CIVIL ENGINEERING STUDIES
Illinois Center for Transportation Series No. 013-011
UILU-ENG-2013-2006
ISSN: 0197-9191

TRAVEL RELIABILITY INVENTORY FOR CHICAGO

Prepared By
Yu (Marco) Nie
Xing Wu
Qianfei Li
Northwestern University

Peter Nelson
University of Illinois at Chicago

Research Report No. FHWA-ICT-013-011

A report of the findings of
ICT-R27-84
Develop Travel Reliability Inventory for Highway Networks

Illinois Center for Transportation

April 2013

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
FHWA-ICT-13-011		

4. Title and Subtitle	5. Report Date
Travel Reliability Inventory for Chicago	April 2013
	6. Performing Organization Code
	8. Performing Organization Report N o.
7. Author(s) Yu (Marco) Nie, Xing Wu, Qianfei Li, and Peter Nelson	ICT-13-011 UILU-ENG-2013-2006

9. Performing Organization Name and Address	10. Work Unit (TRAIS)
Northwestern University, Office for Sponsored Research 633 Clark St, Crown - North Tower #2 Evanston, IL 60208	11. Contract or Grant No. R27-84
	13. Type of Report and Period Covered

12. Sponsoring Agency Name and Address	
Illinois Department of Transportation Bureau of Materials and Physical Research 126 E. Ash Street Springfield, IL 62704	14. Sponsoring Agency Code

15. Supplementary Notes

16. Abstract

The overarching goal of this research project is to enable state DOTs to document and monitor the reliability performance of their highway networks. To this end, a computer tool, TRIC, was developed to produce travel reliability inventories from various traffic data sources. In due course, travel reliability inventories will provide necessary inputs to next-generation transportation decision-making tools that are sensitive to travel reliability. TRIC incorporates reliable routing algorithms to construct and rank travel reliability indexes on routes between any given origin-destination pair in the network. It also implements a basic graphical user interface for creating and visualizing travel reliability indexes and archiving them in ESRI shape format, which can be read by most GIS (geographic information system) software. Case studies using data from the Chicago area were conducted demonstration purposes.

17. Key Words	18. Distribution Statement
Travel reliability inventory; reliable routing; geographic information system; case study	No restrictions. This document is available to the public through the National Technical Information Service, Springfield, Virginia 22161.

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	57	

Form DOT F 1700.7 (8-72) **Reproduction of completed page authorized**

ACKNOWLEDGMENT, DISCLAIMER, MANUFACTURERS' NAMES

This publication is based on the results of ICT-R27-84, Develop Travel Reliability Inventory for Highway Networks. ICT-R27-84 was conducted in cooperation with the Illinois Center for Transportation; the Illinois Department of Transportation, Division of Highways; and the U.S. Department of Transportation, Federal Highway Administration.

Members of the Technical Review Panel are the following:

Jeff Galas, TRP Chair, Illinois Department of Transportation

EXECUTIVE SUMMARY

Travel reliability is a critical performance dimension of transportation systems and services. It enables people and firms to make better use of available resources, including time, through effective scheduling of personal and business activities. Shippers and freight carriers need predictable travel times to fulfill on-time deliveries and other commitments to remain competitive. The ability to arrive on time, with high reliability, is imperative to emergency responders. However, urban transportation systems are affected by uncertainties of various sorts, which can be broadly classified as those affecting the supply of transportation (e.g., weather, accidents, natural and manmade disasters) and those associated with the demand for transportation (e.g., travel and activity behavior, special events). Taken individually or in combination, these factors could adversely affect and perturb the quality of transportation services. Travel behavior researchers have established that unanticipated long delays on highways typically produce much worse frustration among motorists than “predictable” ones.

Currently, state and local agencies neither archive travel reliability data nor have access to modeling tools that properly account for unreliability of travel times in the planning practice. Integrating travel reliability into methods of transportation network analysis presents a pressing challenge that is of both theoretical and practical importance. However, a prerequisite for the development and application of such tools is the availability of travel reliability data.

To this end, this project developed necessary procedures and a computer tool named TRIC to systematically document travel reliability information for highway networks. The travel reliability inventory created by TRIC consists of various reliability measures on two types of facilities: individual road segments and key routes between selected origin-destinations (O-D) pairs. The four reliability measures recommended by the Federal Highway Administration and considered in this research were 90th or 95th percentile travel time, buffer index, planning time index, and frequency that congestion exceeds some expected threshold. The research approach used to create these measures includes the following three steps:

1. Develop empirical travel time distributions on individual road segments in the study area from available data sources, namely, the traffic data archived by the GCM database between 2004 and 2008.
2. Compute travel time distributions on key routes for heavily traveled corridors and origin-destinations.
3. Calculate corresponding reliability measures from road, route, and O-D travel time distributions.

An important step in producing the travel reliability inventory (Step 2 of the research approach) is to establish empirical travel time distributions between O-D pairs of interest. Unlike individual road segments, however, travel time data between O-D pairs are not directly available in most cases. Consequently, one has to analyze travel reliability between O-D pairs using proper routing algorithms that not only compute travel time distribution on routes that connect the O-D pair but also rank the routes according to their travel time distributions. At the core of this analysis is finding *a priori* paths that are the shortest to ensure a specified probability of on-time arrival. This problem, known as the reliable a priori shortest path problem (RASP), is difficult to solve because each O-D pair in a real network is

connected by a huge number of routes, and enumerating all of them is virtually impossible. This project implements a RASP algorithm revised for the purpose of analyzing O-D–based travel time reliability. Although the algorithm was proposed and implemented by the project's principal investigator (PI) and his co-authors in previous efforts, this report provides the technical details necessary to understand its underlying mechanism for the sake of completeness. The implementation of the algorithm is based on a software platform known as Toolkit of Network Modeling (TNM), a C++ class library developed by the PI in the previous efforts. The details of the implementation are also documented in this report, along with an introduction to the software's graphical user interface.

To validate the tool and provide guidelines for its application, case studies for the Chicago area were conducted. Examples were prepared to demonstrate how to create, visualize, and interpret travel reliability information for different O-D pairs. The case studies show how to export and use the travel reliability inventory in a GIS-friendly format.

CONTENTS

CHAPTER 1 INTRODUCTION

1.1 BACKGROUND

Tackling traffic congestion has been a great challenge for modern society (Schrank and Lomax 2011). In addition to recurrent events such as rush hour traffic, traffic congestion also arises from various disruptions, ranging from disastrous events such as terrorist attacks or earthquakes to minor incidents such as bad weather conditions, highway maintenance, or accidents. The unexpected delays resulting from these disruptions produce far greater frustration among motorists than expected delays (FHWA 2000). The U.S. Federal Highway Administration (FHWA 2000) estimates that 50% to 60% of congestion delays are non-recurrent, and the percentage is even higher in smaller urban areas.

Currently, most existing decision-making tools for routing assume that road travel times are deterministic. The free-flow travel time is used to compute optimal routes. Even when the stochastic nature of the system is acknowledged, the mean value is usually employed as the nominal travel time when computing optimal routes. However, neither the free-flow travel time nor the mean travel time reflects travel time reliability.

Travel time reliability is significant to many highway users (Cambridge Systematics and TTI 2005). Personal and business travelers value reliability because it allows them to make better use of their time. Shippers and freight carriers require predictable travel times to remain competitive. The lack of travel reliability forces motorists to choose between running the risk of being late (therefore missing important appointments or just-in-time deliveries) or budgeting a large buffer time, much of which is often wasted.

Figure 1.1 shows an example from the Chicago area. The right panel in the figure displays the empirical distribution of travel times observed during weekday morning rush hour on a stretch of freeway that connects downtown Chicago to O'Hare International Airport (shown in the left panel), the second busiest airport in the United States. The travel times vary from as low as about 15 minutes to as long as 80 minutes in that period. In light of the magnitude of the variance, it is not surprising that travel time estimates based on the free-flow travel time or the mean travel time are at risk of being wildly inaccurate. For example, the figure shows that if a traveler wishes to have a 90% chance of catching a flight on time, 48 minutes must be budgeted for travel, which is more than 50% greater than the mean travel time (31 minutes).

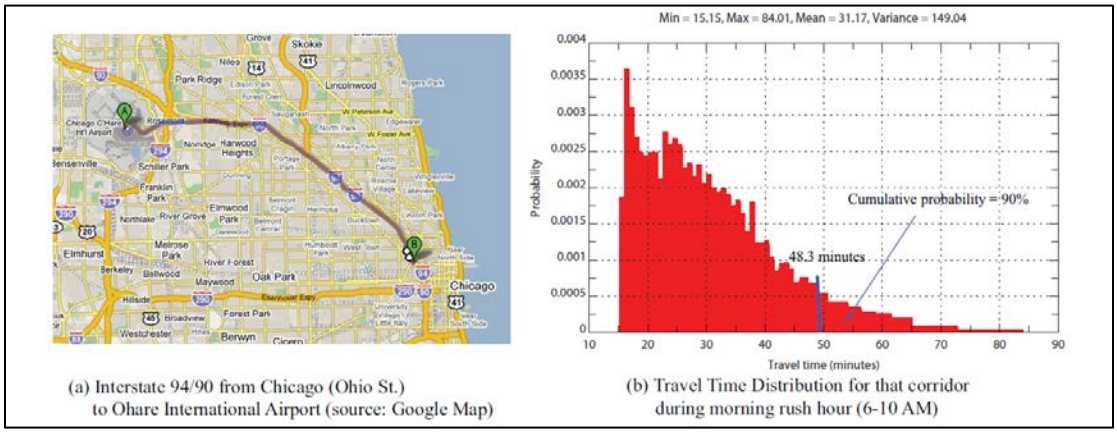


Figure1.1 An illustration of travel time variances in the Chicago area.

To facilitate decision-making by travelers to hedge against that uncertainty, it is important to develop, implement, and validate new decision-supporting tools that make the best use of various sources of data to (1) reveal and document random patterns of travel times on highway networks and (2) provide a real-time routing decision-making tool that takes uncertainty and motorist requirements for travel time reliability into account.

For the first objective, we constructed the travel time distributions for expressway links for different times of a day (morning peak, evening peak, and midday), week (weekday and weekend), and season (spring, summer, fall and winter) using traffic data collected by loop detectors and at I-PASS plazas in previous research (Nie et al. 2009). We also built the travel time distributions for arterial roads and local streets at different times of a day and week by estimating from traffic assignment data (Nie et al. 2009) and Chicago Transit Authority (CTA) bus data (Nie et al. 2010). Given the travel time distribution of each link in a network, the second objective can be addressed: providing a reliable routing model that considers uncertainty and motorist requirements for travel time reliability.

A reliable routing model would enhance mobility and allow travelers to make better use of their time, in that it would help avoid overly conservative time budgeting. Reliable routing is also useful to freight carriers and parcel delivery firms whose trucks must move through peak-period traffic and work zones on a regular basis. A reliable routing model would allow carriers to evaluate alternative routing plans for their fleets against the likelihood of on-time delivery, which is often an important criterion of level of service in the trucking industry.

Travel time reliability has various definitions (see Chapter 2 for more details). In our research, travel time reliability is defined as the probability that a motorist can arrive at the destination on time within the given time budget. In this project, we proposed a "reliable *a priori* shortest path" (RASP) problem, which aims to find the shortest *a priori* paths that can ensure a specified probability of on-time arrival.

Based on real traffic data collected in the Chicago metropolitan area (Nie et al. 2009), we developed software called the Travel Reliability Inventory for Chicago (TRIC). This software provides data, functions, and decision-making tools at three levels:

4. At the link level, it provides statistics information on expressway links in the Chicago metropolitan network, such as the link travel time distributions in a given time period (e.g., the morning peak on weekdays in spring).
5. At the origin-destination (O-D) pair level, the software offers a function to identify reliable routes between an O-D pair. The connection and statistics information of the identified reliable routes is provided, such as a list of links traversed by a route, the travel time distribution in a given time period, and the 95th and 50th percentile travel times of the reliable routes. It also provides a tool to compare the travel time distributions of identified reliable routes.
6. At the network level, the software gives the topology of the network and provides a visualized and dynamic view of travel speed changes on all expressway links in the entire network during a specified day, which reflects real traffic conditions.

TRIC not only can help motorists plan trips according to a desired travel reliability, but it also can help transportation planning agencies calculate the travel reliability for some corridors in Chicago, such as from downtown Chicago to O'Hare airport or from the North Shore to the south suburbs, so as to establish better transportation planning and management policies. Moreover, TRIC can archive all information on links and reliable routes as shape files that can be used in GIS application packages. Chapters 4 and 5 provide more-detailed descriptions of TRIC.

1.2 RESEARCH OVERVIEW

A small example is employed here to demonstrate the concept of the RASP problem. Figure 2 shows the cumulative density functions (CDFs) of the random travel times for two routes. For the cumulative probability α, we have $P(X \le t) = \alpha$, where X refers to the random travel time and t refers to the desired travel time budget. α can be regarded as an on-time arrival probability. From Figure 2, it is shown that given α, Paths 1 and 2 provide time budgets t_1 and t_2 (i.e., α-percentile travel times), respectively, so that the probability of being late ($P(X > t)$) will not exceed α. $t_1 < t_2$ indicates that Path 1 is a better choice. In other words, Path 1 provides a later departure time than Path 2 to ensure a desired probability of not being late.

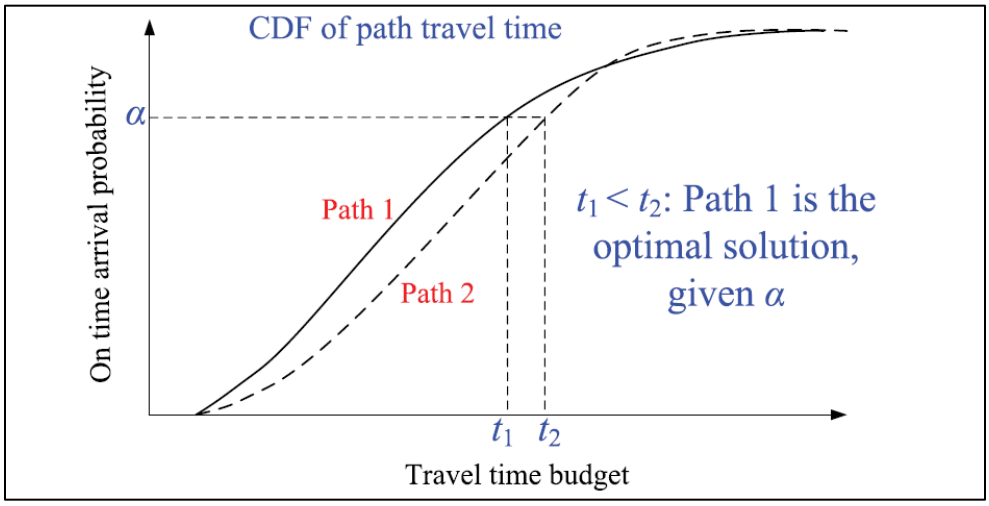


Figure 1.2 An illustrative example of the RASP problem.

The optimal path can be found by enumerating all paths and comparing the distributions of path travel times, as shown in the above example. However, enumerating all paths is not practicable because it is difficult if not impossible to perform for large networks. Instead, the RASP problem can be formulated as a general dynamic programming (GDP) problem that identifies non-dominated paths by using the first-order stochastic dominance theory. The reliable shortest paths are included in the set of non-dominated paths. We show that Bellman's principle of optimality can be applied to construct non-dominated paths. Acyclicity of non-dominated paths is established and used for proving finite convergence of solution procedures. A label-correcting algorithm is proposed to solve the RASP problem, and the complexity of the algorithm is analyzed.

If the distributions are continuous, the computation of the convolution integral is difficult. To avoid this problem, link travel time distributions must be discretized. An adaptive discretization approach (ADA) was proposed. The process of computing the convolution integral strongly affects the computational performance of the solution algorithm. We also developed some convolution schemes based on the discrete Fourier transform (DFT). The DFT-based convolution schemes were compared with the ADA-based direct convolution methods through numerical examples.

1.3 ORGANIZATION

The rest of this report is organized as follows: Chapter 2 briefly reviews the literature on reliable routing models and the stochastic dominance theory. Chapter 3 presents the existing formulation and solution algorithms for the reliable routing problem and proposes techniques to improve computational performance. Chapter 4 first introduces the Toolkit of Network Modeling (TNM), a C++ class library for solving various transportation network problems, and then gives a detailed description of the Travel Reliability Inventory for Chicago (TRIC). Chapter 5 presents several case studies to help users better understand TRIC so as to improve decision-making processes. Chapter 6 concludes this report.

CHAPTER 2 LITERATURE REVIEW

In this chapter, we briefly review the literature of reliable routing models and the stochastic dominance theory, then we introduce the notation for describing them.

2.1 NOTATION

Consider a directed and connected network $G(N, A, P)$ consisting of a set of nodes N with the number of nodes $|N| = n$, a set of links A with the number of links ($|A| = m$), and a probability distribution P describing the variation in link travel times. The analysis time period is set to $[0, T]$. Let the destination of routing be node s and the desired arrival time be aligned with the end of the analysis period T. The travel times on different links (denoted as c_{ij}) are assumed to be random variables with a probability density function $p_{ij}(\cdot)$. Let $F_{ij}(\cdot)$ be the cumulative density function (CDF) of c_{ij}. For brevity, other notations used in the proposal are summarized in Table 2.1.

Table 2.1 Notation

Network (node, link, path and OD pairs):	
$\mathcal{A}, \mathcal{N}, \mathcal{D}, \mathcal{A}(i)$	set of links, nodes and OD pairs and set of outgoing links from node i, respectively
n, m	number of nodes and links, respectively
rs	OD pair between the origin r and destination node s, $rs \in \mathcal{D}$
k^{rs}, K^{rs}, K	path k from node r to node s, then $K^{rs} = \{k^{rs}, \forall k\}, \forall rs$, and $K = \cup_{rs \in \mathcal{D}} K^{rs}$
$\bar{k}^{rs,\alpha}$	path with minimum travel time budget for desired on time arrival probability α (sometimes superscript α is ignored) for OD pair rs
Γ^{rs}_{FSD}	set of FSD-admissible paths between r and s
Travel time, service rate and flow of links or paths:	
$c_{ij}, \hat{c}^{\alpha}_{ij}$	random travel time and α-percentile travel time on link ij, respectively
$\pi^{rs}_k, \hat{\pi}^{rs,\alpha}_k$	random travel time and α-percentile travel time on path k^{rs}
b, T	travel time budget and maximum travel time budget
ϕ, ϵ	resolution of travel time and probability, respectively
$k^{js} \diamond ij$	extension of path k^{js} along link ij
Probability (link distributions are static and independent):	
X, Y	random variables
$\mathrm{E}(X), \mathrm{Var}(X)$	mean and variance of random variable X, respectively
$F_X(\cdot), p_X(\cdot)$	cumulative density function (CDF) and probability density function (PDF) of X
$F_{ij}(\cdot), p_{ij}(\cdot)$	CDF and PDF of travel time on link ij, respectively
$u^{rs}_k(\cdot), p^{rs}_k(\cdot)$	CDF and PDF of path travel time π^{rs}_k, respectively
$v^{rs}_k(\cdot)$	inverse function of $u^{rs}_k(\cdot)$
$u^{rs}_{\text{FSD}}(b)$	$u^{rs}_{\text{FSD}}(b) = \max(u^{rs}_k(b), \forall k^{rs} \in \Gamma^{rs}_{\text{FSD}})$
$v^{rs}_{\text{FSD}}(\alpha)$	$v^{rs}_{\text{FSD}}(\alpha) = \min(v^{rs}_k(\alpha), \forall k^{rs} \in \Gamma^{rs}_{\text{FSD}})$
Other:	
$U(\cdot)$	utility function

2.2 STOCHASTIC ROUTING PROBLEM

The routing problem aims to direct vehicles from an origin to a destination along a path that is considered optimal in some way or another. Depending on whether the guidance is coordinated by a central control unit, the problem can be classified as centralized or

decentralized. The routing problem can also be labeled as adaptive or *a prior*i, according to whether en-route re-routing is allowed. Two other factors that are often used in problem classification are dynamics (i.e., if travel time varies over time) and uncertainties (i.e., if travel time is random). This research considers decentralized, *a priori* routing problems for stochastic networks (our routing model can be applied to dynamic networks, but we usually do not include them). The focus is to incorporate travel reliability as an integrated objective of routing.

Where uncertainties are concerned, optimal routing, either adaptive or *a priori*, has many different meanings. A classic definition considers a routing strategy optimal if it incurs the least expected travel time (LET) (Hall 1986; Polychronopoulos and Tsitsiklis 1996; Fu and Rilett 1998; Cheung 1998; Miller-Hooks and Mahmassani 2000; Miller-Hooks 2001; Fu 2001; Waller and Ziliaskopoulos 2002; Provan 2003; Gao and Chabini 1971; Fan et al. 2005a). Clearly, the LET path may not properly weigh travel time reliability because it overlooks travel time variations. This concern gives rise to the reliability-based routing problems.

Table 2.2 classifies previous studies on the stochastic routing problem into four categories, using two criteria. Our focus is the right bottom cell (i.e., reliability-based *a priori* routing problem).

