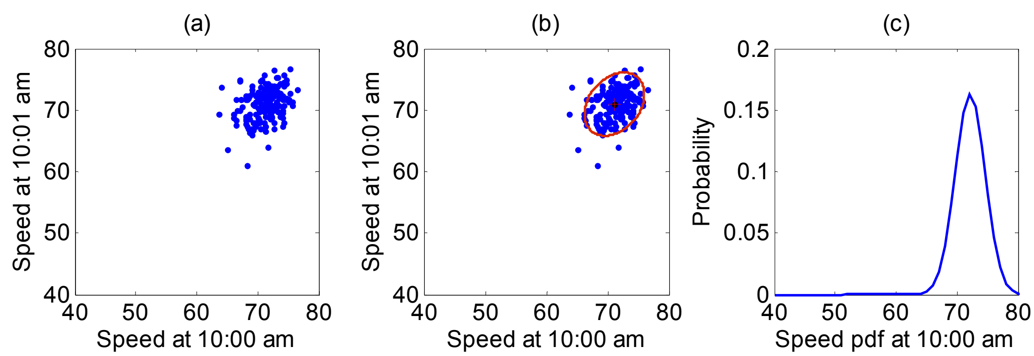

Fig. 6 (a) Example joint plots of traffic speeds in consecutive periods for modeling state-transitions at 10:00 am; (b) Single cluster joint distribution of speed at 10:00am generated by GMM; (c) Partitioned traffic states based on projections.

Step 2: Post-Processing

The second step is the post-processing of the number of clusters and the cut-off speed to get smoother transition of cut-off speeds during the day.

The post-processing of the number of clusters (NC) is a heuristic method in which we flip some of the number of clusters from 2 to 1 or vice versa so that the number of clusters is smoother in time series during the day. We assume the number of clusters must be same at least in 3 consecutive time periods. The method first flags the number of clusters that doesn't hold this assumption (i.e. two consecutive one clusters in the middle of a series of two clusters). The next step is deciding what should be the number of clusters for flagged ones. Regardless of the size of a flagged series we flip all flagged ones two its neighbor number of clusters if both (before and after the series) neighbors have the same number of clusters. If neighbors are not same then we look at the original values of flagged ones and select the value that is more than the other value. In a tie we randomly select the final value. Fig. 7 illustrates the procedure for two different cases. Fig. 8 gives the number of clusters before and after the procedure for arc 1 of the example network.

Time	…	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	…
Initial NC	…	1	1	1	2	2	1	1	1	…
Flag					f	f				
Case	Both neighbors are series of 1 convert flagged series to 1									
Final NC	…	1	1	1	1	1	1	1	1	…

Time	…	t	t+1	t+2	t+3	t+4	t+5	t+6	t+7	t+8	t+9	t+10	…
Initial NC	…	1	1	1	2	2	1	2	1	2	2	2	…
Flag					f	f	f	f	f				
Case	Neighbors are different; convert flagged series to 2 since it is more.												
Final NC	…	1	1	1	2	2	2	2	2	2	2	2	…

Fig. 7 Illustrative examples for number of cluster (NC) smoothing.

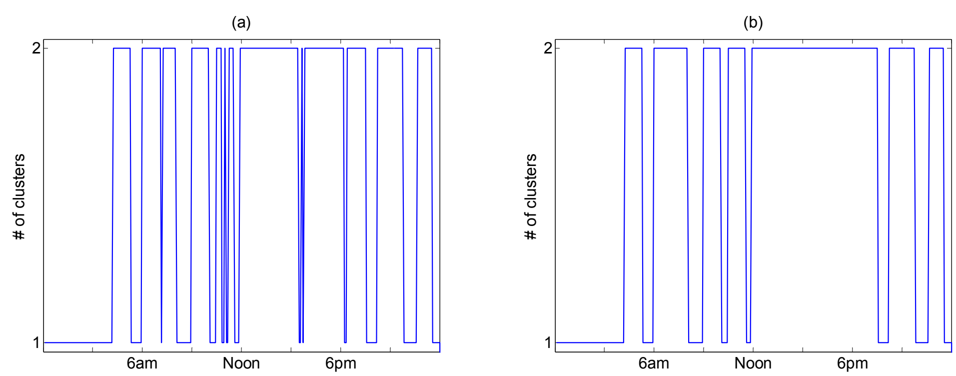

Fig. 8 Number of clusters of arc 1 before (a) and after (b) post-processing during the day.

After getting the processed number of clusters we use these to get smoother cut-off speeds. To do this we first impose GMM distribution function the "new number of clusters". After getting the cut-off speeds we smooth them through window averaging as given in equation (11) where $\theta_a(t)$ is the parameter (e.g $c_a(t)$) at time t

for arc *a*. In Fig. 9 we give the number of clusters before and after the window averaging.

$$\theta_a(t) = \sum_{\mathbf{w}=-w}^{w} \theta_a(t+\mathbf{w}) \Big/ 2w+1 \tag{11}$$

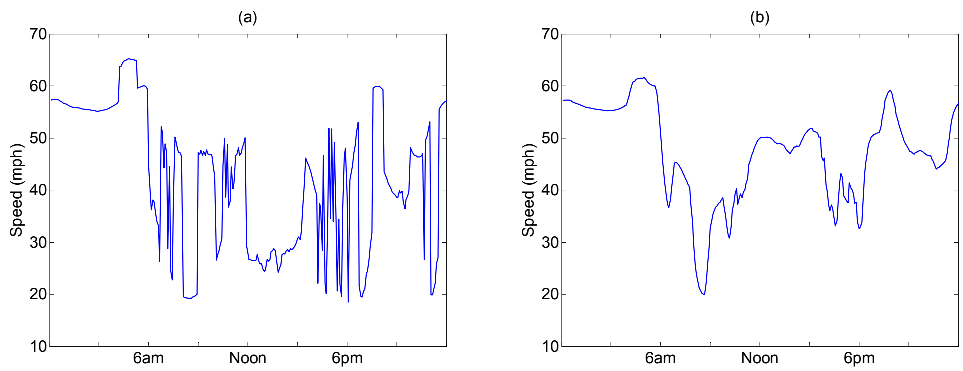

Fig. 9 Cutoff speed of arc 1 before (a) and after (b) post-processing during the day.