Table 2.2 Various Definitions of Stochastic Optimal Paths

	LET-Based	**Reliability-Based**
Adaptive	Cheung 1998; Fu 2001; Miller-Hooks 2001; Provan 2003; Waller and Ziliaskopoulos 2002; Polychronopoulos and Tsitsiklis 1996; Gao and Chabini 2006	Bander and White 2002; Fan et al. 2005a
A priori	Hall 1986; Fu and Rilett 1998; Miller-Hooks and Mahmassani 2000; Fan et al. 2005b	Frank 1969; Sigal et al. 1980; Loui 1983; Sivakumar and Batta 1994; Yu and Yang 1998; Montemanni and Gambardella 2004; Bard and Bennett 1991; Miller-Hooks 1997; Miller-Hooks and Mahmassani 1998b, 2003

Reliability-based stochastic routing has been studied extensively, with the majority of the literature focused on *a priori* path problems. Frank (1969) defined the optimal path as the one that maximizes the probability of the travel time equal to or less than a given threshold T, i.e.,

$$\bar{k}^{rs} = \arg\left(\max_{\forall k^{rs} \in K^{rs}} \{P(\pi_k^{rs} \le T)\} \right) \tag{2.1}$$

Note that the distributions of all link travel times are continuous. Let $C = (c_{12}, \cdots, c_{ij}, \cdots)$ and $\Pi^{rs} = (\pi_1^{rs}, \cdots, \pi_k^{rs}, \cdots)$ be vectors of link travel times and travel times of paths between an origin-destination (O-D) pair rs, respectively. The joint distribution of C is known. Using the characteristic functions of C and Π^{rs}, the joint distribution of Π^{rs} can be obtained. Therefore, the optimal path can be identified. This method, however, is not

practicable because (1) the calculation of convolution integrals is complicated, and (2) it requires that all paths be enumerated.

Mirchandani (1976) presented a recursive algorithm to solve a discrete version of Equation 2.1. However, the algorithm is suitable only for small problems because it requires not only that all paths be enumerated but also all travel time possibilities for each path through a network expansion be enumerated.

Sigal et al. (1980) suggested using the probability of a path being the shortest as an index to define the optimal path. For path l, the optimality index R_l is defined as

$$R_l = P(\pi_l \leq \pi_k), \forall k \neq l \tag{2.2}$$

To calculate R_l, a multi-dimensional integral must be evaluated. To evaluate the integral, it is necessary to obtain the set of links that are used by more than one path, as well as the joint distribution of these links. Determining such a set requires enumerating all paths.

Expected utility theory has also been used to define path optimality.

Loui (1983) showed that the Bellman's principle of optimality could be used to find an optimal path when utility functions are affine or exponential. This restriction was also noticed by Eiger et al. 1985. For a general polynomial and monotonic utility function, Loui's expected utility problem can be reduced to a bi-criterion (mean and variance) shortest path problem. In effect, a traveler is allowed to trade off the expected value and variance using a generalized dynamic program (GDP) (see, e.g., Carraway et al. 1990) based on Pareto optimality (or non-dominance relationship).

More general non-linear utility functions may be approximated by piecewise linear functions (Murthy and Sarkar 1996, 1998) who also proposed a few efficient solution procedures based on relaxation. The mean-variance trade-off has been treated in other ways. For example, Sivakumar and Batta (1994) added an extra constraint into the shortest path problem to ensure that the identified LET paths have a variance smaller than a benchmark. In Sen et al. (2001), the objective function of stochastic routing is a parametric linear combination of mean and variance. In either case, GDP cannot be applied. Instead, non-linear programming solution techniques must be used.

Stochastic routing has also been discussed in the context of robust optimization, that is, a path is optimal if its worst-case travel time is the minimum (Yu and Yang 1998; Montemanni and Gambardella 2004; Bertsimas and Sim 2003). Depending on the setting, such robust routing problems are either NP-hard (Yu and Yang 1998; Montemanni and Gambardella 2004) or solvable in polynomial time (Bertsimas and Sim 2003).

Bard and Bennett (1991) defined the optimal path as the one that maximizes the expected utility in a stochastic acyclic network. Compared with the study of Loui (1983), where utility functions have to be polynomial and monotonic, Bard and Bennett (1991) required only that the utility functions be non-linear and monotonic. To find the global optimal path, all paths have to be enumerated. To improve computational efficiency, they attempted to reduce the network size by using the theory of first-order stochastic dominance (see Section 2.3.2 for details). Specifically, if a path is dominated, the links used by this path are elected for possible elimination. In Bard and Bennett's (1991) paper, the first-order stochastic dominance restraint is relaxed as post-median stochastic dominance; that is, only the points on the tail of CDF (after the median) are checked, in order to eliminate more paths. It is shown that through reduction, 90% of paths in an acyclic network are eliminated,

which makes path enumeration feasible. This approach, however, is applicable only to acyclic networks.

Miller-Hooks (1997) and Miller-Hooks and Mahmassani (2003) also employed first-order stochastic dominance to define path optimality. Besides first-order stochastic dominance, they also defined two types of path dominance: (1) deterministic dominance and (2) expected value dominance. Label-correcting algorithms were proposed to find non-dominant paths under the path dominance rules. Recognizing that the exact algorithm does not have a polynomial bound, heuristics were considered by Miller-Hooks (1997), who attempted to limit the size of the retained non-dominant paths by a predetermined number. As noted by Miller-Hooks (1997), however, those heuristics might not identify any non-dominant paths.

Reliability has also been defined using the concept of connectivity (Chen et al. 2006; Kaparias et al. 2007). This approach models reliability as the probability that the travel time on a link is greater than a given threshold. Accordingly, the reliability of a path is the product of the reliability of links used by that path (assuming independent distributions). A software tool known as ICNavS was developed based on this approach (Kaparias et al. 2007).

2.3 STOCHASTIC DOMINANCE

2.3.1 Conventional Theory of Stochastic Dominance

Bard and Bennett (1991), Miller-Hooks (1997), and Miller-Hooks and Mahmassani 2003) all employed first-order stochastic dominance (SD) to find reliable *a priori* shortest paths. In fact, SD theory has been extensively used in finance and economics to rank random variables when their distributions are known (Hanoch and Levy 1969; Hadar and Russell 1971; Rothschild and Stiglitz 1970; Whitemore 1970; Bawa et al. 1983; Muller and Stoyan 2002; Dentcheva and Ruszczynski 2003). Conventionally, SD theory is established by assuming that the utility function is increasing. We provide the following definitions.

Definition 2.1 (*First-order stochastic dominance (FSD)* $\succeq^1$)

A random variable X dominates another random variable Y in the first order, denoted as $X \succeq^1 Y$ if $F_X(t) \le F_Y(t), \forall t$ and $F_X(t) < F_Y(t)$ for some t, where F_X is the cumulative density function (CDF) of random variable X.

Definition 2.2 (*Second-order stochastic dominance (SSD)* $\succeq^2$)

A random variable X dominates another random variable Y in the second order, denoted as $X \succeq^2 Y$ if $\int_{-\infty}^{t} F_X(w)dw \le \int_{-\infty}^{t} F_Y(w)dw, \forall t$ and $\int_{-\infty}^{t} F_X(w)dw < \int_{-\infty}^{t} F_Y(w)dw$ for some t.

According to the random utility theory, random variable X being preferred to Y implies that X has a higher expected utility. SD theory provides not only a tool to rank random variables, but it also explains the ranking within the framework of utility theory, corresponding to utility function $U(\cdot)$ (see, e.g., Levy and Hanoch 1970).

Theorem 2.1 $X \succeq^1 Y$ if and only if $E[U(X)] \ge E[U(Y)]$ for any non-decreasing utility function $U(\cdot)$, i.e., $U' \ge 0$.

Theorem 2.2 $X \succeq^2 Y$ if and only if $E[U(X)] \geq E[U(Y)]$ for any non-decreasing and concave utility function $U(\cdot)$, i.e., $U' \geq 0$ and $U'' \leq 0$.

See Bawa 1975 and Heyer 2001 for proofs of the above theorems.

In the above definitions and theories, the symbol $\succeq$ z is used. Strictly speaking, $X \succeq Y$ means that X dominates Y or X is equivalent to Y. However, in this report, we consider strictly dominate only, i.e., $\succ$. Hence, $X \succ Y$ if and only if $E[U(X)] > E[U(Y)]$.

Theorem 2.1 implies that any insatiable decision maker who is never worse off with more quantities of interest ($U' > 0$) prefers X to Y if $X \succ^1 Y$. Theorem 2.2 is for insatiable and risk-averse decision makers because any concave utility function ($U'' < 0$) implies risk aversion (Friedman and Savage1948).

2.3.2 Stochastic Dominance for Decreasing Utility Functions

Definitions 2.1 and 2.2 are both based on the circumstance that decision makers are never worse off with more quantities of the random variable of interest. That is, their utility function is non-decreasing with respect to the quantity. However, as travel time is concerned, travelers usually prefer shorter travel times to longer ones. That is, if the utility depends on travel time, the utility function is decreasing, i.e., $U' < 0$. In this case, the stochastic dominance has to be re-defined (Wu and Nie 2011).

Definition 2.3 (*FSD for decreasing utility functions* $\succ_1$) A random variable X dominates another random variable Y in the first order under decreasing utility functions, denoted as $X \succ_1 Y$, if $F_X(t) \geq F_Y(t), \forall t$ and $F_X(t) > F_Y(t)$ for some t.

Definition 2.4 (*SSD* $\succ_2$) A random variable X dominates another random variable Y in the second order, denoted as $X \succ_2 Y$, if $\int_t^T F_X(w)dw \geq \int_t^T F_Y(w)dw, \forall t$ and $\exists$ at least an open interval $\Lambda \in [0,T]$ with a non-zero Lebesgue measure such that

$$\int_t^T F_X(w)dw > \int_t^T F_Y(w)dw, \forall t \in \Lambda .$$

When stochastic dominance is defined in Definitions 2.3 and 2.4, Theorems 2.1 and 2.2 become

Theorem 2.3 X dominates Y in the first order, i.e., $X \succ_1 Y$, if and only if $E[U(X)] > E[U(Y)]$ for any decreasing utility function $U(\cdot)$, i.e., $U' < 0$.

Theorem 2.4 X dominates Y in the second order, i.e., $X \succ_2 Y$, if and only if $\mathrm{E}[U(X)] > \mathrm{E}[U(Y)]$ for any U such that $U' < 0, U'' < 0$.

See Wu and Nie (2011) for the proof of Theorems 2.3 and 2.4.

Given that the utility function still has to be concave in Theorem 2.4, the second-order stochastic dominance (SSD) for decreasing utility functions is still associated with risk aversion (Friedman and Savage 1948).

CHAPTER 3 FORMULATION AND SOLUTION ALGORITHM

In this chapter, we introduce the mathematical formulation and solution algorithm for the reliable *a priori* shortest path (RASP) problem, which aims to find *a priori* paths that are shortest to ensure a specified probability of on-time arrival.

A priori path refers a non-adaptive path that is independent of past experience. In Section 1.2, an example (Figure 1.2) was created to illustrate the RASP problem. Given an on-time arrival probability α, the path is optimal if its random path travel time has the least α-percentile value. If X is a random variable, let X^{α} refer to the α-percentile value of X, where α is the cumulative probability. The RASP problem can be formulated as the following stochastic programming:

$$\min_{x_{ij}} \left(\sum_{\forall ij \in A} x_{ij} c_{ij} \right)^{\alpha} \tag{3.1}$$

$$\text{s.t.} \sum_{j:ji \in A} x_{ji} - \sum_{j:ij \in A} x_{ij} = d_i, \forall i \in N \tag{3.2}$$

$$x_{ij} = \{0,1\} \tag{3.3}$$

where α is a given probability, $0 \le \alpha \le 1$; X^{α} is the α percentile value of random variable X; d_i denotes the total shipping demands at node i (note that $d_r = -1, d_s = 1$ and $d_i = 0, \forall i \notin \{r,s\}$), where r and s refer to the origin and destination node respectively; c_{ij} represents the random travel time on link ij; and $x_{ij}, \forall ij \in A$ is the unknown variable: link ij is selected if $x_{ij} = 1$, and otherwise if $x_{ij} = 0$. Constraint 3.2 ensures the connectivity of a path from origin r to destination s.

3.1 FORMULATION BASED ON FIRST-ORDER STOCHASTIC DOMINANCE

This section aims to show that the RASP paths can be found from the Pareto frontier of all non-dominated paths under first-order stochastic dominance (FSD). To this end, we first define first-order dominance relationship as follows:

Definition 3.1 (*Path dominance)* Let π_k^{rs} and π_l^{rs} be the random travel times on path k^{rs} and l^{rs}, respectively. If π_k^{rs} dominates π_l^{rs} in the first order, i.e., $\pi_k^{rs} \succ_1 \pi_l^{rs}$, path k^{rs} is said to dominate path l^{rs} in the first order.

Definition 3.2 (*FSD-admissible paths)* A path k^{rs} is FSD admissible if no path in K^{rs} can dominate path k^{rs} in the first order, where K^{rs} is the set of all paths from node r to node s.

Then we have the following definition:

Definition 3.3 (*FSD-optimal paths)* An FSD-admissible path $\bar{k}^{rs}$ is FSD optimal if, for at least one time budget b, no other path can provide a higher on-time arrival probability.

We denote the set of FSD-admissible paths from node r to destination s by Γ_{FSD}^{rs}. Admissible paths are also called non-dominated paths in the literature (Miller-Hooks 1997; Miller-Hooks and Mahmassani 1998a, 2003).

If $u_k^{rs}(b)$ is the cumulative density function (CDF) of the travel time on path k^{rs}, we define

$$u_{\text{FSD}}^{rs}(b) = \max_{\forall k^{rs} \in \Gamma_{\text{FSD}}^{rs}} \{u_k^{rs}(b)\}, \quad \forall b \in [0,T] \tag{3.4}$$

as the Pareto frontier. A Pareto frontier has the following properties: (1) no feasible solution exists beyond the Pareto frontier, (2) all FSD-optimal solutions are located on the Pareto frontier of non-dominated paths, and (3) the solutions inside the Pareto frontier must not be optimal. If a path k^{rs} is FSD optimal, $\bar{k}^{rs} = argmax[u_k^{rs}(b), \forall k^{rs} \in K^{rs}]$ at b, given a time budget b, path $\bar{k}^{rs}$ ensures the highest on-time arrival probability among all paths from node r to node s. In other words, the solution to the RASP problem is always located on the Pareto frontier described in Equation 3.4.

The above analysis employs $u_k^{rs}(\cdot)$ that gives an on-time arrival probability α based on a specific travel time budget b. Conversely, we can define the optimality with a given on-time arrival probability α. Denote $v_k^{rs}(\cdot)$ as the inverse function of $u_k^{rs}(\cdot)$, and an alternative Pareto frontier $v_{\text{FSD}}^{rs}(\cdot)$ and the FSD-optimal path $\bar{k}^{rs}$ are respectively defined as

$$v_{\text{FSD}}^{rs}(\alpha) = \min_{\forall k^{rs} \in \Gamma_{\text{FSD}}^{rs}} \{v_k^{is}(\alpha)\}, \quad \forall \alpha \in [0,1] \tag{3.5}$$

$$\bar{k}^{rs} = argmin[v_k^{rs}(\alpha), \forall k^{rs} \in K^{rs}], \quad \forall \alpha \in [0,1] \tag{3.6}$$

argmax[u $^{rs}(b_0)$], and $v_{\bar{k}}^{rs}(\alpha_0) = v^{rs}(\alpha_0) = b_0$.

From Definitions 3.2 and 3.3, we have the following:

Proposition 3.1 An FSD-admissible path is not necessarily FSD optimal.

We use Figure 3.1 to demonstrate Proposition 3.1. There are three paths: 1, 2, and 3. The CDFs of the three paths' travel times are shown in the figure. The three paths are all FSD-admissible paths because no path can dominate the other(s) in the first order. The Pareto frontier is marked by the thick line. However, $u_3^{rs}(\cdot)$ does not contribute to the Pareto frontier. Therefore, Path 3 is not an FSD-optimal path.

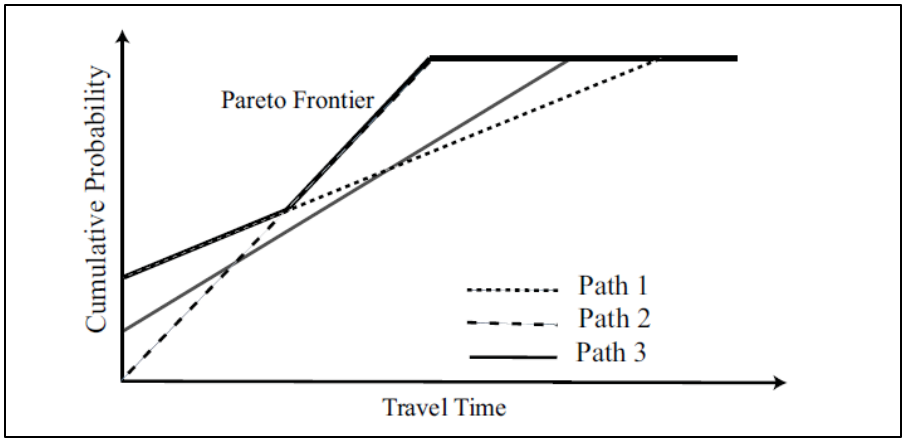


Figure 3.1 An FSD-admissible path that is not FSD optimal.

Let $p_{ij}(\cdot)$ be the probability density function (PDF) of travel time on link ij. If the distribution of link traversal time is continuous and independent, $u_k^{is}(b)$ can be recursively calculated by

$$u_k^{is}(b) = \int_0^b u_k^{js}(b-w)\, p_{ij}(w)dw, \forall i \neq s \tag{3.7}$$

$$u_k^{ss}(b) = 1 \tag{3.8}$$

Equation 3.8 is the boundary condition.

FSD-admissible paths have two important properties:

Proposition 3.2 (*i) Subpaths of FSD-admissible paths must also be FSD-admissible paths; (ii) FSD-admissible paths must not contain any cycle.*

Proof. To prove the first property, suppose path $k^{is}, i \neq s$ is FSD admissible, but its subpath at node j, $j \neq s$ k^{js} ($k^{is} = k^{js} \lozenge ij$) is not FSD admissible. Therefore, there must exist a path l^{js} such that $l^{js} \succ_1 k^{js}$. Therefore, we have $u_l^{js}(b) \geq u_k^{js}(b), \forall b$, and $u_l^{js}(b) > u_k^{js}(b)$ for some b.

According to Equation 3.7, we have

$$\begin{cases} u_l^{is}(b) &= \int_0^b u_l^{js}(b-w)\, p_{ij}(w)dw \\ u_k^{is}(b) &= \int_0^b u_k^{js}(b-w)\, p_{ij}(w)dw \end{cases} \tag{3.9}$$

then

$$u_l^{is}(b) - u_k^{is}(b) = \int_0^b [u_l^{js}(b-w) - u_k^{js}(b-w)]\, p_{ij}(w)dw \geq 0, \forall b, \text{and} > 0, \exists b$$

It implies $l^{is} \succ_1 k^{is}$. It contradicts the assumption that k^{is} is FSD admissible.

To prove the second property, without loss of generality, suppose that paths k^{rs} and l^{rs} are FSD admissible. l^{rs} contains one and only one cycle starting at node i, $i \neq r, s$; while k^{rs} is acyclic. Then according Equation 3.7, assuming the cycle from node i to node i is reduced to a link $\tilde{i}i$, u_l^{is} is calculated as

$$u_l^{is} = \int_0^b u_k^{is}(b-w)\, p_{\tilde{i}i}(w)dw$$

Then we have $u_l^{is}(b) \leq \int_0^b u_k^{is}(b)\, p_{\tilde{i}i}(w)dw = u_k^{is}(b)\int_0^b p_{\tilde{i}i}(w)dw \leq u_k^{is}(b), \forall b$. The first inequality is due to the monotonicity of CDF. The second inequality holds because $p_{\tilde{i}i}(\cdot)$ is a probability density function, $\int_0^\infty p_{\tilde{i}i}(w)dw = 1$, and $\exists b < \infty$, so $\int_0^b p_{\tilde{i}i}(w)dw < 1$. Therefore, $u_k^{is}(b) \geq u_l^{is}(b), \forall b$ and $u_k^{is}(b) > u_l^{is}(b)$ for some b (see Definition 3.1). Therefore l^{rs} has no chance to be an FSD-admissible path.

Using the first property of FSD-admissible paths, the RASP problem can be formulated as the following general dynamic programming problem:

$$\text{Find } \Gamma_{\text{FSD}}^{is}, \forall i \text{ such that } \Gamma_{\text{FSD}}^{is} = \gamma_{\succ_1}(k^{is} = k^{js} \Diamond ij \mid k^{js} \in \Gamma_{\text{FSD}}^{js}, \forall ij \in A), \forall i \neq s; \Gamma_{\text{FSD}}^{ss} = 0^{ss} \qquad (3.10)$$

where $k^{ij} \Diamond ij$ extends path k^{js} along link ij; $\gamma_{\succ_1}(K^{rs})$ represents the operation that retrieves FSD-admissible paths from a path set K^{rs} using Definition 3; 0^{ss} is a dummy path representing the boundary condition.

3.2 SOLUTION ALGORITHM

Solving the RASP problem (Equation 3.10) involves two main operations: iteratively constructing and storing admissible paths, and evaluating their travel time distributions. Central to either operation is the capability to discretize the underlying problem. We first introduce the basic algorithmic concept using a simple and easy-to-implement discretization scheme and then discuss more-sophisticated approaches.

3.2.1 Basic Discretization Scheme

The convolution integral (Equation 3.7) can be evaluated in the continuous space. For instance, Fan et al. (2005a) suggested using Laplace transform to perform a similar integral. Their method makes use of the fact that Laplace transform of a convolution equals the product of the individual transforms; that is

$$U(s) = F(s).G(s), F(s) = \int_0^\infty e^{-st} f(t)dt, G(s) = \int_0^\infty e^{-st} g(t)dt, U(s) = \int_0^\infty e^{-st} u(t)dt \qquad (3.11)$$

where $u(t) = \int_0^t f(t-w)g(w)dw$.

The evaluation of a convolution is divided into three steps. First, both functions (in Equation 3.7, they are $u_k^{is}(\cdot)$ and $p_{ij}(\cdot)$) are transformed and numerically integrated for a set of discrete s. The second step calculates the convolution of the transformed functions, which turns out a point-to-point multiplication. The resulting function is reverted to the original domain. The last step involves solving a linear system $U = Vu$ in which U and u are vectors of $U(s)$ and $u(t)$ evaluated at discrete points s and t, respectively, and V is a Vandermonde matrix. We did not adopt this method in this research because the resulting Vandermonde matrix is usually ill conditioned and the inverse operation is therefore unstable.

In most cases, it is more convenient to approximate the continuous distributions with their discrete counterparts. By assuming that the central limit theorem (CLT) always holds, which guarantees that the path travel time follows a normal distribution, the path travel time distribution can be calculated from a continuous analytical expression (Lo and Tung 2003; Lo et al. 2006). However, the normal distribution, which is symmetric, usually cannot reflect general link travel time distributions in the real world (Waller et al. 2006).

Let T be divided into L uniform discrete intervals with $L\varphi = T$ .The probability mass of interval l is then calculated as

$$P_{ij}(l) = \int_{(l-1)\varphi}^{l\varphi} p_{ij}(w)dw, l = 1,...,L. \tag{3.12}$$

Accordingly, Equation 3.9 can be discretized as

$$U_k^{is}(l) = \sum_{l'=0}^{l} U_k^{js}(l-l')P_{ij}(l'), l = 1,\cdots,L. \tag{3.13}$$

where $U_k^{is}(l)$ denotes the cumulative probability on path $k \in K^{is}$ up to the lth discrete interval. Intuitively, the quality of convolution results is controlled by the resolution of the discretization L: a larger L generally leads to greater accuracy. This scheme is called *b*-discrete because it explicitly considers discrete supports. The convolution based on Equation 3.13 is also called *b*-based direct convolution in the following text.

3.2.2 A Label-Correcting Algorithm

We now describe a label-correcting algorithm to search for all FSD-admissible paths. The algorithm is named FSD-LC, with FSD standing for first-order stochastic dominance, and LC standing for label correcting. The following definitions are used in the algorithm description: a list of candidate paths is denoted using Q; $\omega(\cdot)$ is a subpath operator used to track paths (e.g., $\omega(k^{is}) = k^{js}$ such that $k^{is} = k^{js} \Diamond ij$); Ω_{is} denotes a tentative set of FSD-admissible paths.

Algorithm FSD-LC

Step 0 Initialize. Set $Q = \varnothing$, $u^{is}(b) = 0, \forall b = 0, \varphi, .., L\varphi, \forall i \neq s, \Omega_{is} = \varnothing, \forall i$. For the destination node s , set $u^{ss}(b) = 1, \forall b = 0, \varphi, ..., L\varphi$. Create a path from s to itself, 0^{ss} and set $u_0^{ss}(b) = 1, \forall b = 0, \varphi, ...L\varphi$. Let $Q = Q \cup \{ 0^{ss} \}$.

Step 1 Check optimality. If $Q = \varnothing$, terminate the procedure. The optimal solution is found; otherwise, proceed to Step 2.

Step 2 Take the first path k^{js} stored in Q and scan every incoming link ij of node j.

step 2.1 If all links ij have been scanned, go to Step 1; otherwise, take the next link $ij \in A$.

step 2.2 Check whether path k^{js} has already traversed node i. If yes, go back to step 2.1; otherwise, proceed to step 2.3.

step 2.3 Set $l = |\Gamma_{\mathrm{FSD}}^{is}| + 1$, create a new path l^{is}, calculate Equation 3.13.

step 2.4 If $\Gamma_{\mathrm{FSD}}^{is} = \varnothing$, set $\Gamma_{\mathrm{FSD}}^{is} = l^{is}, u_{\mathrm{FSD}}^{is}(b) = u_l^{is}(b)$, $\overline{k}^{is}(b) = l^{is}$, $\forall b = 0, \varphi, ..., L\varphi$. Update $\sigma(l^{is}) = L+1, \omega(l^{is}) = k^{js}$;

otherwise, call **Procedure FSD-CHECK**. If l^{is} is not FSD admissible, go to step 2.1; otherwise, set $\omega(l^{is}) = k^{js}$ and update $Q = Q \cup \{l^{is}\}$. Go to step 2.1.

Procedure FSD-CHECK

Inputs A new path l^{is}, a set of FSD-admissible paths $\Gamma_{\mathrm{FSD}}^{is}$, as well as the associated Pareto frontier $u^{is}(\cdot)$.

Return A Boolean value LR indicating whether l^{is} is FSD admissible, and update u_{FSD}^{is} and $\Gamma_{\mathrm{FSD}}^{is}$.

Step 0 Set LR = TRUE, set $\sigma(l^{is}) = 0$, set $Q' = \varnothing$ (Q' is the set of paths that are currently FSD admissible but have a zero degree of strong dominance).

Step 1 Update the Pareto frontier and identify Q.

for each $b = 0, \varphi, ..., L\varphi$ do

set $k^{is} = \overline{k}^{is}(b)$.

If $u_l^{is}(b) > u_{\mathrm{FSD}}^{is}(b)$

update $u_{\mathrm{FSD}}^{is}(b) = u_l^{is}(b), \overline{k}^{is}(b) = l^{is}, \sigma(l^{is}) = \sigma(l^{is}) + 1, \sigma(k^{is}) = \sigma(k^{is}) - 1$

If $\sigma(k^{is}) = 0$, set $Q' = Q' \cup \{k^{is}\}$. end if

end if

end for

Step 2 Update the set of FSD-admissible paths.

while LR = TRUE and Q' is not empty, do

take path k^{is} out of Q', set $n_l = 0, n_e = 0, n_g = 0$.

for $b = 0, \varphi, ..., L\varphi$ and if ($n_l = 0$ or $n_g = 0$) do

if $u_l^{is}(b) > u_k^{is}(b)$, set $n_g = n_g + 1$;

else if $u_l^{is}(b) = u_k^{is}(b)$, $n_e = n_e + 1$;

else, $n_l = n_l + 1$.

end if

end for

if $n_l = 0$, set LR = FALSE;

else if $n_g = 0$, set $\Gamma_{\text{FSD}}^{is} = \Gamma_{\text{FSD}}^{is} / \{k^{is}\}$. end if

end while

If LR = TRUE, set $\Gamma_{\text{FSD}}^{is} = \Gamma_{\text{FSD}}^{is} \cup \{l^{is}\}$. end if. Return LR.

The following remarks are in order.

Remark 1: Although they deal with different problems, Algorithm FSD-LC is conceptually similar to the EV algorithm of Miller-Hooks and Mahmassani (2000), in which a non-dominance relationship is defined with respect to departure times instead of time budgets. However, FSD-LC promises to reduce the amount of work required to carry out the dominance check (although the strategy does not improve the worst-case scenario).

Remark 2: For each node i, Algorithm FSD-LC needs to store u^{is} and $\overline{k}^{is}$. Both are vectors of length $L+1$. Moreover, a vector u_k^{is} (length $L+1$) must be stored for each path $k^{is} \in \Gamma_{\text{FSD}}^{is}$.

To examine the complexity of Algorithm FSD-LC, the following proposition is needed.

Proposition 3.3 In Algorithm FSD-LC, a scanned path will never reenter the candidate list Q.

Proof. Note that a path enters Q only at initialization (Step 0) or in step 2.4. In the latter situation, the path is always newly generated in step 2.3, on top of the existing set of FSD-admissible paths Γ_{FSD}^{is}.

Because the number of acyclic paths in a directed network is finite, Algorithm FSD-LC must terminate after finite steps. This is formally stated in Theorem 3.1.

Theorem 3.1 *Algorithm FSD-LC terminates after a finite number of steps and yields a set of FSD-admissible paths* Γ_{FSD}^{is} *for each node* i.

Proof. The finite termination directly follows from Proposition 3.3. Upon the termination, all acyclic paths between any node $i \neq s$ and s should have been examined because the procedure essentially performs a breadth-first search. Through FSD-CHECK, only acyclic

paths that are not dominated will be kept in Γ_{FSD}^{is} at termination. Therefore, the retained paths form a final Γ_{FSD}^{is}.

It is clear that the complexity of FSD-LC depends on the size of Γ_{FSD}^{is}. In theory, $|\Gamma_{\text{FSD}}^{is}|$ is bounded by $|K^{is}|$ only, which grows exponentially with the number of nodes n (roughly n^{n-1} in the worse case. Thus, no algorithm of polynomial complexity exists for the RASP problem. We note that the RASP problem belongs to a class of multi-criteria shortest path problems, which are known to be intractable (Hansen 1979; Henig 1985; Miller-Hooks and Mahmassani 2000).

Proposition 3.4 *Algorithm FSD-LC runs in a non-polynomial time* $O(mn^{2n-1}L + mn^n L^2)$.

Proof. In the worst case, the algorithm may have to examine all possible paths for any O-D pair is. There are roughly $n \times n^{n-1} = n^n$ paths in total. For each path, all links might be scanned. Therefore, Step 2 of the algorithm may be executed mn^n times. In Step 2, $O(n)$ operations are required to check the acyclicity, and $O(L^2)$ operations are required to calculate the convolution integral. In FSD-CHECK, $O(L)$ and $O(n^{n-1}L)$ operations are consumed in Step 1 and 2, respectively. Thus, the complexity is on the order of $O(mn^n(n + L^2 + L + n^{n-1}L))$, which yields the above result, ignoring the $n + L$ portion in the parentheses.

In practice, we expect that $|K^{is}|$ is much smaller than n^{n-1}, in particular for sparse networks commonly seen in transportation applications. This has been noticed by a number of authors in numerical experiments (see, e.g., Brumaugh-Smith and Shier 1989 and Miller-Hooks and Mahmassani 2000). Using the results of Henig (1985), we can show that the expected number of FSD-admissible paths is bounded by $\sum_{k=1}^{|K^{is}|} \frac{1}{k} \simeq \log(|K^{is}|)$ when $L = 1$ (i.e., two discrete time budgets). For $L > 1$, however, it is more difficult to establish such a theoretical bound.

3.3 ADAPTIVE DISCRETIZATION APPROACH

A key component in the RASP algorithm is calculation of the path travel time distributions u_k^{rs} by convolving the travel time distributions of its member links. In the last section, we showed a straightforward convolution method (see Equation 3.13). However, our previous research has shown that this method is not computationally efficient (Wu and Nie 2009). We further propose a discretization and convolution approach called the adaptive discretization approach (ADA).

ADA starts by dividing the support of each random variable into L intervals with a uniform length φ, and then computes the probability mass functions using Equation 3.12. The key difference here, comparing with Equation 3.13, is that φ may vary from one random variable to another, depending on the range of the support (discrete random variables often have a well-defined, finite support range. (For continuously distributed random variables that have infinite support ranges, the upper [lower] bound of the support can be taken at a $100(1-\varepsilon)$ (ε) percentile value, where ε is a small positive number

selected by the modeler. The value used in our study is 0.001). Moreover, a procedure termed *consolidation* is introduced to merge consecutive intervals such that no more than one interval has a probability mass smaller than $1/L$. Merging two consecutive discrete intervals means removing the boundary between them and assigning the sum of their probability masses to the new interval. Consolidation produces a set of effective support intervals (ESI) whose size is often much smaller than the initial discretization resolution L.

Consider two random variables X and Y, which, after discretization and consolidation, can be represented by a set of discrete support points, W_x and W_y, and associated probability mass vectors, Q_x and Q_y. Let the number of effective support points for X and Y be D and E, respectively. We have

$$W_x = [w_1^x, ..., w_D^x], W_y = [w_1^y, ..., w_E^y], Q_x = [q_1^x, ..., q_D^x], Q_y = [q_1^y, ..., q_E^y]$$

The following procedure can be used to convolve X and Y, i.e., to obtain $Z = X \otimes Y$.

ADA-Based Direct Convolution Scheme

Inputs W_x, W_y, Q_x, Q_y

Outputs W_z, Q_z, i.e., the vectors of discrete support points and probability mass for $Z = X \otimes Y$.

Step 0 Obtain the range of support for Z. Set $b_0^z = 1.5(w_1^x + w_1^y) - 0.5(w_2^x + w_2^y), b_L^z = 1.5(w_D^x + w_E^y) - 0.5(w_{D-1}^x + w_{E-1}^y)$. Divide $[b_0^z, b_L^z]$ into L intervals of uniform length, and compute $\varphi = (b_L^z - b_0^z)/L$. Initialize $p_l^z = 0, \forall l = 1, \cdots, L$.

Step 1 for $i = 1, 2, \cdots, D$,

for $j = 1, 2, \cdots, E$, -8pt

Calculate $ts = w_i^x + w_j^y$ and $tp = q_i^x q_j^y$. Define $l = [\frac{ts - b_0^z}{\varphi}]_-$, and set $p_l^z = p_l^z + tp$

end for

end for

Step 2 Call Procedure Consolidation to obtain W_z, Q_z with inputs $\{b_0^z, b_1^z, ..., b_L^z\}, \{p_1^z, ..., p_L^z\}$.

Consolidation

Inputs Vectors of initial discrete intervals and probability $\{b_0, b_1, ..., b_L\}, \{p_1, ..., p_L\}$

Outputs $W = \{w_1, ..., w_{L_0}\}, Q = \{q_1, ..., q_{L_0}\}$, where L_0 is the number of effective support points.

Step 0 Initialization. Set $l = 1, i = 1, \varphi = 1/L$. Set $tp = 0, ts = 0$.

Step 1 if $i > L$; stop, set $L_0 = l$; otherwise, $tp = p_i, ts = 0.5(b_{i-1} + b_i)p_i$, go to Step 2.

Step 2 If $tp > \varphi$, $q_l = tp, w_l = ts / tp$ set $i = i+1, l = l+1$, go to Step 1; otherwise, go to Step 3.

Step 3 Set $i = i+1, tp = tp + p_i; ts = ts + 0.5(b_{i-1} + b_i)p_i$, go to Step 2.

The direct convolution scheme has a quadratic complexity. That is, if the number of support points is L, the computation effort required by each convolution is on the order of $O(L^2)$. Our previous research also proposed a group of convolution methods based on the discrete Fourier transform (DFT) (Wu and Nie, 2011). It is well-known that the Fourier transform of the convolution of two integrable functions equals the product of the Fourier transforms of two functions, which is reduced to the pointwise multiplication of two corresponding discrete Fourier transforms (DFT) if the two functions are discrete (Bracewell 2000). Because DFT can be computed in time $O(L \log L)$ using a fast Fourier transform (FFT) algorithm (Cormer et al. 2001), the DFT-based convolution may be much more efficient than the direct convolution methods for large L. However, recent research conducted by us showed that the ADA-based direct convolution outperformed all proposed DFT-based convolution schemes (Wu and Nie 2011) in terms of computational performance. The ADA-based convolution scheme strikes a better balance between accuracy and efficiency. Therefore, in TRIC, all link distributions are discretized by using ADA, and all path travel time distributions are computed by using the ADA-based convolution method.

CHAPTER 4 IMPLEMENTATION

The algorithm described in the last chapter is implemented on top of the Toolkit of Network Modeling (TNM), a C++ class library for solving various transportation network problems. A graphical user interface (GUI) tool named Travel Reliability Inventory for Chicago (TRIC) was developed using MFC and MYSQL. TRIC documents information about travel reliability on Chicago's highway network. We first briefly introduce TNM then describe TRIC.

4.1 CLASS HIERARCHY

This section provides an overview of class hierarchy, with a focus on the classes designed for the stochastic routing problem. A detailed class implementation can be found in Appendix A.

Originally, TNM defines four major network classes:

- `TNM_SNET`: for static applications such as traffic assignment.
- `TNM_DNET`: for macroscopic dynamic applications such as dynamic network loading and dynamic traffic assignment.
- `TNM_MNET`: for microscopic dynamic applications, such as studying vehicles' lane-changing behavior.
- `TNM_ProbeNet`: reserved for stochastic applications.

All four network classes are derived from `TNM_SNET`, whose major data members include, among others, lists of nodes and links, which together represent the topology of the network. Each type of network is usually associated with link and node objects of corresponding types. For example, Class `TNM_SLINK` is the basic link type for `TNM_SNET`, while Class `TNM_DLINK` is used by `TNM_DNET`. For a detailed description of these classes, see Nie (2006).

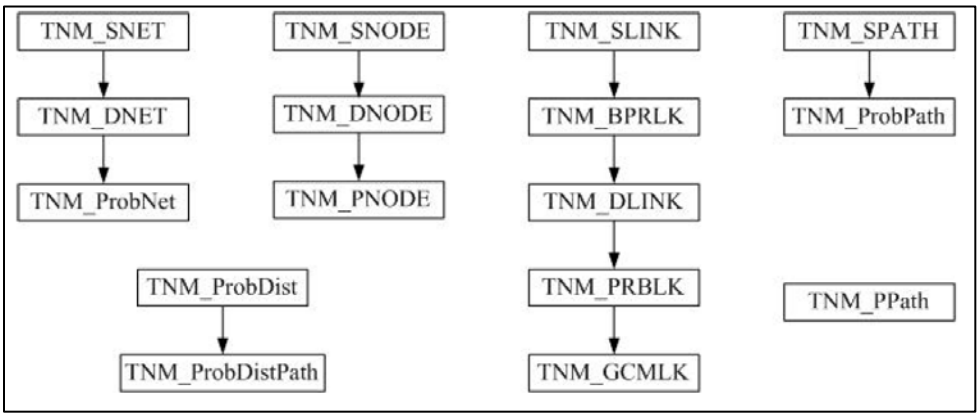


Figure 4.1 A class hierarchy tree.

As shown in Figure 4.1, `TNM_ProbeNet` is derived from `TNM_DNET`. While this class exists in the original TNM, it has been substantially expanded in this research to support reliable routing. The important new functions include generating non-dominant paths and I-O functions for travel time distributions, as well as building functions for a new network format.

In addition, new node and link classes are introduced. TNM_PNODE, derived from TNM_DNODE, is used to construct, store, manipulate, and compare paths. The most important operations defined in Algorithm FSD-LC are actually implemented in this class. TNM_PRBLK and TNM_GCMLINK are derived classes of TNM_DLINK. No important methods are defined in these classes. Instead, they are primarily holders of new data members, such as detector information and road names.

TNM_ProbDist is a new class intended to handle discrete probability distribution. In addition to regular functionality, the class provides an efficient implementation of the convolution integral using the hybrid discretization scheme introduced in Chapter 3. It also allows easy comparison of one distribution with another using stochastic dominance. TNM_ProbDist is further wrapped by class TNM_PPATH, which implements the connection among paths stored at adjacent nodes. Through this connection, it is possible to construct a path from its pointer at any given node. Finally, to be consistent with the existing hierarchy, a new class TNM_ProbPATH is derived from the existing path class TNM_SPATH and wraps TNM_PPATH.

4.2 INTRODUCTION TO TRIC

Software called Travel Reliability Inventory for Chicago (TRIC) was developed based on TNM. In this section, we provide an overview of TRIC, covering various menus, tool bars, and functions.

4.2.1 Get Started

To start TRIC, click "File → Start" on the menu bar at the upper left corner of the window to load Chicago Highway Network, as shown in Figure 4.2. This network is adapted from the planning network developed by the Chicago Metropolitan Agency. It contains roadways with a speed limit of 45 mph or higher and roadways that are necessary to maintain a reasonable degree of connectivity.

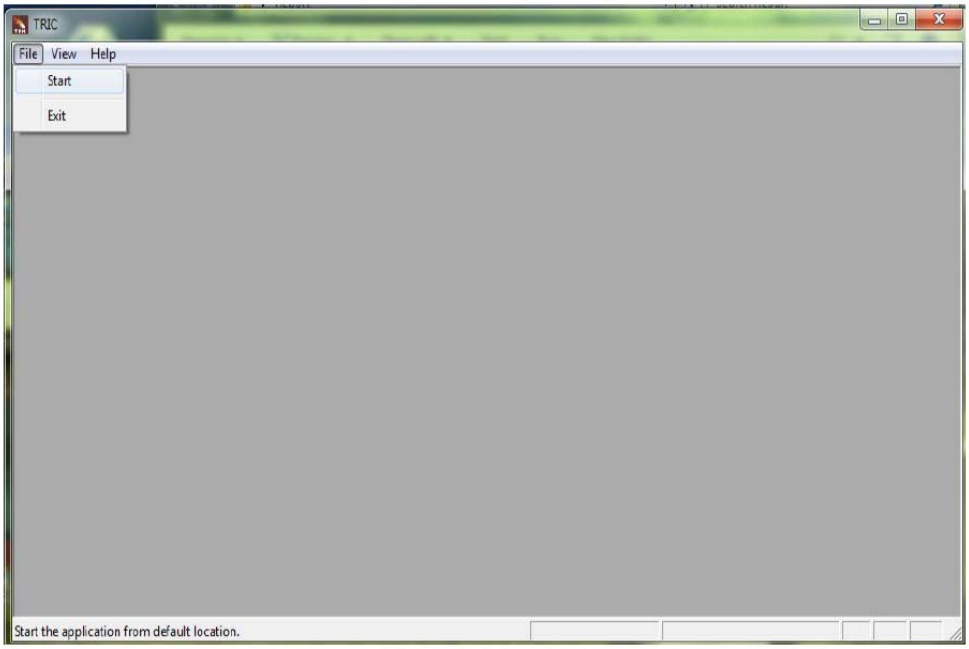


Figure 4.2 Start TRIC.

After the map is initialized, the TRIC screen appears, as shown in Figure 4.3.