Step 3: Estimate State-Transition Probabilities

In this step we first estimate the transition probabilities for an arc as described in section 3 from the joint velocity distribution. Then we do a window averaging as given in equation (11) to get more smooth transition probabilities. Smoothed transition probabilities for arc 1 is given in Fig. 10, Fig. 11, and Fig. 12 for different correlation scenarios. In figures, the α denotes the probability of state transition from congested state to congested state and β denotes the probability of state transition from uncongested state to uncongested state. 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.

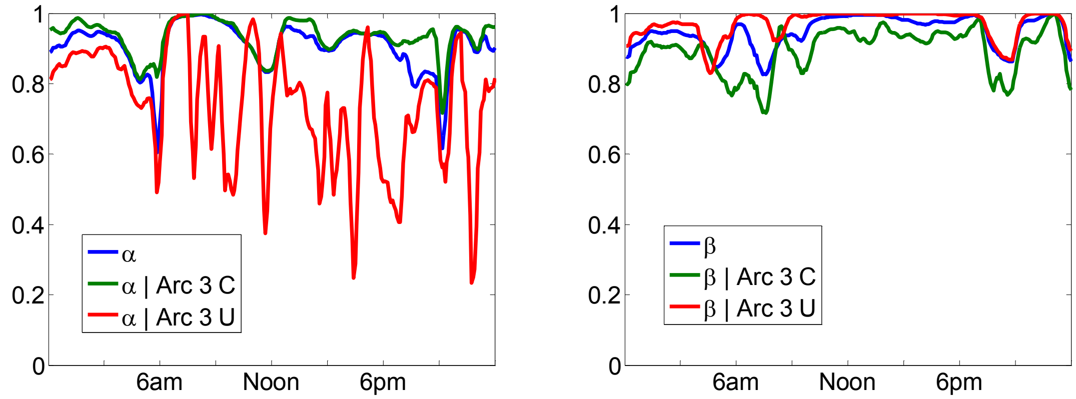

Fig. 10 Unconditional (α, β) and conditional (on arc 3 congestion states where C: congested and U: uncongested state) congestion state probabilities of arc 1 during the day.

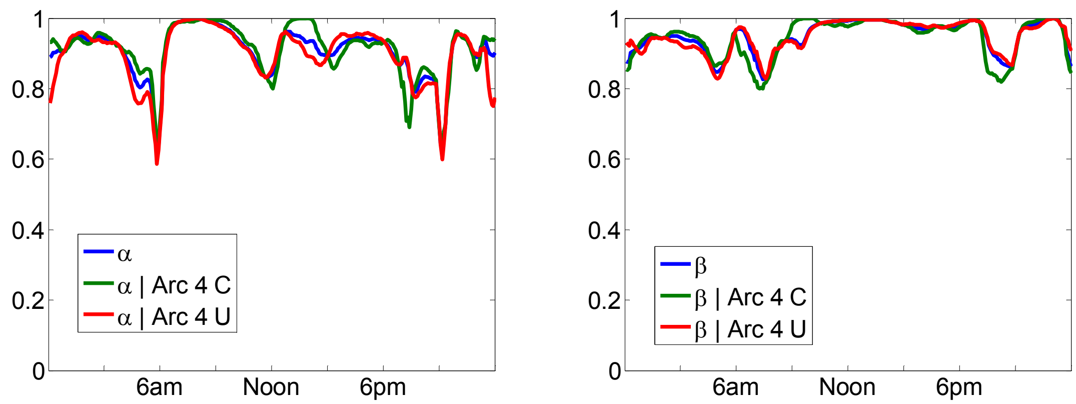

Fig. 11 Unconditional (α, β) and conditional (on arc 4 congestion states where C: congested and U: uncongested state) congestion state probabilities of arc 1 during the day.

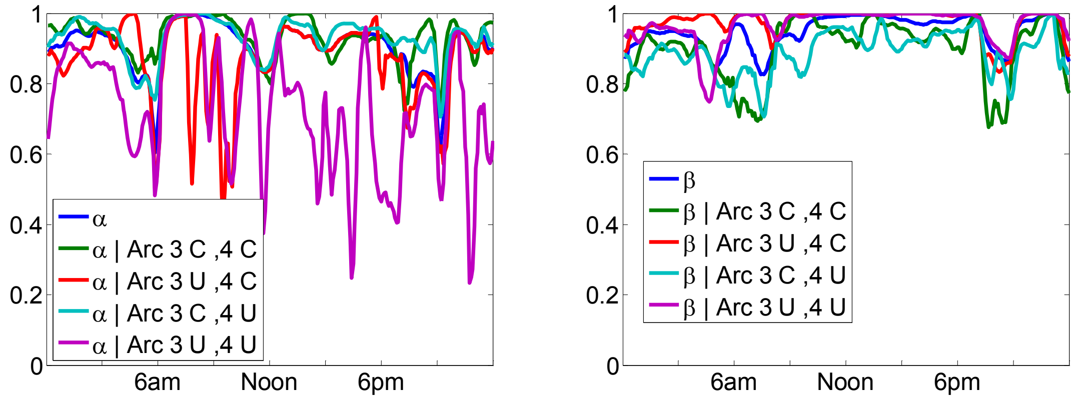

Fig. 12 Unconditional (α, β) and conditional (on arc 3 and 4 congestion states at the same time where C: congested and U: uncongested state) congestion state probabilities of arc 1 during the day.

4.3 Results

This section highlights the potential savings from explicit modeling of recurrent congestion with correlated arcs for dynamic routing policy. We compare the results of following three routing policies:

3. *Static routing policy* (**SRP**): In practice, almost all commercial logistics software aims to identify a static path from an origin to a destination. We select the path with the least expected cost (for most of the time throughout day) as the static path for each OD pair.
4. *Dynamic routing policy* (**DRP**)**:** In this policy, we route the vehicle based on dynamic routing policy without modeling arc correlations.
5. *Dynamic routing policy with arc corelation* (**DRP-A**)**:** In this policy, we route the vehicle based on the policy defined in this paper.