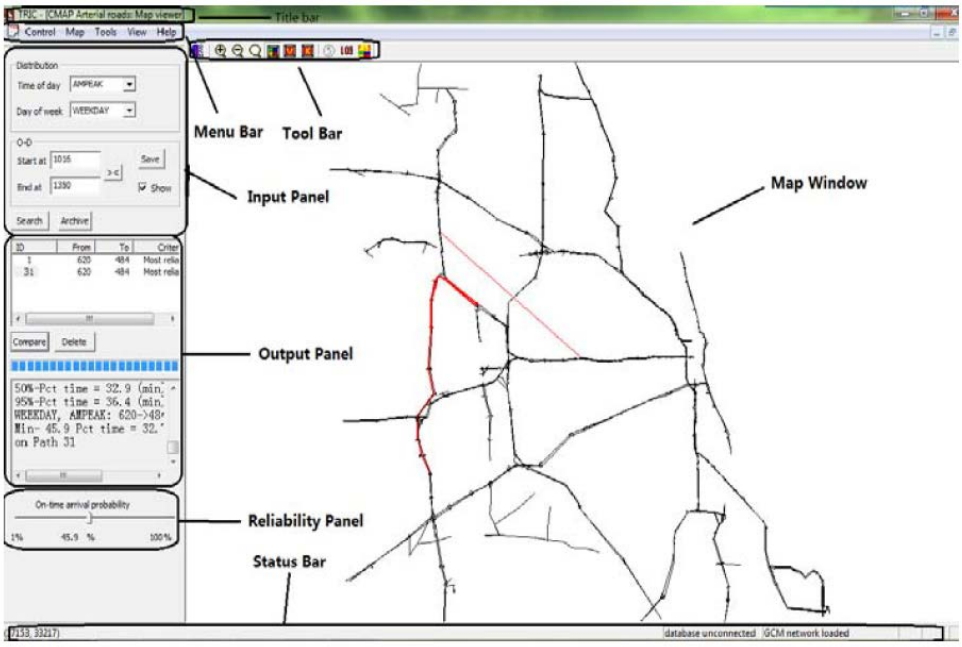


Figure 4.3 TRIC screen.

- **Title bar:** Displays the name of the software and map information.
- **Menu bar:** Offers access to TRIC's commands.
- **Tool bar:** Provides shortcuts to frequent commands.
- **Map window:** Displays the map of Chicago metropolitan area.
- **Input panel:** Sets up parameters for path travel time distribution.
- **Output panel:** Provides information about the reliable path.
- **Reliability panel:** Adjusts on-time arrival probability.
- **Status bar:** Provides information about the loaded map.

4.2.2 Menu Bar

The menu bar provides different commands for TRIC. This section introduces the commands under the menu bar.

1. **Control:** Contains basic commands for TRIC (Figure 4.4).

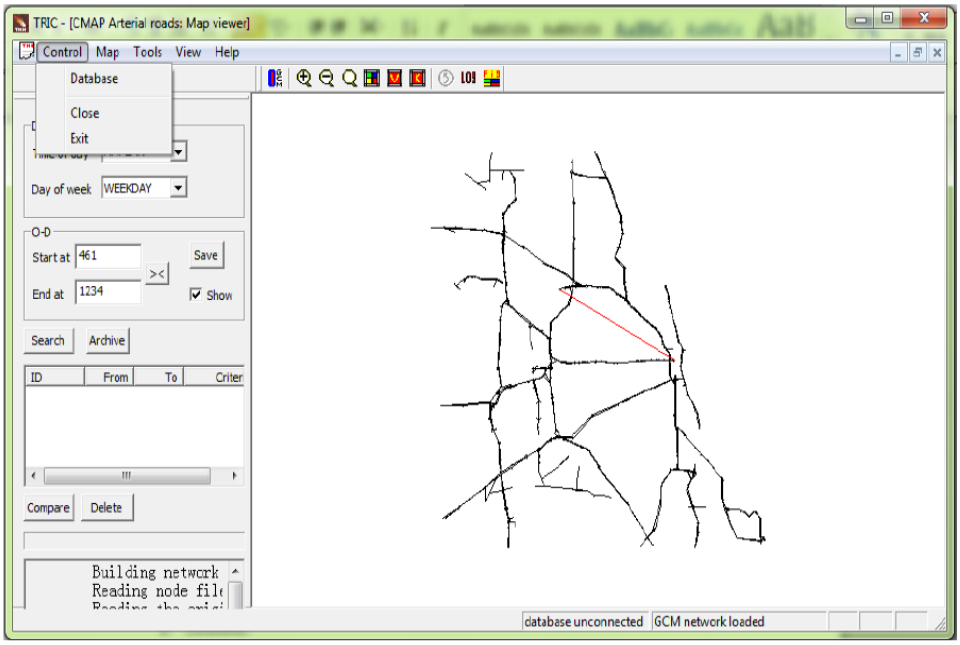


Figure 4.4 Control menu on the menu bar.

- **Database:** Allows users to connect to the database server located at NU-TREND. Click "Control → Database" on the menu bar. A pop-up window appears, as shown in Figure 4.5. In the pop-up window, select the server "NUTranslab" and click "Connect" (accept the default user name and password). Once the server is connected, the status bar will show "gcm at 129.105.69.117". To return to TRIC screen, click "Ok".

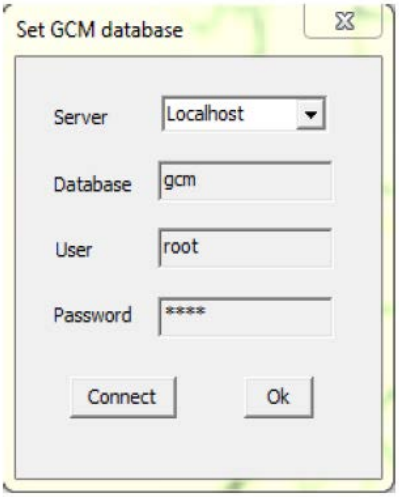


Figure 4.5 TRIC database.

- **Close:** Closes the current map and returns to TRIC before initialization.
- **Exit:** Exits TRIC.

2. **Map:** Provides map navigation and setting commands, as shown in Figure 4.6.

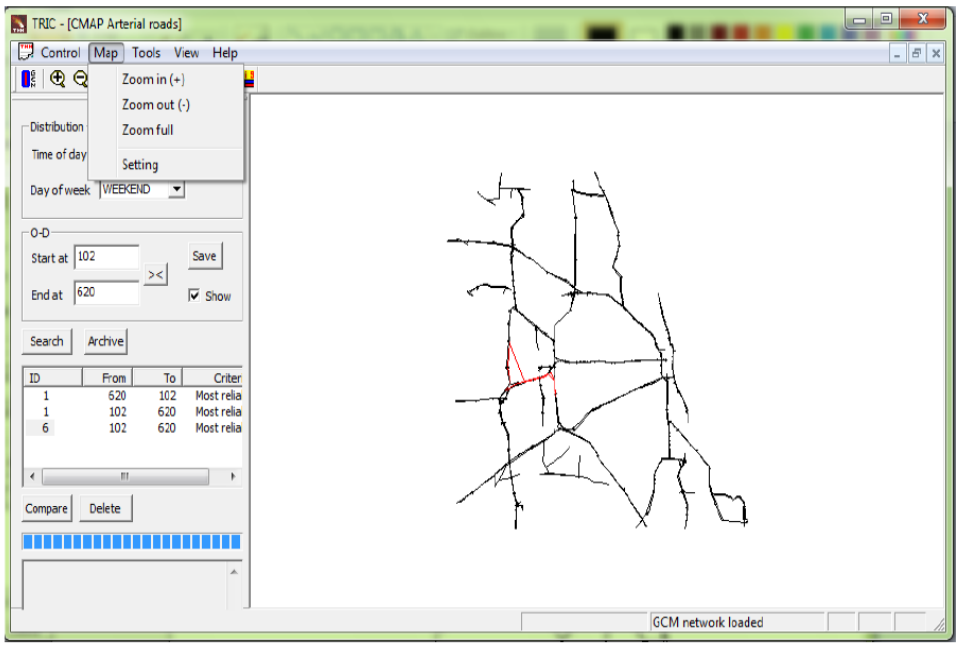


Figure 4.6 Map menu on the menu bar.

- **Zoom in/out:** Zooms the current view in/out.
- **Zoom full:** Zooms to the full size of the map.

  Setting: Changes display options. *It is also accessible by double-clicking on the map window.* Click on “Settings” to bring up the “Set display properties” window, as shown in Figure 4.7.

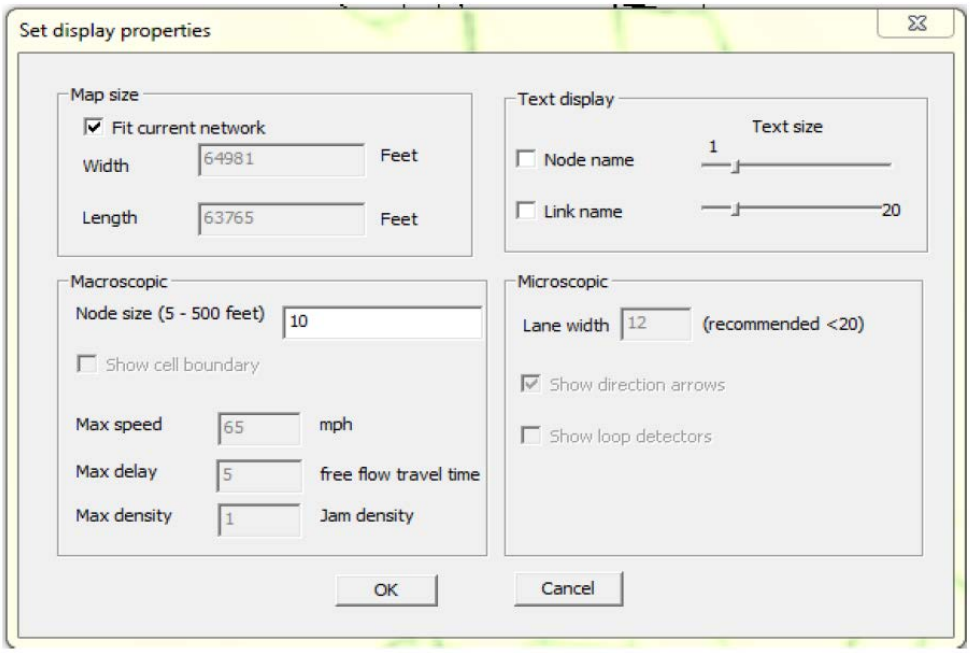


Figure 4.7 Set display properties.

- **Map size:** The default setting is “Fit current network”. It can be changed by adjusting the width and length.
- **Text display:** Enables/disables the display of node/link name, and the text size of the display.
- **Macroscopic:** Provides options to change the node size. Max speed, Max delay, and Max density cannot be changed yet. They are set as their default value.
- **Microscopic:** Provides options to change lane width, as well as display of direction arrows and loop detectors.

3. **Tools:** Provides general TRIC commands (Figure 4.8).

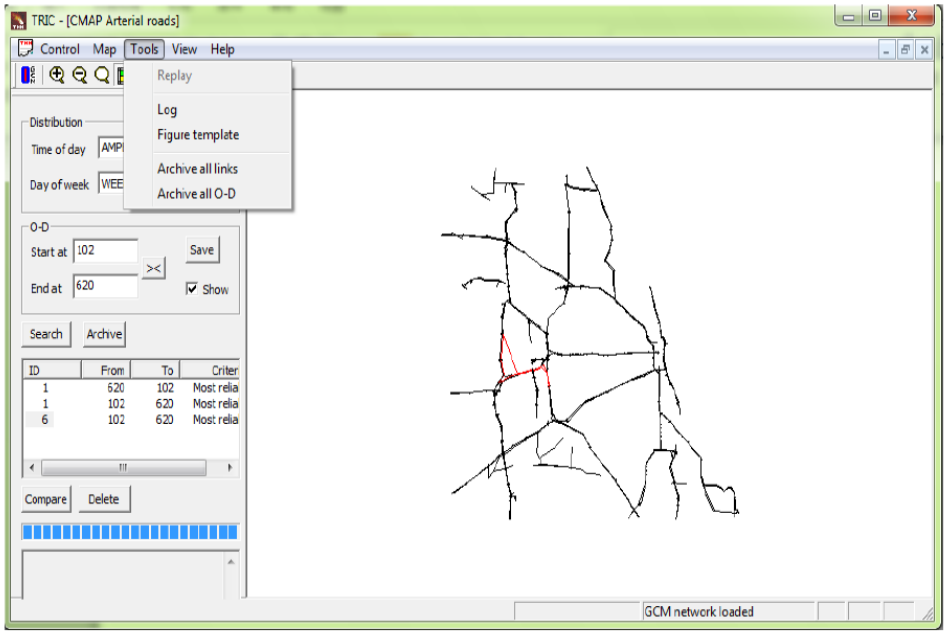


Figure 4.8 Tools menu on the menu bar.

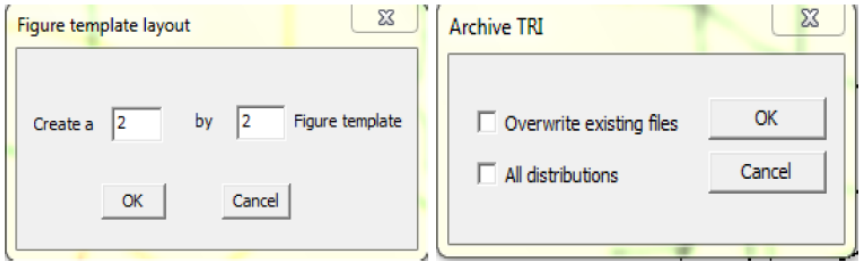
- **Replay:** Provides a visualized as well as dynamic view of traffic speed. To enable replay, switch to speed map first (View → Maps → Speed). Specify the time and data type, and then click “Play” (Figure 4.9). Traffic speed is denoted by different colors as a fraction of free-flow travel time. A change in color on the map indicates a change in traffic speed. Control the replay by dragging the slide bar or using the “F step” (forward) and “B step” (backward) buttons.

Figure 4.9 Replay function to show visualized traffic speed.

- **Log:** Opens a log window that contains detailed trip information from origin node to destination node, as well as warnings or error massages that are otherwise invisible to users in the log panel.

- **Figure template:** Opens a PGL graph editor. All plots produced by CTR are in PGL format and can be saved as a PGL file. The saved PGL files can be loaded and edited using this function and exported to other formats such as jpeg, eps, etc. To create a figure template, click on “Figure template” to open the “Figure template layout” window. Figure 4.10(a) shows a layout of 2 × 2 figures.

  Archive all links: Saves all links in the network to the folder “NU-Trend → TRICSetup → TRI”. It can write all link statistics into the corresponding shape files (up to 30 shape files, each corresponding to one distribution) in a single run. However, the user can choose to write one shape file (corresponding only to the current distribution setting) at a time. The user can also decide whether the existing shape files should be overwritten or the existing shapes in the current file should be kept. As shown in Figure 4.10(b), when “Archive all links” is selected, two options are provided: “Overwrite existing files” and “All distributions”. The first option creates new files and overwrites all previously archived links. (Overwrite will greatly accelerate the process because it waives the need for checking the existence of particular shapes.) The second option archives 30 distributions for all link statistics obtained from 30 combinations of “Time of day” and “Day of week”.



(a) Figure template layout (b) Archive all links/OD

Figure 4.10 Figure template layout and the panel for archiving all links/O-D.

- **Archive all O-D:** Provides a one-step archive function. It can write all reliable path statistics for all O-D pairs stored in the O-D pair list into the corresponding shape files (up to 30 shape files, each corresponding to one distribution) in a single run. However, the user can choose to write one shape file (corresponding only to the current distribution setting) at a time. The user can also decide whether the existing shape files should be overwritten or the existing shapes in the current file should be kept. (Overwrite will greatly accelerate the process because it waives the need for checking the existence of particular shapes.)

4. **View:** Provides commands to display different figures/tables and to display/hide the status bar and tool bar, as shown in Figure 4.11.

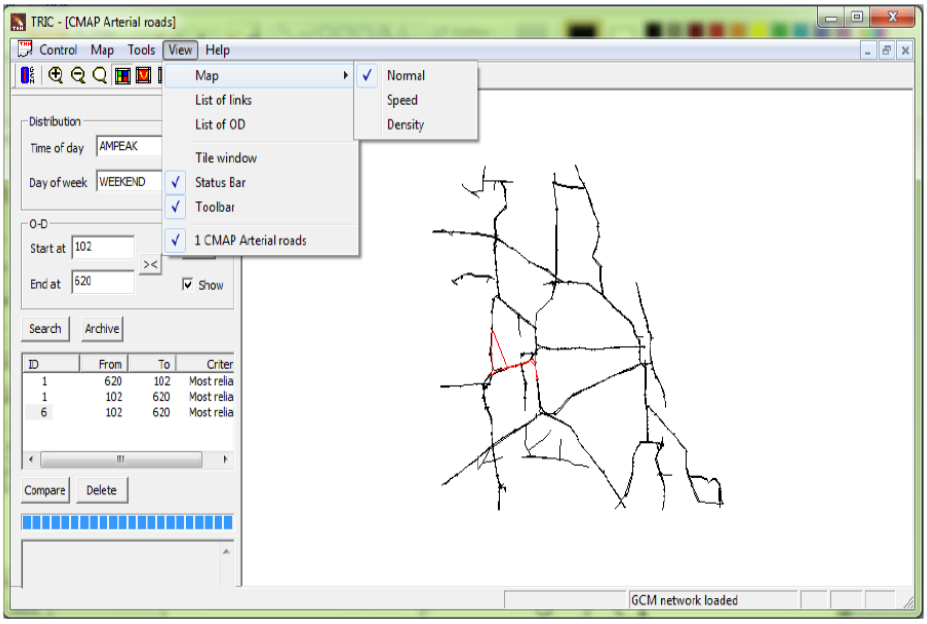


Figure 4.11 View menu on the menu bar.

- **Map:** Provides three different choices:

  Normal: The normal GCM map.

  Speed: Provides speed information on the map. The colored bar on the right-hand side shows the speed as a fraction of free-flow speed, as illustrated in Figure 4.12.

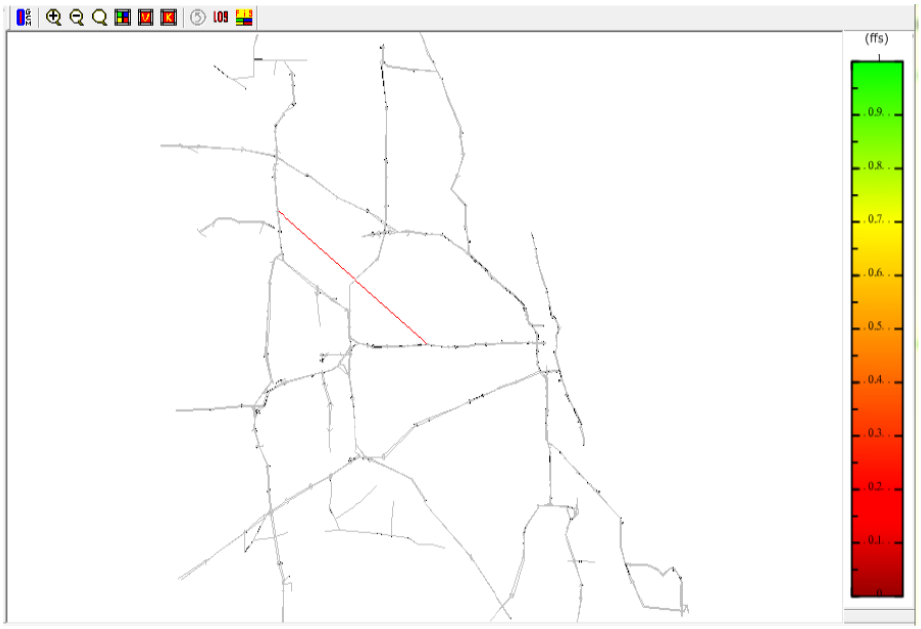


Figure 4.12 Speed map with the colored legend.

- **Density:** Provides a colored scheme for data coverage. As shown in Figure 4.13, the blue line indicates that data are collected from loop detectors. The amber line indicates that data are collected from traffic reports. The green line indicates that data are collected from both sources.

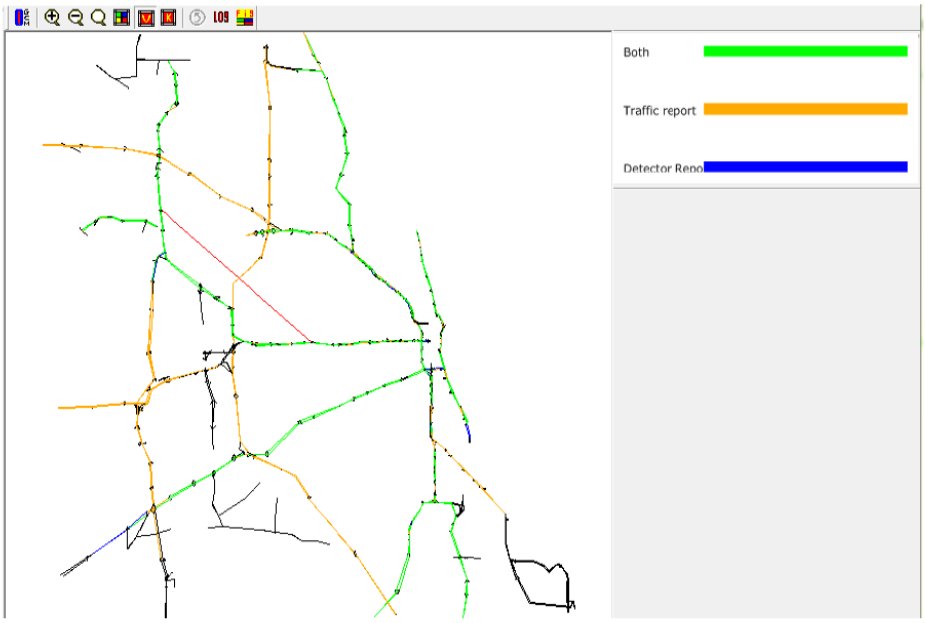


Figure 4.13 Density map.

- **List of links:** Displays the link table, as shown in Figure 4.14. The number of links in the network is 1,970. The link table contains link ID, Type, Start node, End node, Length, Capacity, Free flow speed, RoadName, MODES, Stamp ID, and detection type. The link list and map view are linked inherently. That is, when you click a link on the link view, the map view will automatically zoom to the link and highlight it. This feature makes it easier to locate a link on a map according to its ID (and other properties) or vice versa. The windows can be tiled (vertically or horizontally) to show the dynamic linkage between the link view and map view.

ID	Type	Start node	End no...	Length (ft)	Capacity	Free flow speed	RoadName	MODES	Stamp ID	Detection
1	GCMLK	1	1392	0.280000	1000.000000	30.000000	I-294	ASHThmlb	18	NONI
2	GCMLK	1	1103	0.260000	1000.000000	30.000000		ASHThmlb	20	NONI
3	GCMLK	2	474	0.120000	1000.000000	30.000000		ASHThmlb	191	NONI
4	GCMLK	3	1056	0.890000	12000.000000	55.000000	I-294	ASHThmlb	204	IPASS
5	GCMLK	4	263	0.250000	4000.000000	55.000000	unknown	ASHThmlb	205	IPASS
6	GCMLK	5	238	0.080000	4000.000000	55.000000	LAKE COOK	ASHThmlb	206	NONI
7	GCMLK	6	1340	0.570000	2000.000000	25.000000	unknown	ASHThmlb	207	NONI
8	GCMLK	7	934	0.150000	1000.000000	30.000000		ASHThmlb	386	NONI
9	GCMLK	8	988	0.450000	6000.000000	50.000000		ASHThmlb	391	IPASS
10	GCMLK	11	448	0.150000	1000.000000	30.000000	IL-83	ASHThmlb	629	NONI
11	GCMLK	11	974	0.360000	4800.000000	55.000000	IL-83	ASHThmlb	630	NONI
12	GCMLK	11	275	0.150000	1000.000000	30.000000		ASHThmlb	632	NONI
13	GCMLK	12	122	0.160000	1000.000000	30.000000	EISENHOWER	ASHThmlb	663	NONI
14	GCMLK	13	1360	0.150000	1000.000000	30.000000	OLD ORCHARD	ASHThmlb	703	NONI
15	GCMLK	14	795	0.250000	1000.000000	30.000000	41A	ASHThmlb	709	NONI
16	GCMLK	15	870	0.150000	6000.000000	55.000000	EAST-WEST	ASHThmlb	753	NONI
17	GCMLK	16	1200	0.750000	5400.000000	55.000000		ASHThmlb	757	NONI
18	GCMLK	16	742	0.160000	1000.000000	30.000000	YORK	ASHThmlb	758	NONI
19	GCMLK	17	1232	1.250000	6000.000000	55.000000		ASHThmlb	800	DET+IPA
20	GCMLK	19	803	0.440000	1000.000000	30.000000	I-55	ASHThmlb	815	NONI
21	GCMLK	19	1079	1.100000	6000.000000	55.000000	I-55	ASHThmlb	816	DET+IPA
22	GCMLK	20	635	0.390000	6000.000000	55.000000	I-190	ASHThmlb	840	IPASS
23	GCMLK	20	773	0.200000	2000.000000	30.000000		ASHThmlb	841	NONI
24	GCMLK	21	1234	0.540000	6000.000000	55.000000	I-190	ASHThmlb	842	IPASS
25	GCMLK	23	105	0.200000	2000.000000	25.000000		ASHThmlb	905	NONI
26	GCMLK	24	231	0.280000	1000.000000	30.000000	271B	ASHThmlb	1019	NONI
27	GCMLK	24	1358	2.020000	6000.000000	55.000000	I-55	ASHThmlb	1020	DET+IPA
28	GCMLK	24	1096	0.430000	1000.000000	30.000000		ASHThmlb	1021	NONI
29	GCMLK	26	753	0.040000	4000.000000	30.000000		ASHThmlb	1155	NONI
30	GCMLK	27	26	0.060000	2000.000000	30.000000		ASHThmlb	1157	NONI
31	GCMLK	28	716	0.060000	2000.000000	30.000000	I-355	ASHThmlb	1160	NONI
32	GCMLK	29	1053	1.460000	6000.000000	55.000000	EAST-WEST	ASHThmlb	1166	IPASS
33	GCMLK	30	31	0.300000	2000.000000	30.000000		ASHThmlb	1177	NONI
34	GCMLK	31	92	0.050000	4000.000000	27.500000		ASHThmlb	1179	NONI
35	GCMLK	32	986	0.310000	2000.000000	30.000000	ARLINGTON HEIG...	ASHThmlb	1186	NONI
36	GCMLK	33	810	0.540000	1000.000000	45.000000	unknown	ASHThmlb	1207	NONI
37	GCMLK	33	390	0.470000	4000.000000	55.000000	ELGIN OHARE	ASHThmlb	1208	IPASS
38	GCMLK	34	668	0.420000	1600.000000	30.000000	I-55	ASHThmlb	1237	NONI
39	GCMLK	34	188	3.860000	6000.000000	60.000000	I-55	ASHThmlb	1238	DETECT

Figure 4.14 Link table.

- **List of OD:** Displays a window that stores all archived O-D pairs: the total number of archived O-D pairs and the origin node and destination node of each pair, as shown in Figure 4.15. It is not recommended to directly modify this file. To change O-D pairs, use the “Save: button in the reliable path panel.

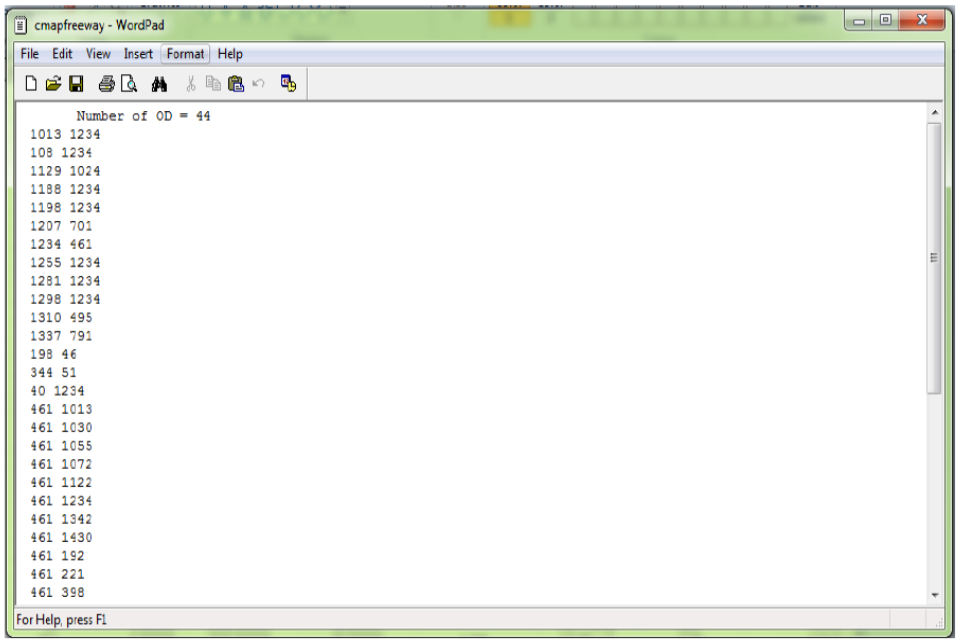


Figure 4.15 Table of O-D pairs.

- **Tile window:** Tiles windows vertically when map view and link list windows are opened simultaneously.
- **Status bar/toolbar:** Displays/hides status bar/toolbar.

5. **Help:** Offers help for TRIC use.

4.2.3 Tool Bar

The tool bar provides shortcuts to commands described in the menu bar section, as shown in Figure 4.16.

Figure 4.16 Tool bar.

 Opens the database window.

 Adjusts the size of the map.

 Displays different kinds of maps.

 Dynamically displays travel time data on the entire network.

 Opens the log window.

 Opens the figure template window.

4.2.4 Map Window

The map window displays a map of expressways in the Chicago metropolitan network, as shown in Figure 4.17.

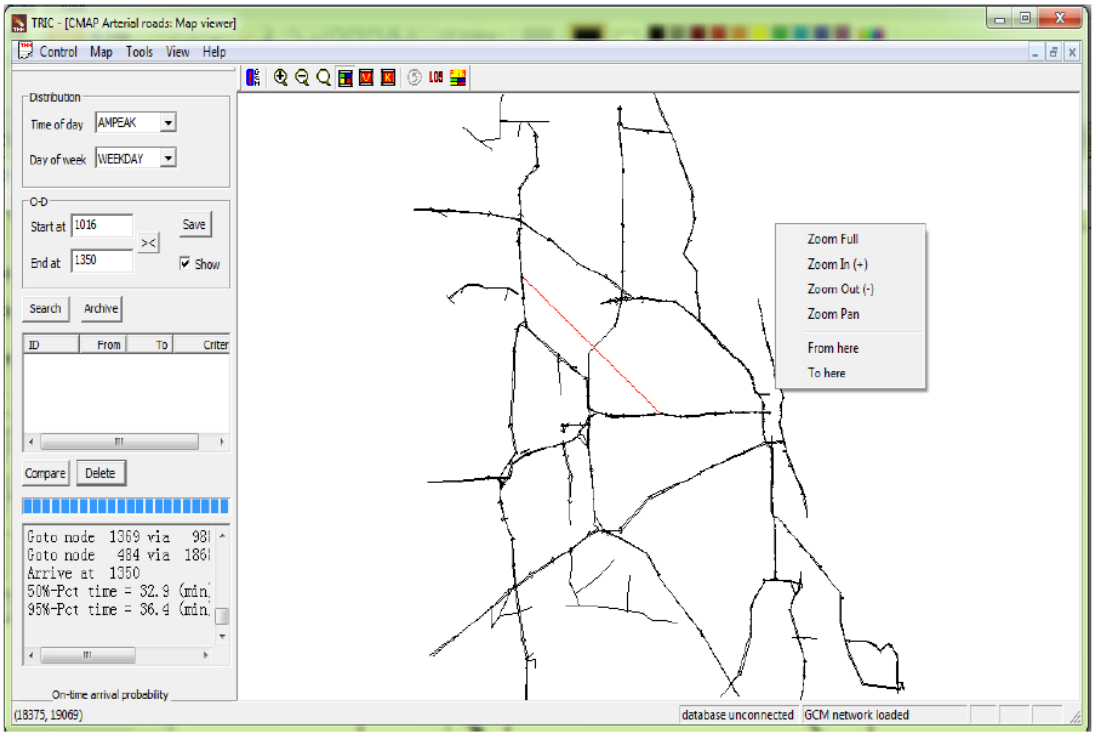


Figure 4.17 Map window.

- Double-click the map to open the “Set display properties” window, as shown in Figure 4.18.

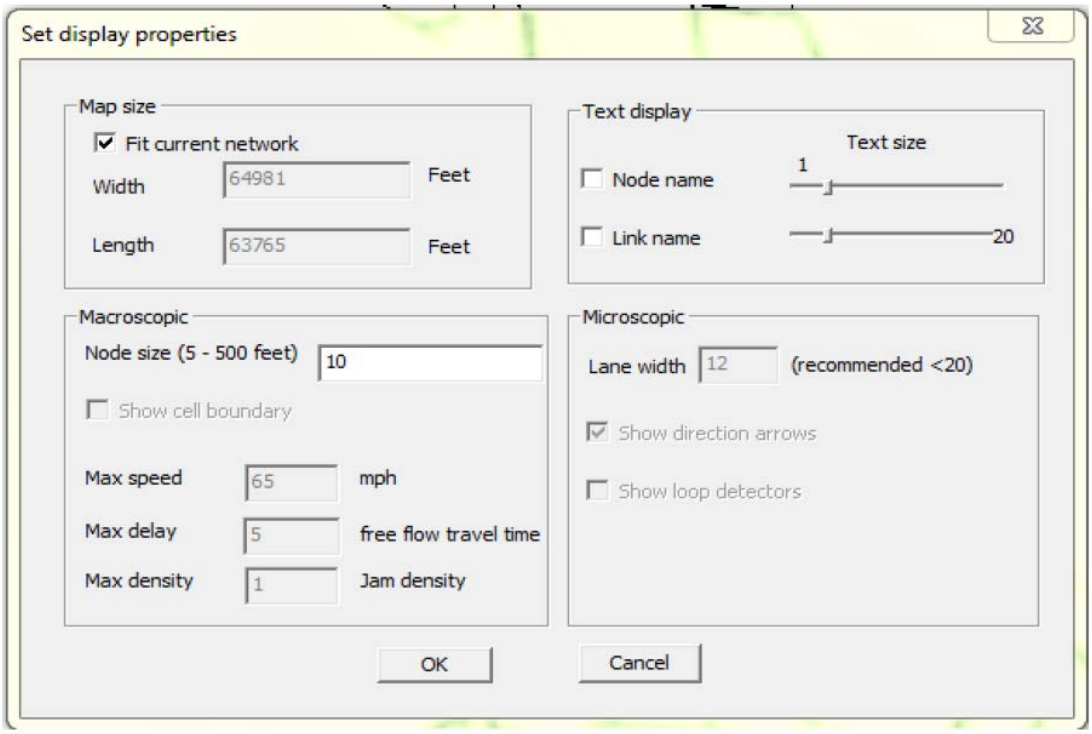


Figure 4.18 Map settings and properties.

- Right-click on the map to open the drop-down list to zoom the map or to select the origin point (from here) and destination point (to here) for a route, as shown in Figure 4.19. Once the origin point and destination point are selected, a red straight line is plotted on the map to connect these two points. The red line does not indicate a path; it is only a convention to denote the origin point and destination point for reference.

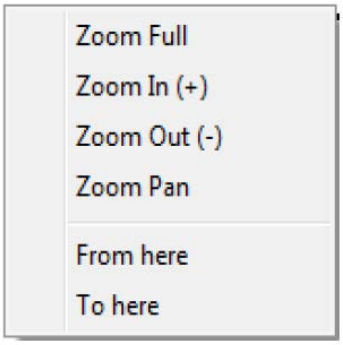


Figure 4.19 Map drop-down list.

- Right-click on a path to open a drop-down list, which consists of two options, as shown in Figure 4.20: archive and inquiry (it is sometimes necessary to zoom in the map to right-click on a path). Click “Archive” to save this path (corresponding to the current specified distribution) or click “Inquiry” to pop up the travel time distribution window. The travel time distribution window provides information about the CDF and PDF of the path travel time, as shown in Figure 4.21. To display the travel time distribution, specify “traffic quantity” first by choosing I-

PASS data or detector data. Then specify parameters related to time (season, day of week, and time of day). You can also change the resolution and lower/upper bounds of the distribution. Once all parameters are set up, click the “Distribution” button to display the travel time distribution (Figure 4.21).

-

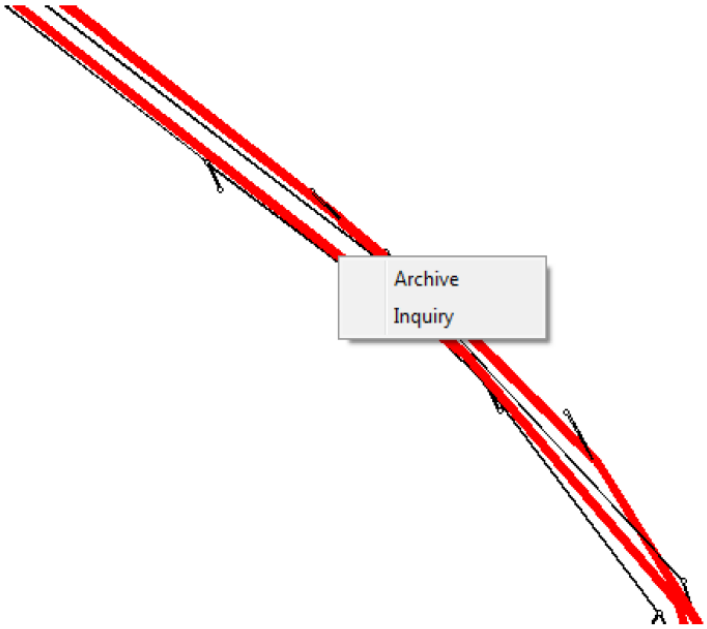


Figure 4.20 Drop-down list for a selected path.

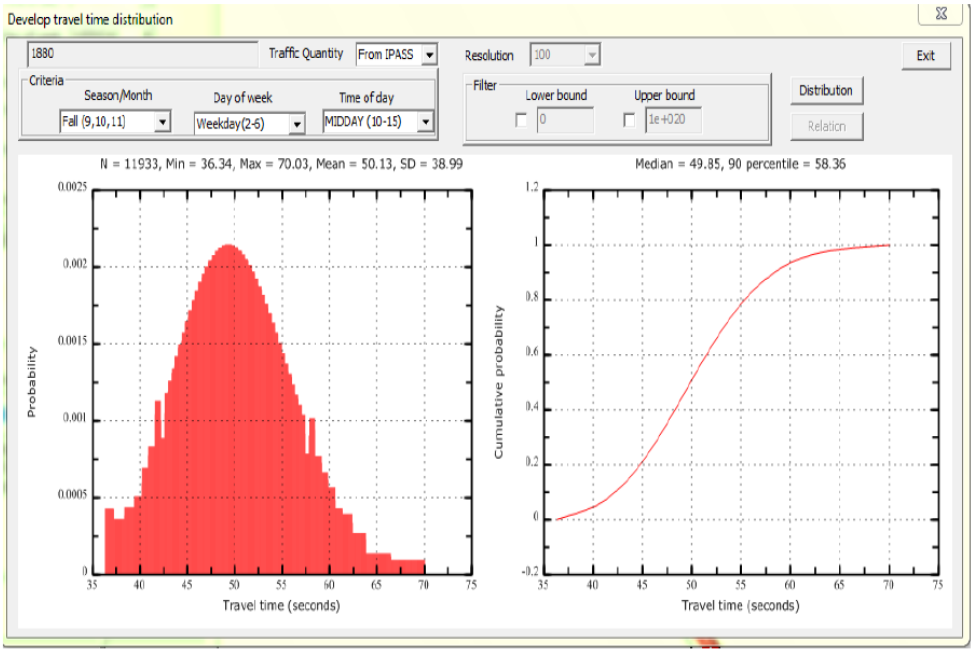
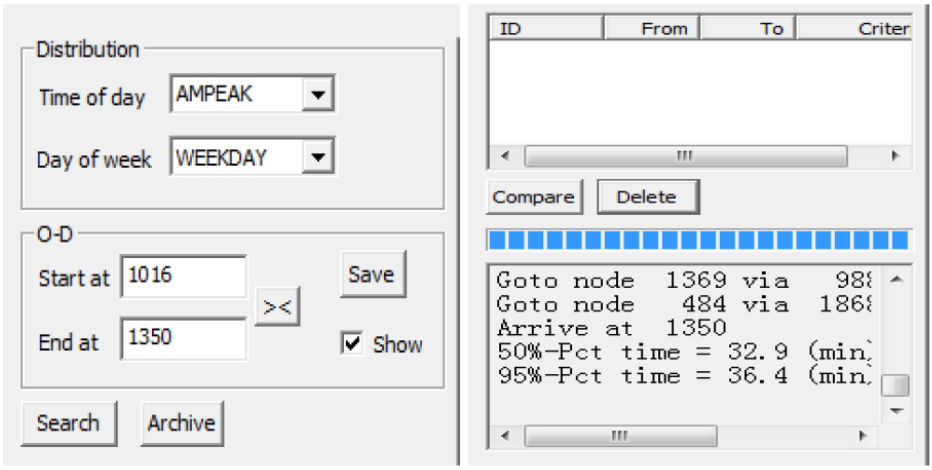


Figure 4.21 Travel time distribution window for a specified path.

4.2.5 Input Panel

The input panel provides options for entering different parameters. Users can choose both distribution of link travel time and O-D pairs, as shown in Figure 4.22(a).



(a) Input panel (b) Output panel

Figure 4.22 Input and output panel.

Distribution panel: Adjusts different link travel time distributions at different times. This panel generates different travel time distributions based on the user's choice of travel time. There are 30 different distributions on each link. When the choice for time of day or day of week is changed, the distribution information will be reloaded.

- "Time of day" provides five options from the drop-down list.
  - AMPEAK: Morning peak hour (6–10 a.m.)
  - PMPEAK: Afternoon peak hour (4–8 p.m.)
  - MIDDAY: Middle of the day (10 a.m.–4 p.m.)
  - OFFPEAK: Non-peak hour (other time)
  - NA: All day
- "Day of week" provides six options from the drop-down list.
  - WEEKDAY
  - WEEKEND
  - 0: Sunday
  - 6: Saturday
  - 5: Friday
  - NA: All week

O-D panel: Allows users to choose different O-D pairs. Users can either enter the O-D pair on this panel or right-click the map to set up O-D pairs (select "From here" and "To here," respectively). Users can click the "show"

box to display/hide the red line that indicates the origin and destination of a route.

>< Switches between origin and destination nodes.

Save Saves O-D pairs to list of O-D pairs. This list will be used when an O-D archive is performed.

Once distribution and O-D pairs are set up, click “Search” or “Archive” to find or save the reliable path between the O-D pair for the specified distribution. All reliable paths for the selected O-D pair will be listed. The search result is reported in the output panel.