Note that both dynamic routing policies does not identify an optimal path, rather, identifies the best policy based on the time of the day, location of the vehicle, and the traffic state of the network .

Simulation of the network dynamics can be done in two different ways: simulation of arcs using estimated arc parameters (from historical data) independently or using the available historical data each day at a simulation run. In the former option independent simulation of each arc's congestion states leads to uncorrelated arc states and might cause incorrect simulation of travel times. Also, in most real transportation networks, the congestion states among the arcs are highly correlated. To avoid such problems, we chose the latter: We ran the simulations for 66 weekdays of June, July, and August 2010 and obtained travel times from arc speeds (with assuming the speed does not change when traversing the arc and equals to arrival speed to the arc) at different departure times. Although the number of runs was small, we believe it captured the dependency of arc congestion states better and accurately predicts the routing scenario's outcome.

To compare the results of different policies given above we identified 6 OD pairs as follows: OD1: 32 to 7; OD2: 7 to 32; OD3: 8 to 15; OD4: 15 to 8; OD5: 24 to 30; OD6: 30 to 24. Fig. 13 plots the median travel times for every half hour between 6am and 9pm for all policies. Fig. 14 plots the corresponding median percentage savings of employing DRP-A over SRP and DRP. It is clear that savings are higher (if there is) and rather significant during peak traffic times and lower when there is not much congestion, as can be expected. These results clearly illustrate the importance of using dynamic routing policy with arc correlation.

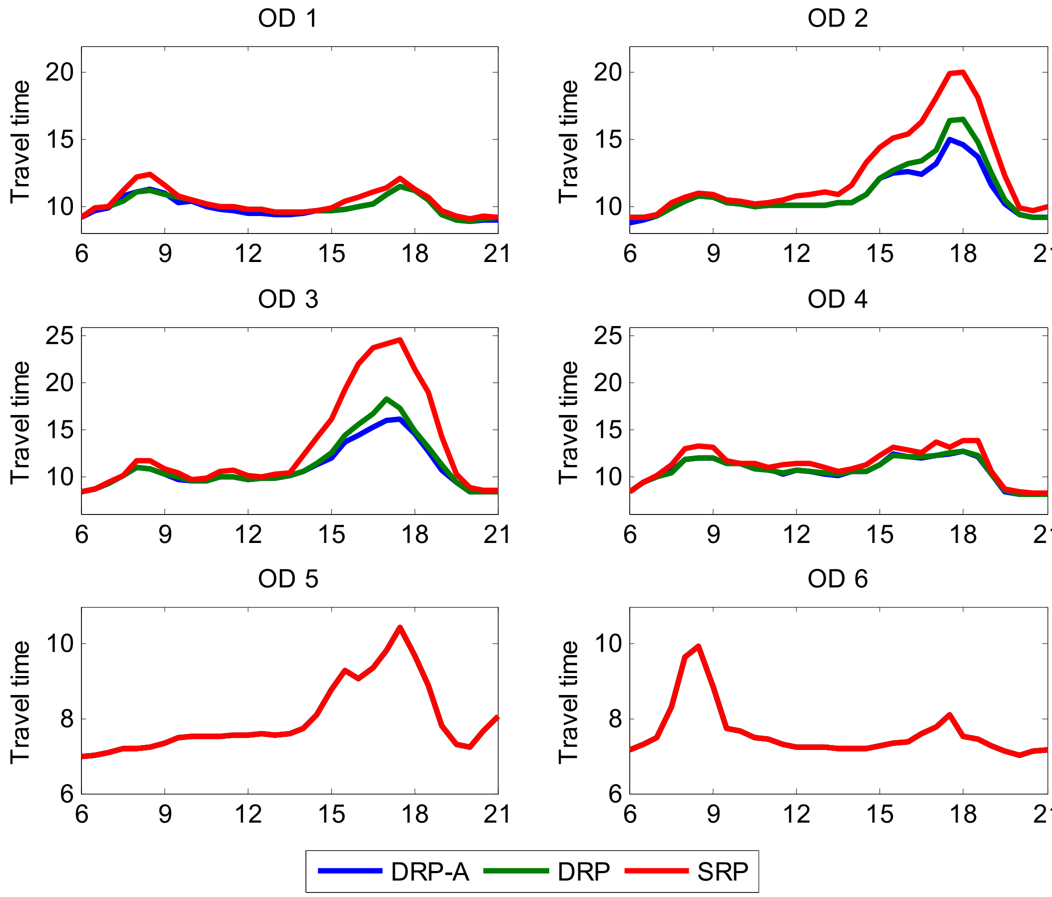

Fig. 13 Median travel times of 66 experiment days for every half hour between 6am and 9pm. DRP-A: Dynamic routing policy with arc correlations. DRP: Dynamic routing policy. SRP: Static routing policy.