4.2.6 Output Panel

The output panel provides information about reliable paths based on user-specified input, as shown above in Figure 4.22(b).

Display the CDF and PDF of a reliable path: Double-click the path. At the top of the PDF plot, minimum, maximum, mean, and standard deviation are calculated (Figure 4.23). At the top of CDF plot, 50th percentile, 95th percentile, B-index (buffering index, the ratio of the 95th percentile travel time to free-flow travel time) and P-index (planning index, the ratio of the 95th percentile travel time to mean travel time) are calculated.

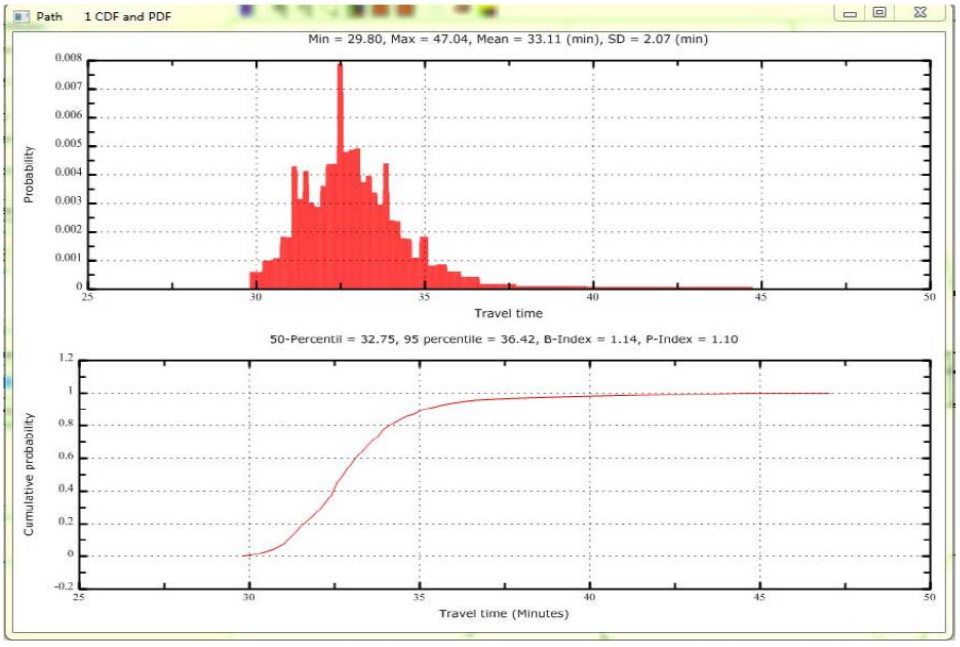


Figure 4.23 Display PDF and CDF of reliable path.

- **Adjust the properties of the plots**: Double-click the plot.

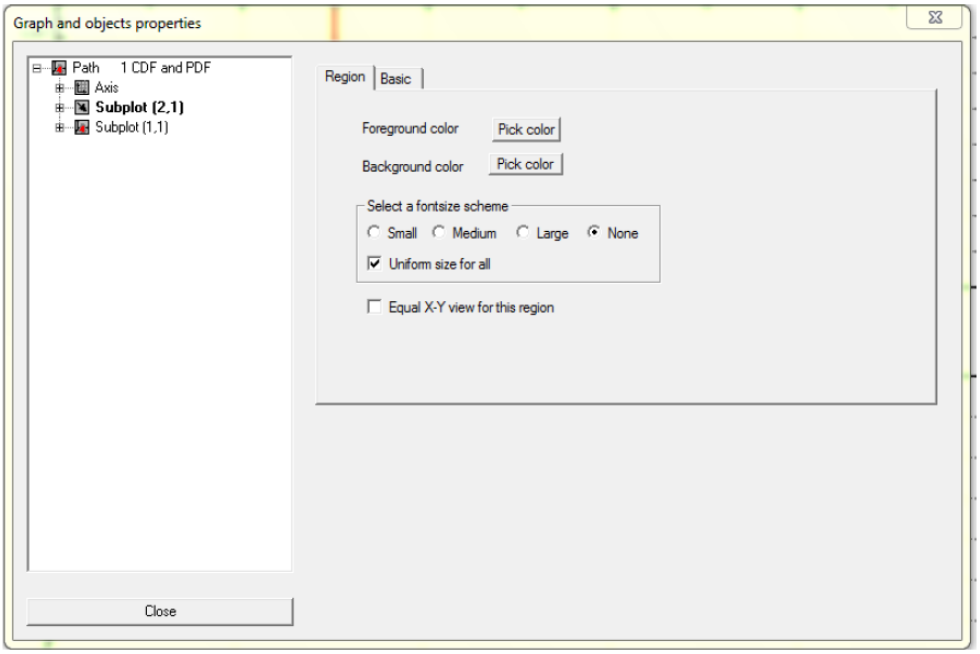


Figure 4.24 Graph properties.

Compare Compares multiple paths. Select multiple paths with Shift or Ctrl, then click "Compare" to display the comparison graph of multiple paths, as shown in Figure 4.25.

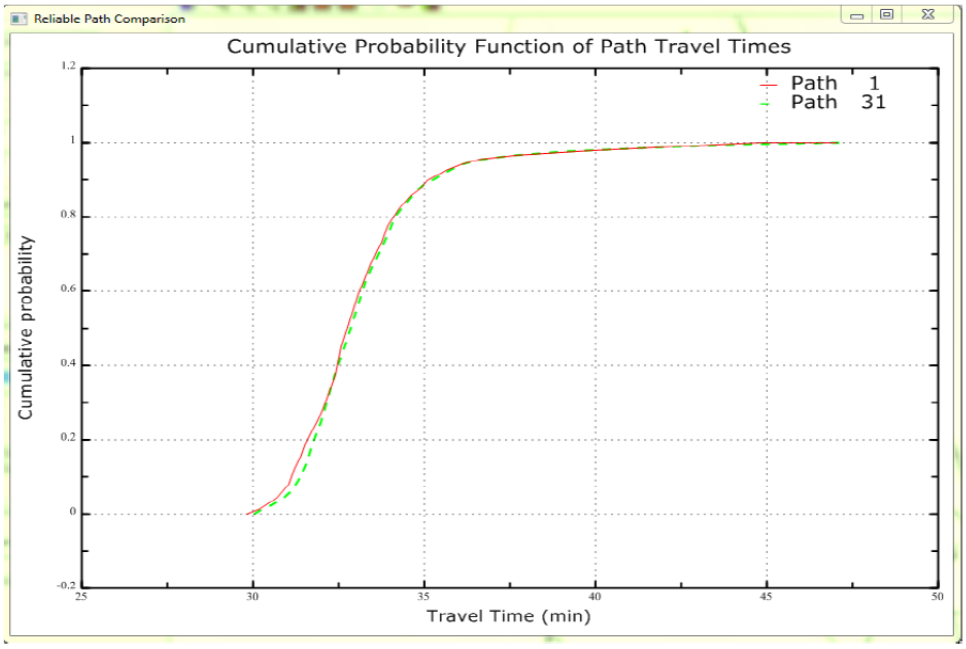


Figure 4.25 Multiple paths comparison.

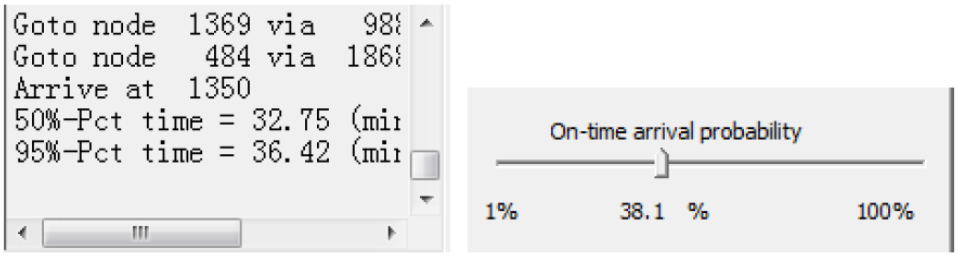
Delete Deletes paths from the list. Select paths from the list and click “Delete” to remove the paths from the list.

4.2.7 Log Panel

The log panel is below the path list, as shown in Figure 4.26(a). It provides a summary of the path. It directs the path choice at each node and summarizes the 50th percentile travel time and 95th percentile travel time.

4.2.8 Reliability Panel

The reliability panel adjusts the user’s preference regarding on-time arrival probability, as shown in Figure 4.26(b). The scroll bar is used to adjust on-time arrival probability. Travel time corresponding to the selected on-time arrival probability is displayed in the log panel after adjustment. When there is more than one path for a given O-D pair (and if they are all selected), the minimum percentile travel time will be displayed, and the path that gives the optimal value will be highlighted.



(a) Log panel (b) Reliability panel

Figure 4.26 Log panel and reliability panel.

CHAPTER 5 CASE STUDIES AND INTEGRATION WITH GIS

5.1 TRIC CASE STUDIES

In this chapter, we provide several case studies to help users better understand the how the TRIC software package can improve their decision-making process. We chose two specific nodes as the O-D pair. Node 908 is the starting node, and Node 132 is the ending node (Figure 5.1). Node 908 is close to the Chicago Transit Authority's Clinton Blue Line Station. Node 132 is close to Montrose Avenue where Kennedy Expressway merges with the southern end of the Edens Expressway. This O-D pair was chosen for two reasons: (1) this O-D pair is a connecting segment between downtown Chicago and O'Hare airport, and (2) the pair traverses two major routes: I-90 and the I-90 Expressway.

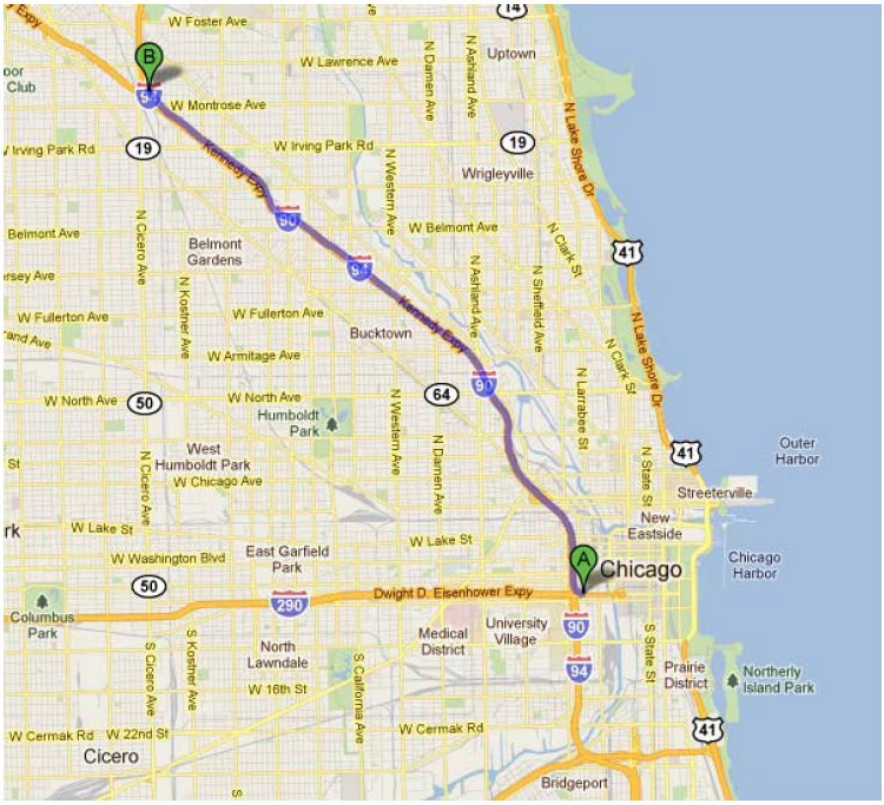


Figure 5.1 Snapshot between Node 908 (A) and Node 132 (B).

Three case studies are provided to illustrate use of the software package, each of which is analyzed between the same O-D pair at different times. These cases are as follows:

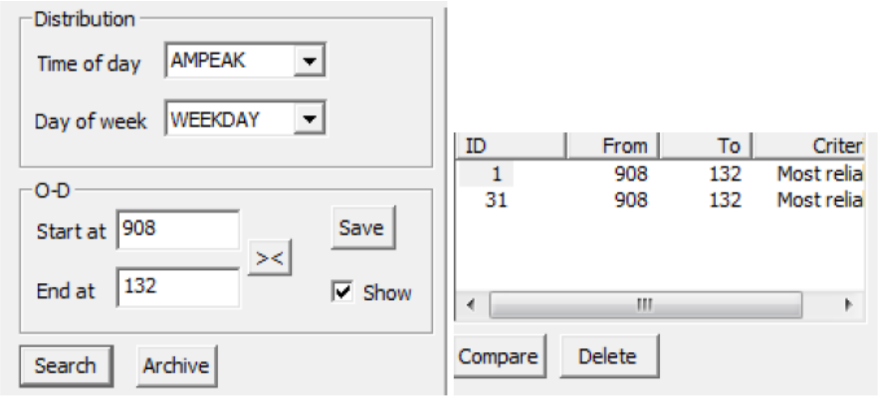
1. Weekday morning peak hours (WEEKDAY AMPEAK, 6–10 a.m.)
2. Weekday middle of the day (WEEKDAY MIDDAY, 10 a.m.–4 p.m.)
3. Weekend morning peak hours (WEEKEND AMPEAK, 6–10 a.m.)

Traffic conditions vary significantly during each time period.

5.1.1 Case 1: WEEKDAY AMPEAK

The first case is for weekday morning traffic during peak hours. We set up the distribution as shown in Figure 5.2. We can choose the O-D pair from the map by right-clicking the nodes on the map or by enter the O-D pair directly into TRIC. In this case, we chose Node 908 as the start node and Node 132 as the end node.

Once all parameters are set up, click “Search” to find the reliable paths [Figure 5.2(a)]. In this case, two reliable paths are found, as shown in Figure 5.2(b).



(a) Input parameters in Case 1 (b) Reliable shortest paths in Case 1

Figure 5.2 Input parameters and the identified reliable shortest paths in Case 1.

Note: It is helpful to show the log panel (click the “Log” button in tool bar). The log panel provides detailed information about the reliable paths, including origin node, path detail, destination node, and percentile travel time, as shown in Figure 5.3.

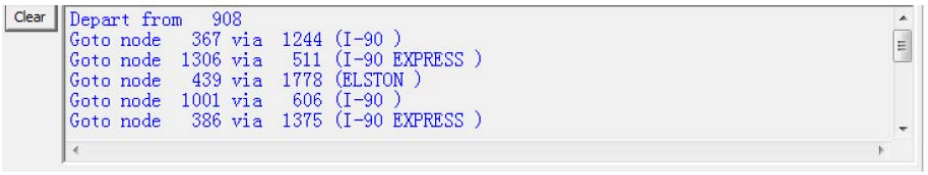


Figure 5.3 Log panel.

Display detailed path information: Click on either path to display the detailed path information in the log panel [Figure 5.2(b)]. In this case, the reliable path starts at Node 908, then goes to Node 367 via Link 1244 (I-90), as shown in Figure 5.3. Each row in the result includes the next node and the link leading to the node. The result provides two ways to track the reliable path: tracking the next node and tracking the next link. We can use either way to find the reliable path between the origin node and destination node.

The end of log (Figure 5.4) shows the percentile travel time of the selected path. In this case, the 50th percentile time = 12.92 (min). This means that if we want to guarantee a 50% probability of arriving on time, 12.92 minutes should be reserved for the trip. In the meantime, if we want a higher on-time arrival probability, say 95%, 18.17 minutes should be reserved for the trip. The result is consistent with common sense—the higher the desired arrival probability, the more time we have to budget for our trips.

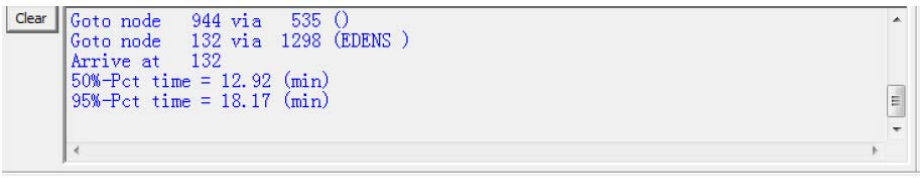


Figure 5.4 Path choice in Case 1.

Display CDF and PDF: Double-click a path to display the cumulative density function (CDF) and probability density function (PDF) of the selected path, as shown in Figure 5.5 for Path 1. The minimum travel time on Path 1 is 8.05 minutes, the maximum travel time is 23.41 minutes, the mean travel time is 13.26 minutes, and the standard deviation is 2.67 minutes. The CDF curve shows that the 50th and 90th percentile times are 12.92 minutes and 18.17 minutes, respectively. The B-index gives the ratio of the 95th percentile time to free-flow travel time. The P-index gives the ratio of the 95th percentile time to mean travel time. In this case, the B-index is 2.11 and the P-index is 1.37.

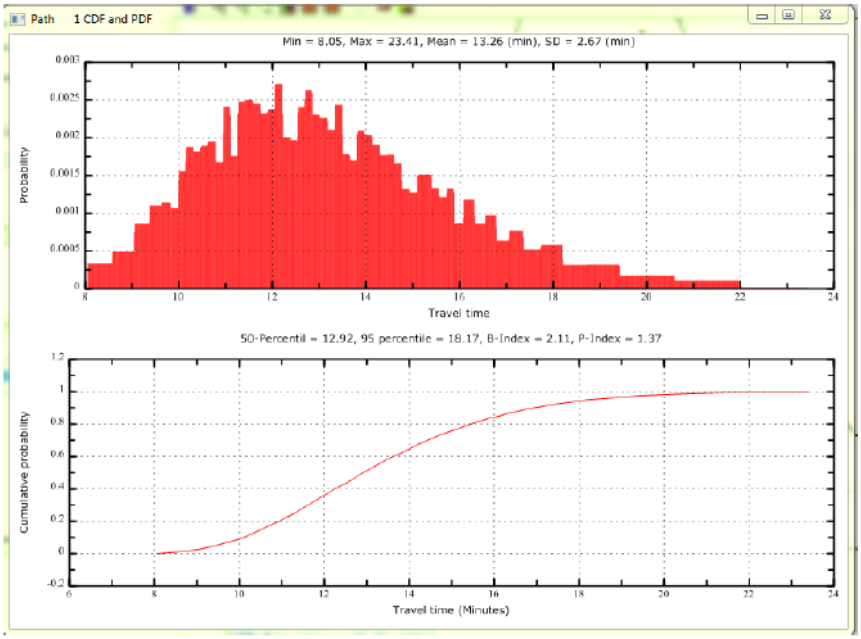


Figure 5.5 PDF and CDF in Case 1.

Compare different paths: We can also compare the CDF of different paths between the O-D pair. Select multiple paths and click the “Compare” button to show the reliable path comparison plot. In this case, two paths are selected, Path 1 (red line) and Path 31 (green line), as shown in Figure 5.6. The two paths intersect around point (12.5, 0.42). Before the intersection, Path 1 lies above Path 31 on the CDF curve. The comparison plot indicates that if on-time arrival probability is less than 0.42, Path 1 is better than Path 31. For example, Path 1 requires less travel time to achieve the same on-time arrival probability or Path 1 generates a higher on-time arrival probability if travel times are identical. On the right side of the intersection, Path 31 is a better choice because the CDF curve of Path 31 lies above Path 1.

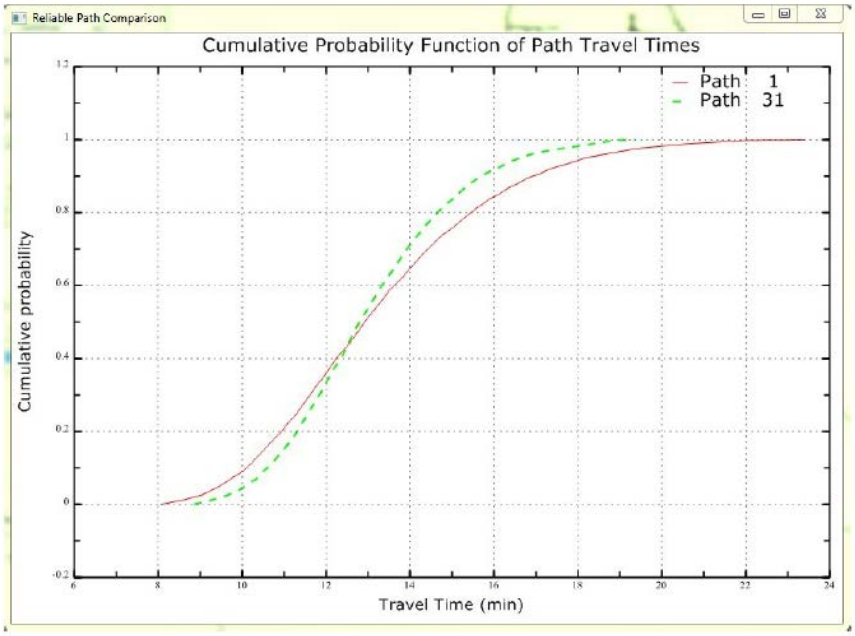


Figure 5.6 Path comparison in Case 1.

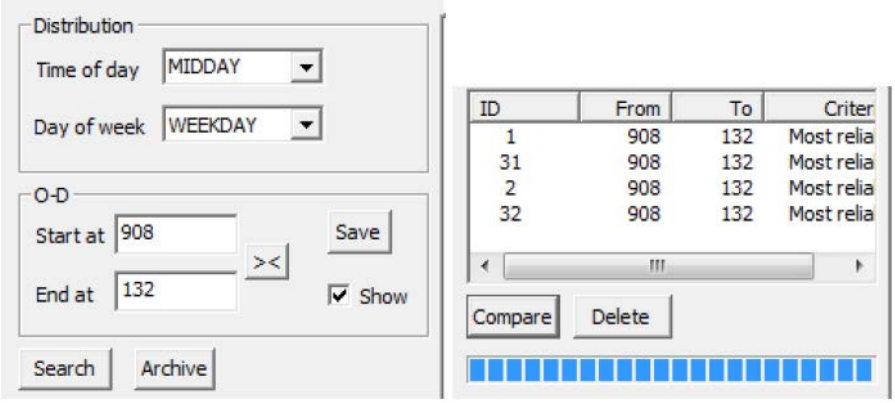
Adjust on-time arrival probability: Use the reliability panel to adjust on-time arrival probability. Once on-time arrival probability is adjusted, travel time is recalculated and shown in the log panel to reflect the adjustment. In this case study, we calculated different on-time arrival probabilities and summarize the results in Table 5.1.

Table 5.1 Travel Time Budgets for Different On-Time Arrival Probabilities

On-time arrival probability	Travel time (min)
38.1%	12.1
50%	12.9
78.1%	15.2
96 %	18.2

5.1.2 Case 2: WEEKDAY MIDDAY

This case is for midday traffic on a weekday. We set up the distribution as shown in Figure 5.7. We choose Node 908 as start node and Node 132 as end node, as in Case 1, so that we can see the impact of time of day on reliable paths.



(a) Input parameters in Case 2 (b) Reliable shortest paths in Case 2

Figure 5.7 Input parameters and the identified reliable shortest paths in Case 2.

Clicking "Search" identifies two reliable paths, Path 2 and Path 32, as shown in Figure 5.7(b). To compare the effect of peak hour, reliable paths for the morning traffic peak from Case 1 (Path 1 and Path 31) are also included.

Display detailed path information: Click on a path (Path 2, for example). Detailed information is displayed in the log panel (Figure 5.8). The panel shows that the 50th percentile travel time is 9.33 minutes and the 95th percentile travel time is 13.09 minutes.

Figure 5.8 Log panel in Case 2.

Display CDF and PDF: Double-click a path to display its CDF and PDF. In this example, we chose Path 2 for display of CDF and PDF, as shown in Figure 5.9. The minimum travel time on Path 2 is 7.23 minutes, the maximum travel time is 17.04 minutes, the mean travel time is 9.69 minutes, and the standard deviation is 1.75 minutes. The CDF curve shows that the 50th and 95th percentile times are 9.33 minutes and 13.09 minutes, respectively. As before, the B-index gives the ratio of the 95th percentile time to free-flow travel time, and the P-index gives the ratio of the 95th percentile time to mean travel time. In this example, the B-index is 1.52, and the P-index is 1.35.

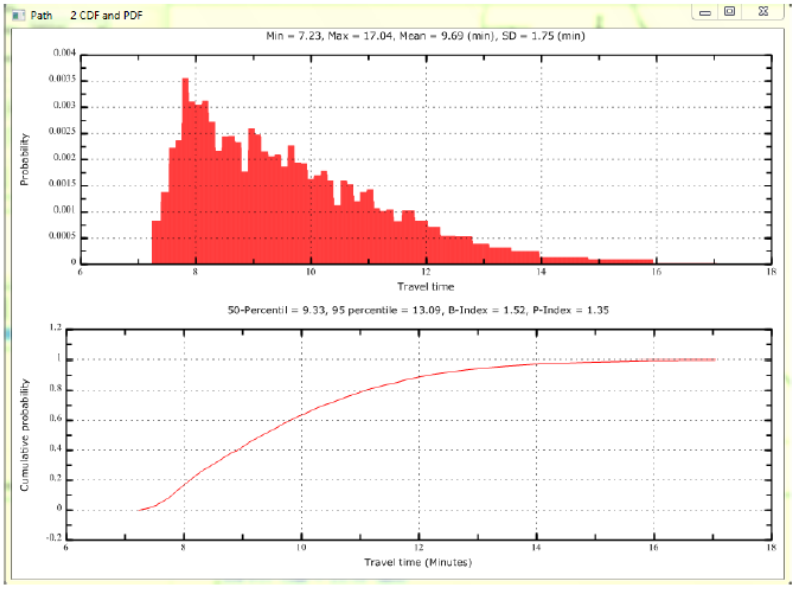


Figure 5.9 CDF and PDF in Case 2.

Compare different paths: As before, we can compare different paths generated for the same time period (Path 2 and Path 32). Moreover, we can compare the same path for different time periods (Path 1 and Path 2). Before comparison, we know that Path 1 was generated during morning rush hour (AMPEAK), while Path 2 was generated during the middle of the day (MIDDAY). Based on the information, we can predict that Path 2 should have higher on-time arrival probability for the same travel time budget and a lower time budget for the same on-time arrival probability. In short, the CDF of Path 2 lies above Path 1. Indeed, this is true in our example. Path 2 lies above Path 1 if we compare the CDF of both paths. In the comparison plot, Path 2 (green line) lies above Path 1 (red line).

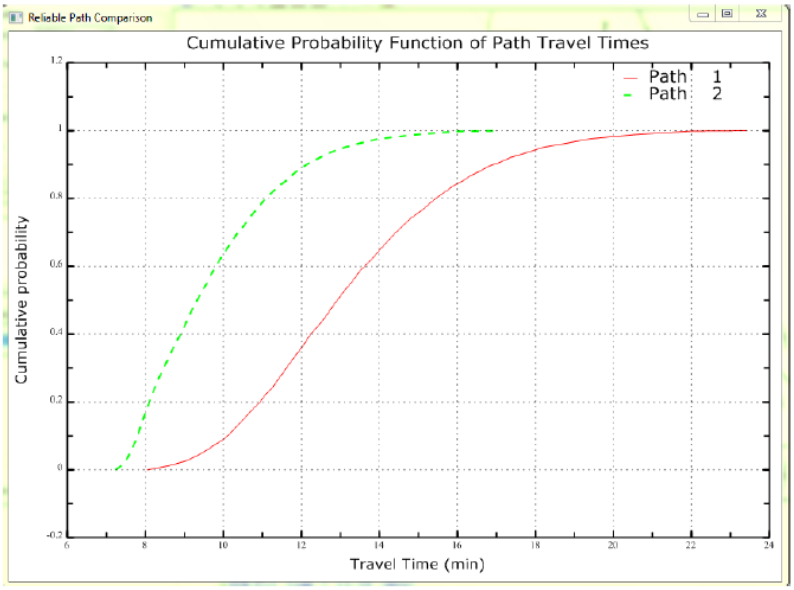


Figure 5.10 Path comparison in Case 2.

Adjust on-time arrival probability: As before, we can adjust on-time arrival probability to reflect the traveler's risk-averse behavior. The higher the on-time arrival probability, the more risk-averse the traveler. Moreover, high on-time arrival probability normally corresponds to more travel time. In this case, we adjust on-time arrival probability on Path 32 to 66.4%, as shown in Figure 5.11. The log panel (Figure 5.12) summarizes the result after the adjustment. The percentile travel time obtained is 10.1 minutes.

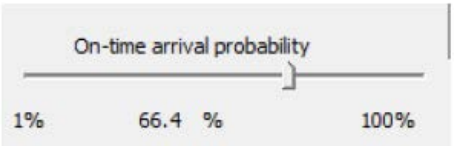


Figure 5.11 On-time arrival probability in Case 2.

Figure 5.12 Output for on-time arrival probability 66.4% in Case 2.

Compare multiple paths: We now have found four reliable paths for two different time periods. These four paths are summarized in Table 5.2.

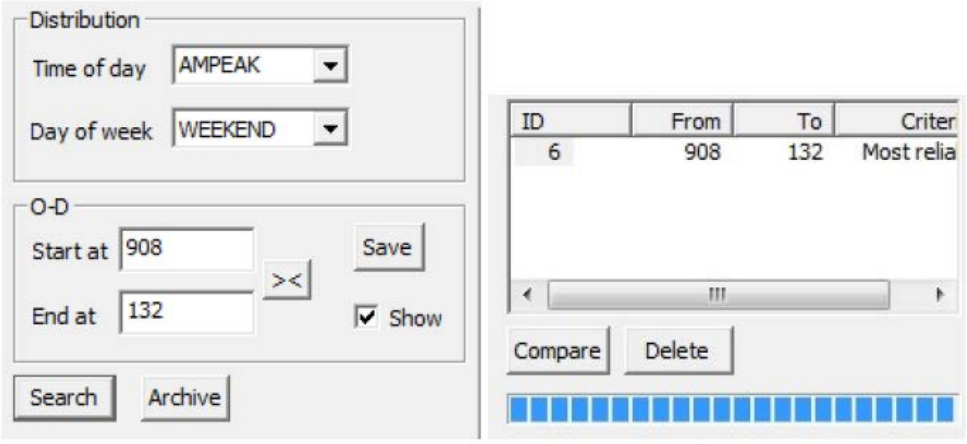
Table 5.2 Comparing Multiple Paths

Path	Path 1(AMPEAK)	Path 31(AMPEAK)	Path 2(MIDDAY)	Path 32(MIDDAY)
Min	8.05	8.85	7.23	7.55
Max	23.41	19.38	17.04	14.55
Mean	13.26	13.02	9.69	9.76
50 % percentile	12.92	12.79	9.33	9.52
95 % percentile	18.17	16.61	13.09	12.37

The performances of Path 2 and Path 32 are better than those of their counterparts for all five measurements. This effect is due to the time period selected for traveling. The effect of morning rush hour (AMPEAK) is obvious in this case. Moreover, the choice between Path 1(2) and Path 31 (32) depends on on-time arrival probability. For low on-time arrival probability, Path 1 and Path 2 are preferred. For high on-time arrival probability, Path 31 and Path 32 are preferred.

5.1.3 Case 3: WEEKEND AMPEAK

In this case, we analyze the travel time during morning peak traffic on the weekend. As before, we use Node 908 as the start node and Node 132 as the end node. The input parameters are shown in Figure 5.13(a).



(a) Input parameters in Case 3 (b) Reliable shortest path in Case 3

Figure 5.13 Input parameters and the identified reliable shortest paths in Case 3.

In this case, only one reliable path is found, Path 6, as shown in Figure 5.13(b). The 50th percentile travel time is 6.63 minutes, and the 95th percentile travel time is 7.26 minutes, as shown in Figure 5.14. Percentile travel time is far lower than in the previous two cases because there is less traffic during weekends. As explained in the following paragraphs, other statistics are also lower than those on weekdays (Figure 5.15).

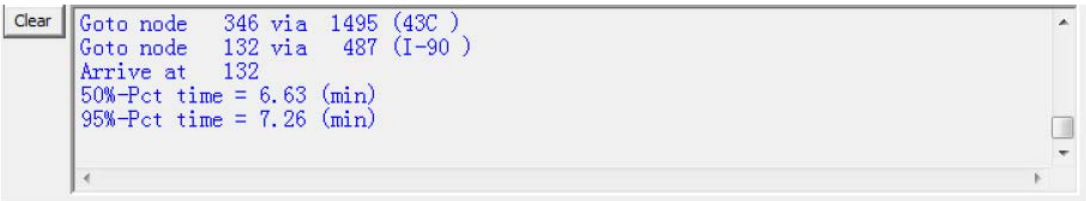


Figure 5.14 Path choice in Case 3.

Display CDF and PDF: Double-click Path 6 to display CDF and PDF of the path, as shown in Figure 5.15. The minimum travel time on Path 1 is 6.29 minutes, the maximum travel time is 8.35 minutes, the mean travel time is 6.70 minutes, and the standard deviation is 0.29 minutes. The standard deviation is much less than during the week. On weekends, road capacity is greater than demand so traffic speed is close to free-flow speed. As shown in Figure 5.15, the CDF curve shows that the 50th and 95th percentile travel times are 6.63 minutes and 7.26 minutes, respectively. The B-index is 0.91, and the P-index is 1.08.

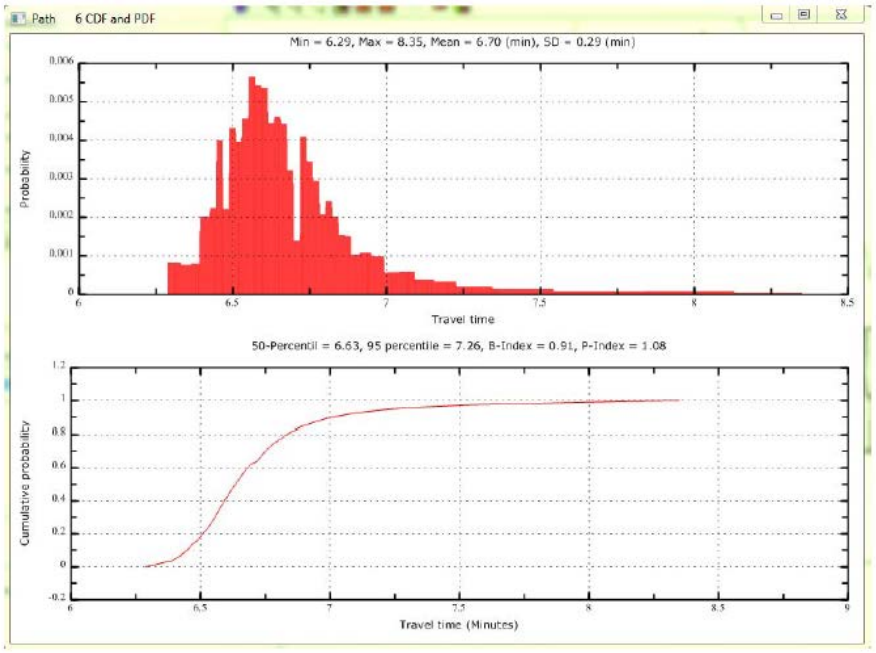


Figure 5.15 CDF and PDF in Case 3.

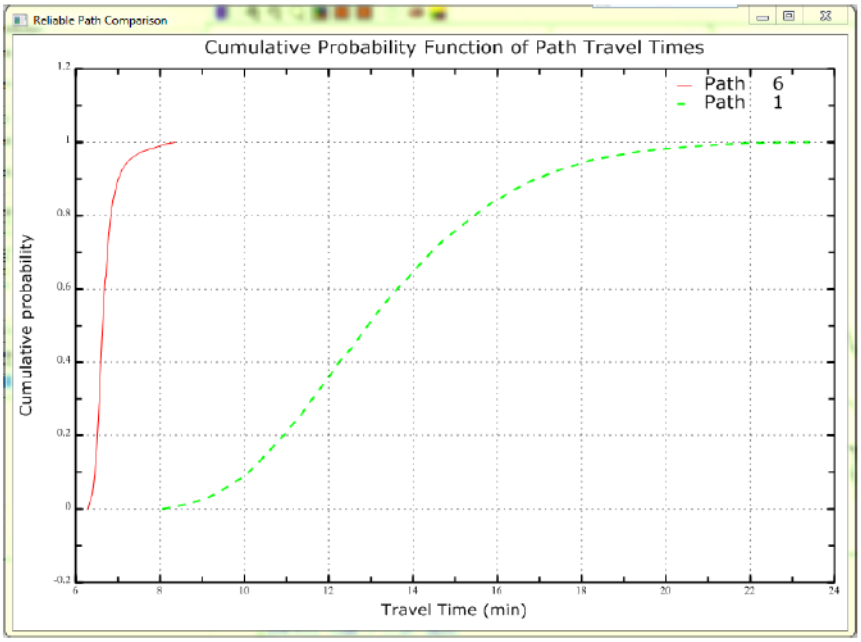


Figure 5.16 Path comparison in Case 3.

Compare different paths: We can compare the CDF of Path 6 (WEEKEND AMPEAK) and Path 1 (WEEDDAY AMPEAK). As shown in Figure 5.16, the difference in the two paths' travel times is obvious. Travel time on Path 6 for any percentile is far less than that on Path 1, and the time variation for Path 6 is smaller than that on Path 1. In short, the performance of Path 6 is better than that of Path 1. (Note: This comparison does not provide any information about path choice because Path 6 and Path 1 are generated for different

days of the week. It only illustrates how the day of the week may affect the travel time distribution.)

Adjust on-time arrival probability: We can also adjust on-time arrival probability for Path 6, as before. Table 5.3 summarizes several percentile travel times corresponding to different on-time arrival probabilities for Path 6.

Table 5.3 Travel Time Budgets for Different On-Time Arrival Probabilities

On-time arrival probability	Percentile travel time(min)
10.8%	6.5
50%	6.6
78.6 %	6.8
95%	7.3

5.2 INTEGRATION WITH GIS

In this section, we integrate our results from TRIC with GIS software packages to demonstrate the capability of TRIC for graphical comparisons. In our example, Quantum GIS is used for demonstration purposes. The same logic applies to other GIS software packages.

Assuming the default install directory, all GIS files can be found at *...\NU-TREND\TRICSetup\Data and\NU-TREND\TRICSetup\TRI*. The Data folder is used to store network information, which includes CMAP_ntwk and CMAP_nodes. The TRI folder stores link and O-D information generated by TRIC. Because all required attributes have been generated by TRIC, we can easily import .shp files into the GIS package. Then we can use GIS software for graphical comparison. (Details about GIS software are not elaborated because they are beyond the scope of this report.)

We first compared the B-index and P-index for the three cases. We colored all links in the network based on their indices (Figures 5.17 through 5.20). It is readily apparent that both indices reach the highest value during AMPEAK of WEEKDAY. On the other hand, they plummet during MIDDAY of WEEKEND. This is reasonable because the numerators of both indices are the 95th percentile travel time, which increases when traffic becomes congested.

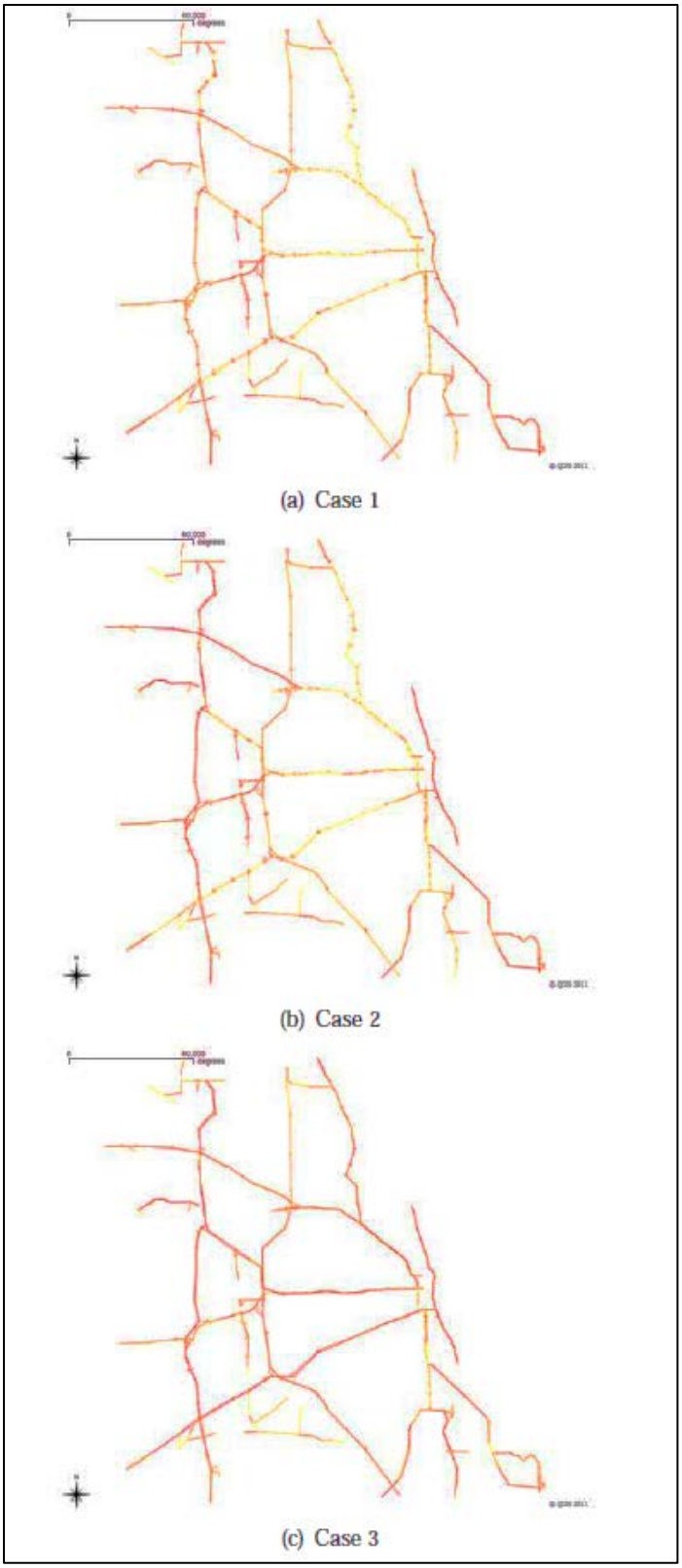
We can see from the figures that the area around Chicago Loop is more congested than the suburban areas. In addition, the indices for the routes between our O-D pair (Node 908 and Node 132) vary widely during different time periods. The justifies use of TRIC to find the most reliable path during different time periods.

Next, we focused our analysis on six major freeways/arterials around the Chicago area (I-90/94 East, I-290, I-55, I-90/94 West, N. Lake Shore Dr., and S. Lake Shore Dr.). These freeways/arterials are major connections between Chicago and its suburban neighborhoods. We studied the indices for all these freeways/arterials at the same time. The purpose of the investigation was to shed light on traffic reliability data around the Chicago area during different time periods.

A close examination of Figures 5.17 through 5.20 reveals that (1) indices of different routes are different at the same time of day and (2) indices for the same route vary for different travel times. For example, I-90/94 West is less reliable during morning rush hour. This is mainly due to the large traffic flow coming from the northwestern suburbs. For the same time period, I-90/94 West becomes more reliable compared with other routes on

weekdays at midday and on weekends during peak morning traffic hours. However, during those times, I-55 becomes less reliable.