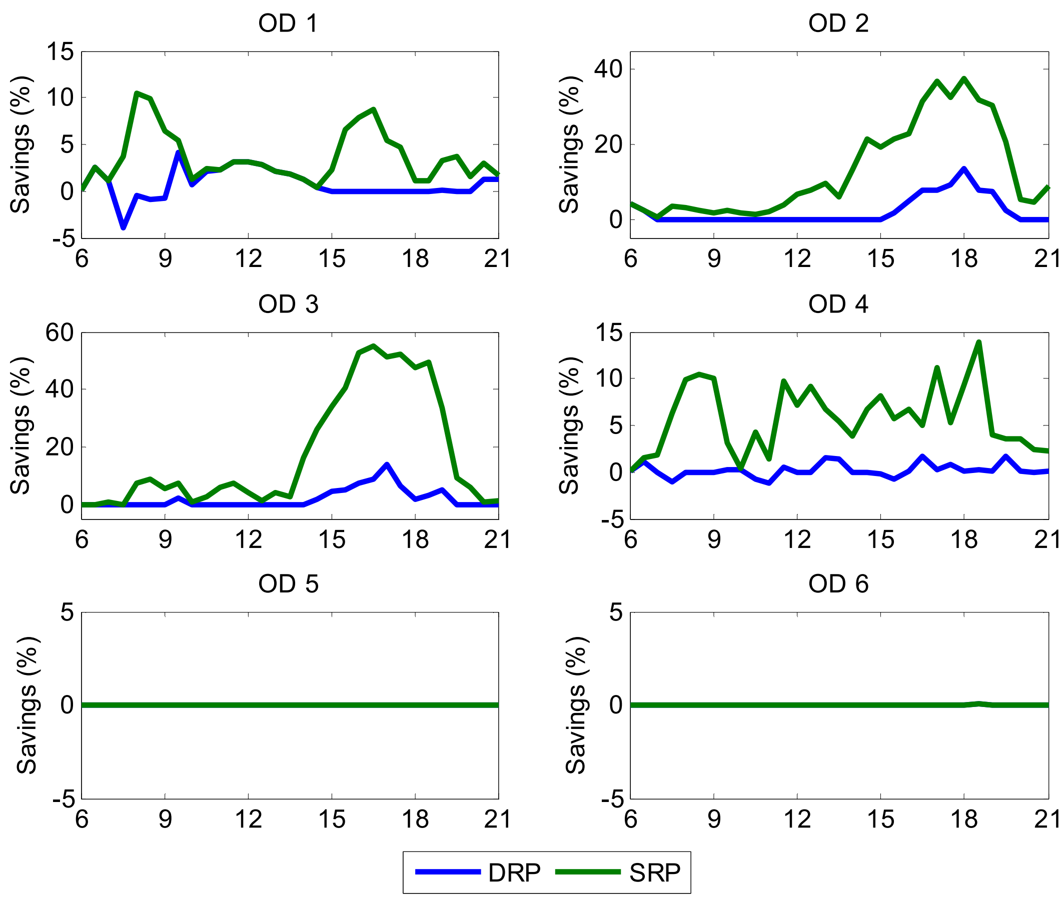

Fig. 14 Savings from DRP-A over DRP and SRP in travel times for 66 experiment days for every half hour between 6am and 9pm. DRP-A: Dynamic routing policy with arc correlations. DRP: Dynamic routing policy. SRP: Static routing (baseline path) policy.

5 Conclusions

Routing in transportation networks that involve arc interactions have not been well studied in the literature. The case of dynamic routing in a stochastic time-dependent network with correlated arc travel times is formulated. The proposed model effectively exploits real-time traffic information from Intelligent Transportation Systems (ITS). 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. 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 visit 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 Los Angeles road network, collected by Caltrans PEMS, is used to illustrate the performance of the proposed models. Our experiments clearly illustrate the superior performance of the stochastic dynamic programming derived dynamic routing policies when they accurately account for recurrent congestion (i.e., they differentiate between congested and uncongested traffic states). 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. As expected, savings are higher during peak travel times and lower when traffic tends to be free-flow.

With the growing availability of reliable traffic information to drivers, more drivers in the network will be adjusting their paths based on newly acquired information. Anticipating and responding to the behavior of the rest of the traffic will become critical for effective dynamic routing and is recommended for further research.

CHAPTER V: CONCLUSIONS & FUTURE RESEARCH

This dissertation proposes practical dynamic routing models that can effectively exploit real-time traffic information from Intelligent Transportation Systems (ITS) regarding recurrent congestion and non-recurrent congestion stemming from incidents (e.g., accidents) in transportation network. 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 first model the problem as a non-stationary stochastic shortest path problem with 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.

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 integrate non-recurrent congestion to our initial model. The proposed model 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).

We also proposed an extension of stochastic TSP and aims to find a robust milk-run tour of a given set of sites (i.e., DC and suppliers) while dynamically routing on a stochastic time-dependent road network between sites' visits to meet the time windows requirements. The solution is comprised of static TSP tour of sites that remains fixed for extended periods (e.g., months) and a dynamic routing policy between pairs of sites. The static tour is motivated by the fact that tours cannot be changed on a regular basis (e.g., daily) for milk-run pickup and delivery in routine JIT production. The objective trades off the expected duration of the tour with its variability, capturing the tradeoff between transportation efficiency and on-time delivery service level.

We proposed a sequential solution approach to TSP with dynamic routing problem. We first determined the travel time distributions between each pair of sites by formulating and solving a stochastic dynamic programming formulation for the dynamic routing problem on a stochastic time-dependent road network. The dynamic routing model exploits the real-time traffic information available from ITS. We proposed effective data driven methods for accurate modeling and estimation of recurrent congestion states and their state transitions. Whereas we assumed arcs are independent in generating dynamic routing policies, we simulated dynamic routing policies using historic data to capture the arc dependencies in all our experiments. Using simulation results, we estimated the site-to-site travel time distributions. Once the travel time distributions were estimated for every pair of sites at different departure times, we employed a stochastic time-dependent dynamic programming (STD-DP) to solve the problem and select the robust tour minimizing the mean-variance objective of the trip time. We provided a time window setting procedure to increase on-time delivery performance and support workload leveling.

Network problems that include arc interactions have not been well studied in the literature. We improved our Markov decision process (MDP) formulation so that arc interactions are also captured. We proposed dynamic routing policy to select the best node to go next based on “state” (vehicle location, time of day, and network congestion state-)and its transition where arc transition probabilities constructed in conjunction with its correlated (e.g. downstream) arcs.

All our methods are tested with real network ITS data either from South-East Michigan road network, collected in collaboration with Michigan Intelligent Transportation System Center & Traffic.com or from Los Angeles road network, collected by Caltrans PEMS 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), non-recurrent congestion attributed to incidents, and arc interactions. 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 arc interactions and non-recurrent congestion stemming from incidents can yield significant savings.

We also tested our proposed TSP with dynamic routing methodology on a real case study application using the road network from Southeast Michigan. This study corresponded to an automotive JIT production system where an OEM's DC is replenished by milk-run pickup and deliveries from multiple suppliers. The study road network covered major freeways and highways in and around the Detroit metropolitan area. To quantify the benefits of using dynamic policy, we compared the selected robust STD-TSP tours with those of the static routing policy between pair of sites. We first experimented without time windows for both static and dynamic policies. The results showed that the dynamic policy saves 8.1% in trip duration on the average and reduces standard deviation of trip duration by 21.6% on the average. After setting the time windows according to the expected site arrival times, we showed that the on-time delivery performance can be increased up to 8% for a site and up to 4% for a tour by using dynamic routing policy. Lastly, we showed that it is possible to further increase the on-time performance by setting the time windows of dynamic routing policy according to those of the static policy. We concluded that the dynamic policy not only decreases transportation cost (measured by trip time), but also increases the delivery service level performance (measured by on-time delivery).

There are several promising extensions of this research. With the availability of more mobile traffic information, more drivers may adjust their paths based on newly acquired information. Anticipating and responding to the behavior of the rest of the traffic incorporated in a dynamic routing framework is recommended for further research.

Another future study is to integrate the proposed approach within the more general problem of VRP, where the supplier-route assignment decisions are made in addition to the routing of individual vehicles.

Appendix

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

$$\mu_a\left(t_k+\Delta,s\right)-\mu_a\left(t_k,s\right)\geq-\frac{q}{c}\Delta$$

Proof. Let $a\in A$ denote the incident arc with origin and destination nodes $\left(n_k,n_{k+1}\right)$. Further, let $t_{k+1}=t_k+\delta_a\left(t_k,s,t_k-t_{inc}^0\right)$ 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_d) under an optimal policy is $E\left\{\delta_a\left(t_k,s,i=\left(t_k-t_{inc}^0\right)\right)+F^*\left(n_{k+1},t_k+\delta_a\left(t_k,s,t_k-t_{inc}^0\right),w\right)\right\}$, where the second term is the cost-to-go from node n_{k+1} at time t_{k+1} 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_d) at time t_k and $t_k+\Delta$ with $D\left(t_k\right)$ and $D\left(t_k+\Delta\right)$, respectively.

$$D\left(t_k\right)=\delta_a\left(t_k,s,t_k-t_{inc}^0\right)+F^*\left(n_{k+1},t_k+\delta_a\left(t_k,s,t_k-t_{inc}^0\right),w\right)$$

$$D\left(t_k+\Delta\right)=\delta_a\left(t_k+\Delta,s,t_k+\Delta-t_{inc}^0\right)+F^*\left(n_{k+1},t_k+\delta_a\left(t_k+\Delta,s,t_k+\Delta-t_{inc}^0\right),w\right)$$

Assumption 3 states that at any node arrival time $\left(t_k\right)$, 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 Δ unit time (smallest discrete time interval),

$$E\left\{D\left(t+\Delta\right)\right\}-E\left\{D\left(t\right)\right\}\geq-\Delta.$$

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 Δ waiting time this leads to the following relation for every t_k:

$$F^*\left(n_{k+1},t_k+\Delta+\delta_a\left(t_k+\Delta,s,t_k+\Delta-t_{inc}^0\right),w\right)-F^*\left(n_{k+1},t_k+\delta_a\left(t_k,s,t_k-t_{inc}^0\right),w\right)\geq-\Delta.$$

Hence, we have the following relation:

$$E\left\{\delta_a\left(t_k+\Delta,s,t_k+\Delta-t_{inc}^0\right)\right\}-E\left\{\delta_a\left(t_k,s,t_k-t_{inc}^0\right)\right\}\geq-\Delta,$$

where,

$$\begin{aligned}E\left\{\delta_a\left(t_k,s,t_k-t_{inc}^0\right)\right\}&=E\left\{\delta_a\left(t_k,s,i=0\right)+\Theta_a\left(\varphi,c,\rho,q,i=t_k-t_{inc}^0\right)\right\}\\&=\mu_a\left(t_k,s\right)+E\left\{\Theta_a\left(\varphi,c,\rho,q,i=t_k-t_{inc}^0\right)\right\},\end{aligned}$$

and, $\mu_a\left(t_k,s\right)$ is the mean travel time on arc *a* at time t_k with congestion state *s*. The expression $E\left\{\Theta_a\left(\varphi,c,\rho,q,i=t_k-t_{inc}^0\right)\right\}$ 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 [31]), e.g., $E\left\{\Theta_a\left(\varphi,c,\rho,q,i=t_k-t_{inc}^0\right)\right\}=\frac{q-\rho}{\rho}\left(t_k-t_{inc}^0\right)$.

The other alternative is the variable-delay regime in which the vehicle experiences a delay somewhere between the no-delay and the maximum delay [31].

$$E\left\{\Theta_a\left(\varphi,c,\rho,q,i=t_k-t_{inc}^0\right)\right\}=\frac{c-\rho}{c}\mu_{inc}-\frac{c-q}{c}\left(t_k-t_{inc}^0\right).$$

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:

$$E\left\{\delta_a\left(t_k+\Delta, s, t_k+\Delta-t_{inc}^0\right)\right\}-E\left\{\delta_a\left(t_k, s, t_k-t_{inc}^0\right)\right\} \geq -\Delta$$

$$\mu_a\left(t_k+\Delta, s\right)+E\left\{\Theta_a\left(\varphi, c, \rho, q, t_k+\Delta-t_{inc}^0\right)\right\}-\mu_a\left(t_k, s\right)-E\left\{\Theta_a\left(\varphi, c, \rho, q, t_k-t_{inc}^0\right)\right\} \geq -\Delta$$

$$\mu_a\left(t_k+\Delta, s\right)-\mu_a\left(t_k, s\right)-\frac{c-q}{c}\left(t_k+\Delta-t_{inc}^0\right)+\frac{c-q}{c}\left(t_k-t_{inc}^0\right) \geq -\Delta$$

$$\mu_a\left(t_k+\Delta, s\right)-\mu_a\left(t_k, s\right) \geq -\frac{q}{c}\Delta .$$

When we take the limit $\Delta \rightarrow 0$, we have, $\frac{d\mu_a(t,s)}{dt}\Big|_{t=t_k} \geq -\frac{q}{c}$.