In short, TRIC reveals varying indices for different routes during different time periods. These indices show the risk associated with traveling a particular route. With TRIC, travelers benefit from better planning based on their travel time budget and risk level. Using TRIC by itself provides estimated travel time between O-D pairs. Integrating TRIC with GIS generates information about macroscopic traffic conditions.

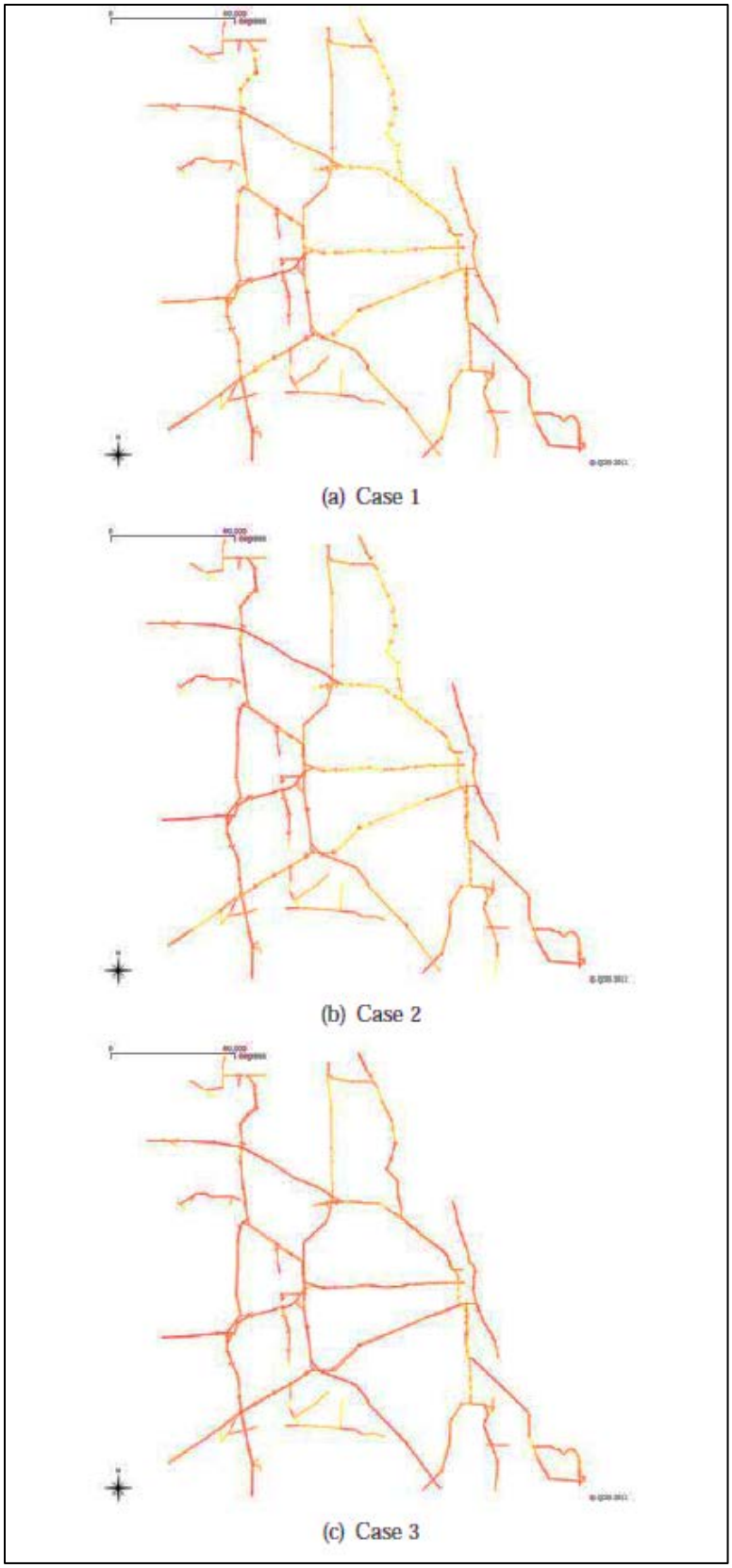


(a) Case 1

(b) Case 2

(c) Case 3

Figure 5.17 B-index for the studied cases.



(a) Case 1

(b) Case 2

(c) Case 3

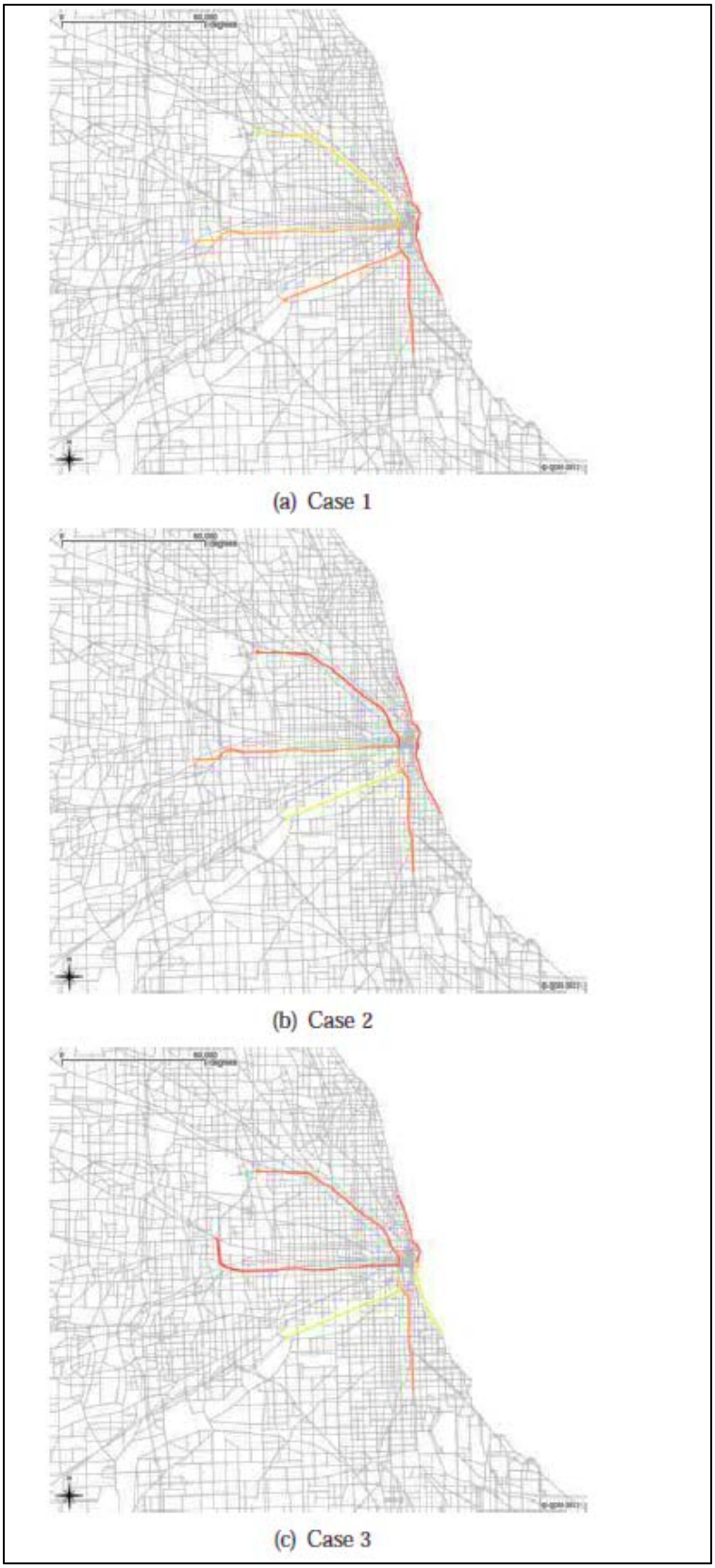
Figure 5.18 P-index for the studied cases.

(a) Case 1

(b) Case 2

(c) Case 3

Figure 5.19 B-index for the studied cases.



(a) Case 1

(b) Case 2

(c) Case 3

Figure 5.20 P-index for the studied cases.

CHAPTER 6 CONCLUSIONS

Travel time reliability is important to highway users. Personal and business travelers value reliability because it allows them to make better use of their time. Shippers and freight carriers require predictable travel times to remain competitive in the market. The lack of travel reliability forces motorists to choose between running the risk of being late (therefore missing important appointments or just-in-time deliveries) or budgeting a large buffer time, much of which is often wasted.

To hedge against such uncertainty, highway users need decision-supporting tools that are capable of exploiting existing data sources to (1) reveal and document random patterns of travel times on highway networks and (2) provide real-time routing decision-making tools that take into account the uncertainties of travel time and traveler requirements for reliability.

We addressed this challenge by developing a software program called Travel Reliability Inventory for Chicago (TRIC), which not only helps visualize and archive statistics information for link and path travel times but also provides a reliable routing decision-making tool. In this chapter, we first summarize what was accomplished in this project and then discuss possible directions for further research.

6.1 MAIN RESULTS

We proposed a reliable *a priori* shortest path (RASP) problem to find the shortest *a priori* paths so that a traveler can arrive at his/her destination on time, or earlier, at a desired probability. We found that solving this problem is equivalent to solving the non-dominated paths under the first-order stochastic dominance rule. We showed that Bellman's optimality principle can be applied to solve all non-dominated paths. In light of this, we developed a label-correcting solution algorithm.

The path travel time distribution was computed by convolving the travel time distribution of its member links. The convolution process was found to have a strong effect on the computational performance of the solution algorithm. To simplify the implementation of integral convolution, the distributions were all discretized. Some solution techniques were proposed and tested to try to improve the computational performance of the solution algorithm. We found that the direct convolution scheme based on a discretization method called the adaptive discretization approach (ADA) achieved the best balance between accuracy and computational efficiency. The ADA-based direct convolution scheme was employed in TRIC to solve the reliable *a priori* shortest paths problem.

The premise in solving this problem is that we have perfect information about link travel time distributions throughout the entire network. We also assumed that the link travel time distributions are not correlated to each other. In one of our previous research projects (Nie et al. 2009), we constructed the travel time distributions of expressway links during different time periods based on the traffic data collected by loop detectors and I-PASS plazas in the Chicago metropolitan area. TRIC contains the data for these link travel time distributions and provides tools to visualize them, as well as to archive them as a shape files for GIS applications.

6.2 FUTURE WORK

Because computational efficiency is always a concern in achieving reliable routing guidance in real time (as with a on-vehicle navigation system), the computational performance of our solution algorithm to the RASP problem could be refined.

Our RASP model does not consider the link travel time correlation, which implies that travel time in one link has nothing to do with that of its adjacent links, which is contrary to beliefs about traffic congestion. The research team did not employ Markovian probability to study the correlation between link travel times (Nie and Wu 2009); therefore, some assumptions must be imposed on transition mapping. For example, if one link has multiple adjacent upstream links, the transition probabilities have to be identical. However, these assumptions were not verified with the real traffic data we obtained from loop detectors in the Chicago metropolitan area. Finding an adequate method to model the link travel time correlation is an interesting direction for further research.

Currently, TRIC covers just the interstate highways and expressways, which accounts for only a small portion of the links in the Chicago metropolitan area. We estimated the travel time distribution of arterial roads and local streets through congestion data from the Chicago Metropolitan Agency Planning (CMAP) (Nie et al. 2009). We also attempted to use Chicago Transit Authority (CTA) buses as traffic probes to evaluate traffic speed in the network of Chicago. However, travel times obtained from CTA buses did not present a clear pattern of distributions because buses make frequent stops (Nie et al. 2010). To obtain travel time data for arterial streets, additional data sources should be exploited. More and more motorists are using the navigation functions of smart phones when driving and motorist travel information is being collected by wireless service providers. Cell phone data represent the actual traffic situation better than the bus data do because cell phone data are associated with individual vehicles in most cases. Moreover, cell phone data provide broader coverage than bus data because not all local streets are covered by bus routes. Therefore, cell phone data may provide high-quality information about individual trips, which can be used to construct distributions of travel times on arterial roads and local streets.

REFERENCES

Bander, J. L. and C. C. White. A heuristic search approach for a nonstationary stochastic shortest path problem with terminal cost. *Transportation Science*, 36(2):218–230, 2002.

Bard, J.F. and J.E. Bennett. Arc reduction and path preference in stochastic acyclic networks. *Management Science*, 37(2):198–215, 1991.

Bawa, V. S. Optimal rules for ordering uncertain prospects. *Financial Economics*, 2:95–121, 1975.

Bawa, V. S., J. Bondurtham, R. Rao, and H. L. Suri. On determination of stochastic dominance optimal set. *Journal of Finance*, 40:417–431, 1983.

Bertsimas, D. and M. Sim. Robust discrete optimization and network flow. *Mathematical Programming Series B*, 98:49–71, 2003.

Bracewell, R.N. *The Fourier Transform and Its Applications*. McGraw-Hill, Boston, 3rd edition, 2000.

Brumbaugh-Smith, J. and D. Shier. An empirical investigation of some bicriterion shortest path algorithms. *European Journal of Operational Research*, 43(2):216–224, 1989.

Cambridge Systematics Inc. and Texas Transportation Institute (TTI). Traffic congestion and reliability: Trends and advanced strategies for congestion mitigation. Technical report, Federal Highway Administration, 2005.

Carraway, R. L., T. L Morin, and H. Moskowitz. Generalized dynamic programming for multicriteria optimization. *European Journal of Operational Research*, 44(1):95–104, 1990.

Chen, Yanyan , Michael G. H. Bell, Dongzhu Wang, and Klaus Bogenberger. Risk-averse time-dependent route guidance by constrained dynamic a* search in decentralized system architecture. *Transportation Research Record*, 1944:51–57, 2006.

Cheung, R. K. Iterative methods for dynamic stochastic shortest path problems. *Naval Research Logistics*, 45(8):769–789, 1998.

Cormen, T.H., C.E. Leiserson, R.L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, Cambridge, MA, 2001.

Dentcheva, D. and A. Ruszczynski. Optimization with stochastic dominance constraints. *SIAM Journal of Optimization*, 14(2):548–566, 2003.

Eiger, A., P. B. Mirchandani, and H. Soroush. Path preferences and optimal paths in probabilistic networks. *Transportation Science*, 19(1):75–84, 1985.

Fan, Y., R. Kalaba, and J. Moore. Arriving on time. *Journal of Optimization Theory and Applications*, 127(3):497–513, 2005a.

Fan, Y., R. Kalaba, and J. Moore. Shortest paths in stochastic networks with correlated link costs. *Computers and Mathematics with Applications*, 49(9-10):1549–1564, 2005b.

Federal Highway Administration (FHWA). *Traffic Incident Management Handbook*. Federal Highway Administration, 2000.

Frank, H. Shortest paths in probabilistic graphs. *Operations Research*, 17(4):583–599, 1969.

Friedman, M. and L.P. Savage. The utility analysis of choices involving risk. *Journal of Political Economy*, 56:279–304, 1948.

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

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

Gao, S. and I. Chabini. Optimal routing policy problems in stochastic time-dependent networks. *Transportation Research Part B*, 40(2):93–122, 2006.

Hadar, J. and W. R. Russell. Stochastic dominance and diversification. *Journal of Economic Theory*, 3:288–305, 1971.
Hall, R. W. The fastest path through a network with random time-dependent travel time. *Transportation Science*, 20(3):182–188, 1986.
Hanoch, G. and H. Levy. The efficiency analysis of choices involving risk. *Review of Economics studies*, 36:335–346, 1969.
Hansen, P. Bicriterion path problems. In G. Fandel and T. Gal, editors, *Multiple criteria decision making theory and application*, pages 109–127, 1979.
Henig, M. I. The shortest path problem with two objective functions. *European Journal of Operational Research*, 25(2):281–291, 1985.
Heyer, Daniel D. Stochastic dominance: A tool for evaluating reinsurance alternatives. In *Casualty Actuarial Society Forum*, 2001.
Kaparias, I., M.G.H. Bell, Y. Chen, and K. Bogenberger. Icnavs: A tool for reliable dynamic route guidance. *IET Intelligent Transportation Systems*, 1(4):225–253, 2007.
Levy, H. and G. Hanoch. Relative effectiveness of efficiency criteria for portfolio selection. *Journal of Financial and Quantitative Analysis*, 5:63–76, 1970.
Lo, Hong K. and Yeou-Kuong Tung. Network with degradable links: Capacity analysis and design. *Transportation Research Part B*, 37(4):345–363, 2003.
Lo, Hong K., X.W. Luo, and Barbara W. Y. Siu. Degradable transport network: Travel time budget of travelers with heterogeneous risk aversion. *Transportation Research Part B*, 40(9):792–806, 2006.
Loui, R. P. Optimal paths in graphs with stochastic or multidimensional weights. *Communications of the ACM*, 26(9):670–676, 1983.
Miller-Hooks, E. *Optimal Routing in Time-Varying, Stochastic Networks: Algorithms and Implementations*. PhD thesis, Department of Civil Engineering, University of Texas at Austin, 1997.
Miller-Hooks, E. D. Adaptive least-expected time paths in stochastic, time-varying transportation and data networks. *Networks*, 37(1):35–52, 2001.
Miller-Hooks, E. D. and H. S. Mahmassani. Least possible time paths in stochastic, time-varying networks. *Computers and Operations Research*, 25(2):1107–1125, 1998a.
Miller-Hooks, E. and H. Mahmassani. Optimal routing of hazardous materials in stochastic, time-varying transportation networks. *Transportation Research Record*, 1645:143–151, 1998b.
Miller-Hooks, E.D. and H. S. Mahmassani. Least expected time paths in stochastic, time-varying transportation networks. *Transportation Science*, 34(2):198–215, 2000.
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*, 146(2):67–82, 2003.
Mirchandani, P.B. Shortest distance and reliability of probabilistic networks. *Computers and Operations Research*, 3(4):347–355, 1976.
Montemanni, R. and L. Gambardella. An exact algorithm for the robust shortest path problem with interval data. *Computers and Operations Research*, 31(10):1667–1680, 2004.
Muller, A. and D. Stoyan. *Comparison Methods for Stochastic Models and Risks*. John Wiley & Sons, Chichester, UK, 2002.
Murthy, I. and S. Sarkar. A relaxation-based pruning technique for a class of stochastic shortest path problems. *Transportation Science*, 30(3):220–236, 1996.
Murthy, I. and S. Sarkar. Stochastic shortest path problems with piecewise linear concave linear functions. *Management Science*, 44(11):125–136, 1998.
Nie, Y. *A Programmer's Manual for Models and Algorithms Toolkit (MAT)*. University of California, Davis, 2006.

Nie, Y. and X. Wu. Reliable a priori shortest path problem with limited spatial and temporal dependencies. *In Proceedings of the 18th International Symposium on Transportation and Traffic Theory*, pages 169–196, 2009.
Nie, Y., X. Wu, P. Nelson, and J. Dillenburg. Providing reliable route guidance using Chicago data. Technical Report 2009-001 1, Center for the Commercialization of Innovative Transportation Technology, Northwestern University, Evanston, IL, 2009.
Nie, Y., X. Wu, J. Zissman, C. Lee, and Michael Haynes. Providing reliable route guidance: Phase ii. Technical report, Center for the Commercialization of Innovative Transportation Technology, Northwestern University, Evanston, IL 2010.
Polychronopoulos, G. H. and J. N. Tsitsiklis. Stochastic shortest path problems with recourse. *Networks*, 27(2):133–143, 1996.
Provan, J. S. A polynomial-time algorithm to find shortest paths with recourse. *Networks*, 41(2):115–125, 2003.
Rothschild, M. and J. E. Stiglitz. Increasing risk. I. a definition. *Journal of Economic Theory*, 2:225–243, 1970.
Schrank, D. and T. Lomax. 2010 urban mobility study. Technical report, Texas Transportation Institute, Texas A&M University, College Station, TX, 2011.
Sen, S., R. Pillai, S. Joshi, and A. Rathi. A mean-variance model for route guidance in advanced traveler information systems. *Transportation Science*, 35(1):37–49, 2001.
Sigal, C. E., A. Alan, B. Pritsker, and J. J. Solberg. The stochastic shortest route problem. *Operations Research*, 28(5):1122–1129, 1980.
Sivakumar, R. and R. Batta. The variance-constrained shortest path problem. *Transportation Science*, 28(4):309–316, 1994.
Waller, S. T. and A. K. Ziliaskopoulos. On the online shortest path problem with limited arc cost dependencies. *Networks*, 40(4):216–227, 2002.
Waller, S.T., K. C. Mouskos, D. Kamaryiannis, and A. K. Ziliaskopoulos. A linear model for continuous network design problem. *Computer-aided civil and infrastructure engineering*, 21:334–345, 2006.
Whitmore, G. A. Third degree stochastic dominance. *American Economic Review*, 60:457–459, 1970.
Wu, X. and Y. Nie. Implementation issues in approximate algorithms for reliable a priori shortest path problem. *Journal of the Transportation Research Board*, 2091:51–60, 2009.
Wu, X. and Y. Nie. Modeling heterogeneous risk-taking behavior in route choice: A stochastic dominance approach. *Transportation Research Part A*, 45: 896-915, 2011.
Wu, X. and Y. Nie. Application of discrete Fourier transform in finding reliable shortest paths. *Journal of the Transportation Research Board*, forthcoming.
Yu, G. and J. Yang. On the robust shortest path problem. *Computers and Operations Research*, 25(6):457–468, 1998.


Illinois Department
of Transportation


ILLINOIS
UNIVERSITY OF ILLINOIS AT URBANA-CHAMPAIGN