In the maximum delay case, the no node waiting condition can be expressed as:

$$E\left\{\delta_a\left(t_k+\Delta, s, t_k+\Delta-t_{inc}^0\right)\right\}-E\left\{\delta_a\left(t_k, s, t_k-t_{inc}^0\right)\right\} \geq -\Delta$$

$$\mu_a\left(t_k+\Delta, s\right)+E\left\{\Theta_a\left(\varphi, c, \rho, q, t_k+\Delta-t_{inc}^0\right)\right\}-\mu_a\left(t_k, s\right)-E\left\{\Theta_a\left(\varphi, c, \rho, q, t_k-t_{inc}^0\right)\right\} \geq -\Delta$$

$$\mu_a\left(t_k+\Delta, s\right)-\mu_a\left(t_k, s\right)+\frac{q-\rho}{\rho}\left(t_k+\Delta-t_{inc}^0\right)-\frac{q-\rho}{\rho}\left(t_k-t_{inc}^0\right) \geq -\Delta$$

$$\mu_a\left(t_k+\Delta, s\right)-\mu_a\left(t_k, s\right) \geq -\frac{q}{\rho}\Delta .$$

When we take the limit $\Delta \rightarrow 0$, we have, $\frac{d\mu_a(t,s)}{dt}\Big|_{t=t_k} \geq -\frac{q}{\rho}$.

Note that since the capacity under incident is less than regular capacity, i.e. $c>\rho$, we have the condition for variable-delay regime more strict than the fixed-delay regime, i.e., $-q/c > q/\rho$. Hence, for arbitrary waiting time Δ, no node waiting condition under incident is:

$$\mu_a(t_k+\Delta,s)-\mu_a(t_k,s)\geq -\frac{q}{c}\Delta \ .$$

□

REFERENCES

1. Cambridge Systematics Inc. and Texas Transportation Institute, *Traffic congestion and reliability: Linking solutions to problems*, 2004, US FHWA Office of Operations: Washington, D.C.
2. Güner, A.R., A. Murat, and R.B. Chinnam, *Dynamic routing under recurrent and non-recurrent congestion using real-time its information.* Computers & Operations Research, 2012. **39**(2): p. 358–373.
3. Güner, A.R., A. Murat, and R.B. Chinnam. *Dynamic Routing Using Real-time ITS Information*. in *Third International Workshop on Intelligent Vehicle Controls & Intelligent Transportation Systems*. 2009. Milan, Italy.
4. Chang, T.-S., Y.-W. Wan, and W.T. OOI, *A stochastic dynamic traveling salesman problem with hard time windows.* European Journal of Operational Research, 2010. **198**(3): p. 748-759.
5. Taniguchi, E., et al., *City logistics: Network modelling and Intelligent Transportation Systems*2001, Pergamon: Elsevier Science.
6. Schrank, D. and T. Lomax, *The 2007 annual urban mobility report*, in *Urban Mobility Information*2007, Texas Transportation Institute.
7. Texas Transportation Institute and Cambridge Systematics Inc., *Monitoring Urban Freeways in 2003: Current conditions and trends from archived operations data*, 2004, US FHWA Office of Operations: Washington, DC.
8. Sigal, C.E., A.A.B. Pritsker, and J.J. Solberg, *The stochastic shortest route problem.* Operations Research, 1980. **28**: p. 1122-1129.

9. Frank, H., *Shortest paths in probabilistic graphs.* Journal of Operational Research, 1969. **17**: p. 583-599.

10. Loui, R.P., *Optimal paths in graphs with stochastic or multidimensional weights.* Communications of the ACM, 1983. **26**(9): p. 670-676.

11. Eiger, A.P., P. Mirchandani, and H. Soroush, *Path preferences and optimal paths in probabilistic networks.* Transportation Science 1985. **19**: p. 75–84.

12. Dijkstra, E., *A note on two problems in connection with graphs.* Numerical Mathematics, 1959. **1**: p. 269-271.

13. Hall, R., *The fastest path through a network with random time-dependent travel time.* Transportation Science, 1986. **20**(3): p. 182-188.

14. Bertsekas, D.P. and J. Tsitsiklis, *An analysis of stochastic shortest path problems.* Mathematics of Operations Research 1991. **16**: p. 580-595.

15. Fu, L. and L.R. Rilett, *Expected shortest paths in dynamic and stochastic traffic networks.* Transportation Research Part B, 1998. **32B**(7): p. 499-516.

16. Bander, J.L. and C.C. White III, *A heuristic search approach for a nonstationary shortest path problem with terminal costs.* Transportation Science, 2002. **36**: p. 218-230.

17. Fu, L., *An adaptive routing algorithm for in vehicle route guidance systems with real-time information.* Transportation Research Part B, 2001. **35B**(8): p. 749-765.

18. Waller, S.T. and A.K. Ziliaskopoulos, *On the online shortest path problem with limited arc cost dependencies.* Networks, 2002. **40** (4): p. 216-227.

19. Gao, S. and I. Chabini, *Optimal routing policy problems in stochastic time-dependent networks.* Transportation Research Part B, 2006. **40B**: p. 93-122.

20. Miller-Hooks, E.D. and H.S. Mahmassani, *Least possible time paths in stochastic, time-varying networks.* Computers & Operations Research, 1998. **12**: p. 1107-1125.

21. Miller-Hooks, E.D. and H.S. Mahmassani, *Least expected time paths in stochastic, time-varying transportation networks.* Transportation Science, 2000. **34** p. 198-215.

22. Psaraftis, H.N. and J.N. Tsitsiklis, *Dynamic shortest paths in acyclic networks with Markovian arc costs.* Operations Research, 1993. **41**: p. 91-101.

23. Polychronopoulos, G.H. and J.N. Tsitsiklis, *Stochastic shortest path problems with recourse.* Networks, 1996. **27**(2): p. 133-143.

24. Azaron, A. and F. Kianfar, *Dynamic shortest path in stochastic dynamic networks: Ship routing problem.* European Journal of Operational Research, 2003. **144**: p. 138-156.

25. Kim, S., M.E. Lewis, and C.C. White III, *Optimal vehicle routing with real-time traffic information.* IEEE Transactions on Intelligent Transportation Systems, 2005. **6**(2): p. 178-188.

26. Kim, S., M.E. Lewis, and C.C. White III, *State space reduction for non-stationary stochastic shortest path problems with real-time traffic congestion information.* IEEE Transactions on Intelligent Transportation Systems, 2005. **6**(3): p. 273-284.

27. Wirasinghe, S.C., *Determination of traffic delays from shock-wave analysis.* Transportation Research, 1978. **12**: p. 343-348.

28. Al-Deek, H., A. Garib, and A.E. Radwan, *New method for estimating freeway incident congestion.* Transportation Research Record 1995. **1494**: p. 30-39.

29. Mongeot, H. and J.B. Lesort, *Analytical expressions of incident-induced flow dynamics perturbations.* Transportation Research Record, 2000. **1710**: p. 58-68.

30. Morales, J., *Analytical procedures for estimating freeway traffic congestion.* Public Roads, 1986. **50**(2): p. 55-61.

31. Fu, L. and L.R. Rilett, *Real-time estimation of incident delay in dynamic and stochastic networks.* Transportation Research Record, 1997. **1603**: p. 99-105.

32. Cohen, H. and F. Southworth, *On the measurement and valuation of travel time variability due to incidents on freeways.* Journal of Transportation And Statistics, 1999. **2**(2): p. 123-131.

33. HCM, *Highway Capacity Manual*2000, Washington, D.C.: Transportation Research Board.

34. Li, J., C.-J. Lan, and X. Gu, *Estimation of incident delay and its uncertainty on freeway networks.* Transportation Research Record, 2006. **1959**: p. 37-45.

35. Baykal-Gürsoy, M., W. Xiao, and K. Ozbay, *Modeling traffic flow interrupted by incidents.* European Journal of Operational Research, 2008, In Press.

36. Lindley, J.A., *Urban freeway congestion: quantification of the problem and effectiveness of potential solutions.* Institute of Transportation Engineers Journal 1987. **57**(1): p. 27-32.

37. Garib, A., A.E. Radwan, and H. Al-Deek, *Estimating magnitude and duration of incident delays.* ASCE Journal of Transportation Engineering 1997. **123**(6): p. 459-466.

38. Sullivan, E.C., *New model for predicting incidents and incident delay.* ASCE Journal of Transportation Engineering, 1997. **123**(5): p. 267-275.

39. Gaver, D.P., *Highway delays resulting from flow stopping conditions.* Journal of Applied Probability, 1969. **6**: p. 137-153.

40. Golob, T.F., W.W. Recker, and J.D. Leonard, *An analysis of the severity and incident duration of truck-involved freeway accidents.* Accident Analysis & Prevention, 1987. **19**(4): p. 375-395.

41. Giuliano, G., *Incident characteristics, frequency, and duration on a high volume urban freeway.* Transportation Research Part A, 1989. **23A**(5): p. 387-396.

42. Khattak, A.J., J.L. Schofer, and M.-H. Wang, *A simple time sequential procedure for predicting freeway incident duration.* IVHS Journal, 1995. **2**(2): p. 113-138.

43. Noland, R.B. and J.W. Polak, *Travel time variability: A review of theoretical and empirical issues.* Transport Reviews, 2002. **22**(1): p. 39-54.

44. Hensher, D. and F. Mannering, *Hazard-based duration models and their application to transport analysis.* Transport Reviews, 1994. **14**: p. 63-82.

45. Nam, D. and F. Mannering, *An exploratory hazard-based analysis of highway incident duration.* Transportation Research Part A, 2000. **34A**(2): p. 85-102.

46. Ferris, M.C. and A. Ruszczynski, *Robust path choice in networks with failures.* Networks, 2000. **35**: p. 181-194.

47. Thomas, B.W. and C.C. White III, *The dynamic shortest path problem with anticipation.* European Journal of Operational Research, 2007. **176**: p. 836-854.

48. Bertsekas, D.P., *Dynamic programming and optimal control*. 2nd ed2001: Athena Scientific. 320.

49. Ross, S.M., *Introduction to Probability Models*. Ninth ed2006: Academic Press, Inc.

50. Verbeek, J.J., N. Vlassis, and B. Kröse, *Efficient greedy learning of Gaussian Mixture Models.* Neural Computation, 2003. **5**(2): p. 469-485.

51. Chen, C., A. Skabardonis, and P. Varaiya. *Travel time reliability as a measure of service*. in *82nd Annual Meeting Transportation Research Board*. 2003. Washington, D.C.

52. Groenevelt, H., *The just-in-time system*, in *Handbooks in Operations Research and Management Science*, S.C. Graves, A.H.G. Rinnooy Kan, and P.H. Zipkin, Editors. 1993, North-Holland: Amsterdam. p. 629-670.

53. Figliozzi, M.A., *The impacts of congestion on commercial vehicle tour characteristics and costs.* Transportation Research Part E, 2010. **46**: p. 496-506.

54. Golob, T.F. and A.C. Regan, *Traffic congestion and trucking managers' use of automated routing and scheduling.* Transportation Research Part E, 2003.

55. Cordeau, J.-F., et al., *The VRP with time windows*, in *The Vehicle Routing Problem*, P. Toth and D. Vigo, Editors. 2000, SIAM: Philadelpia,PA.

56. Chuah, K.H. and J.C. Yingling, *Routing for just-in-time supply pickup and delivery system.* Transportation Science, 2005. **39**(3): p. 328-339.

57. Ohlmann, J.W., M.J. Fry, and B.W. Thomas, *Route design for lean production systems.* Transportation Science, In Press.

58. Johnson, D.S. and L.A. McGeoch, *The traveling salesman problem: A case study in local optimization*, in *Local Search in Combinatorial Optimization*, E. Aarts and J.K. Lenstra, Editors. 1997, Wiley: London. p. 215-310.

59. Junger, M., G. Reinelt, and G. Rinaldi, *The traveling salesman problem*, in *Network Models*, M. Ball, et al., Editors. 1995, North Holland: Amsterdam. p. 225-330.

60. Laporte, G., *Fifty years of vehicle routing.* Transportation Science, 2009. **43**(4): p. 408-416.

61. Laporte, G., *A concise guide to the Traveling Salesman Problem.* Journal of the Operational Research Society, 2010. **61**(1): p. 35-40.

62. Malandraki, C. and R.B. Dial, *A restricted dynamic programming heuristic algorithm for the time dependent traveling salesman problem.* European Journal of Operational Research, 1996. **90**: p. 45-55.

63. Ichoua, S., M. Gendreau, and J.-Y. Potvin, *Vehicle dispatching with time-dependent travel times.* European Journal of Operational Research, 2003. **144**: p. 379-396.

64. Laporte, G., F. Louveaux, and H. Mercure, *The vehicle routing problem with stochastic travel times.* Transportation Science, 1992. **26**(3): p. 161-170.

65. Lambert, V., G. Laporte, and F. Louveaux, *Designing collection routes through bank branches.* Computers & Operations Research, 1993. **20**(7): p. 783-791.

66. Jula, H., M. Dessouky, and P.A. Ioannou, *Truck route planning in nonstationary stochastic networks with time windows at customer locations.* IEEE Transactions on Intelligent Transportation Systems, 2006. **7**(1).

67. Houck, D., et al., *The travelling salesman as a constrained shortest path problem: Theory and computational experiment.* Opsearch, 1980. **17**(2-3): p. 93-109.

68. Holguin-Veras, J. and G. Patil, *Observed trip chain behavior of commercial vehicles.* Transportation Research Record, 2005. **1906**.

69. Hunt, J. and K. Stefan, *Tour-based microsimulation of urban commercial movements*, in *16th International Symposium on Transportation and Traffic Theory (ISTTT16)*2005: Maryland.

70. Vleugel , J. and M. Janic, *Route choice and the impact of 'logistic routes'*, in *Logistics systems for sustainable cities*, E. Taniguchi and R. Thompson, Editors. 2004, Elsevier.

71. Provan, J.S., *A polynomial-time algorithm to find shortest paths with recourse.* Networks, 2003. **41**(2): p. 115-125.

72. Fan, Y., R. Kalaba, and J. Moore, *Shortest paths in stochastic networks with correlated link costs.* Computers and Mathematics with Aplications, 2005. **49**: p. 1549-1564.

73. Boyles, S. and S.T. Waller, *A stochastic delay prediction model for real-time incident management.* ITE Journal, 2007. **77**: p. 18-24.

74. Nie, Y.M. and X. Wu, *Reliable a Priori Shortest Path Problem with Limited Spatial and Temporal Dependencies*, in *Transportation and Traffic Theory*, W.H.K. Lam, S.C. Wong, and H.K. Lo, Editors. 2009, Springer Science. p. 169-195.

75. Sivakumar, R. and R. Batta, *The variance-constrained shortest path problem.* Transportation Science, 1994. **28**: p. 309-316.

76. Sen, S., et al., *A mean-variance model for route guidance in advanced traveler information systems.* Transportation Science, 2001. **35**: p. 37-49.

77. Miller-Hooks, E.D. and H.S. Mahmassani, *Path comparisons for a priori and time-adaptive decisions in stochastic, time-varying networks.* European Journal of Operational Research, 2003. **146**: p. 67-82.

78. Chen, C., *Freeway Performance Measurement System (PeMS)*, in *Electrical Engineering and Computer Sciences*2002, University of California, Berkeley.

79. The MathWorks Inc. *Statistics Toolbox User's Guide*. 2010; Available from: http://www.mathworks.com/access/helpdesk/help/pdf_doc/stats/stats.pdf.

ABSTRACT

DYNAMIC ROUTING ON STOCHASTIC TIME-DEPENDENT NETWORKS USING REAL-TIME INFORMATION

by

ALI RIZA GÜNER

August 2011

Advisor: Dr. Ratna Babu Chinnam

Major: Industrial Engineering

Degree: Doctor of Philosophy

In just-in-time (JIT) manufacturing environments, on-time delivery is one of the key performance measures for dispatching and routing of freight vehicles. Both the travel time delay and its variability impact the efficiency of JIT logistics operations, that are becoming more and more common in many industries, and in particular, the automotive industry. In this dissertation, we first propose a framework for dynamic routing of a single vehicle on a stochastic time dependent transportation network using real-time information from Intelligent Transportation Systems (ITS). Then, we consider milk-run deliveries with several pickup and delivery destinations subject to time windows under same network settings. Finally, we extend our dynamic routing models to account for arc traffic condition dependencies on the network.

Recurrent and non-recurrent congestion are the two primary reasons for travel time delay and variability, and their impact on urban transportation networks is growing in recent decades. Hence, our routing methods explicitly account for both recurrent and non-recurrent congestion in the network. In our modeling framework, we develop alternative delay models for both congestion types based on historical

data (e.g., velocity, volume, and parameters for incident events) and then integrate these models with the forward-looking routing models. The dynamic nature of our routing decisions exploits the real-time information available from various ITS sources, such as loop sensors.

The forward-looking traffic dynamic models for individual arcs are based on congestion states and state transitions driven by time-dependent Markov chains. We propose effective methods for estimation of the parameters of these Markov chains. Based on vehicle location, time of day, and current and projected network congestion states, we generate dynamic routing policies using stochastic dynamic programming formulations.

All algorithms are tested in simulated networks of Southeast-Michigan and Los Angeles, CA freeways and highways using historical traffic data from the Michigan ITS Center, Traffic.com, and Caltrans PEMS.

AUTOBIOGRAPHICAL STATEMENT

Ali Guner received the B.S. degree from Yildiz Teknik University, Istanbul in 2004, the M.S. degree from Fatih University, Istanbul in 2006, and the Ph.D. degree from Wayne State University, Detroit in 2011 all in Industrial Engineering. His research interests are the applications of operations research techniques to the transportation, logistics and supply chain problems.

During his studies at Wayne State University, Ali worked on MI-OH University Transportation Center projects which aim to improve the efficiency of supply chains in the Michigan-Ohio region. Specifically, he, together with other researchers of the projects, was looking into the benefits of intelligent transportation systems to avoid congestion on road networks.

He made a number of technical presentations at INFORMS and IIE annual meetings and several transportation conferences. His articles have been published in journals like Computers &Operations Research, Lecture Notes in Computer Science, and J Artificial Evolution Apps. He is awarded in 2010 and 2011 by Michigan Intelligent Transportation Society for his papers as a student. He is a member of INFORMS and IIE.