The Hong Kong Polytechnic University

Department of Civil and Structural Engineering

Reliable Shortest Path Problems in Networks under Uncertainty: Models, Algorithms and Applications

Bi Yu CHEN

A thesis submitted in partial fulfillment
of the requirements for the
degree of Doctor of Philosophy

July 2011

Certificate of Originality

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it produces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgment has been made in the text.

Signature:_____

Bi Yu CHEN

Abstract

Link travel times in congested urban road networks are highly stochastic. Many empirical studies have found that travellers on such networks prefer to choose reliable shortest paths (RSP) for their travel so that they can arrive at destinations with a higher on-time arrival probability. Therefore, it is necessary to investigate the problems of finding these RSP in realistic road networks with travel time uncertainties.

It is acknowledged that the RSP problems are significant; nonetheless few effective and efficient methods have been recorded in the literature. This is mainly due to the non-additive objective function of the RSP problems. Classical shortest path algorithms, built on the additive property, cannot be used to solve the RSP problems. In view of the above, the research presented in this thesis reveals model and solution algorithm for solving such RSP problems together with illustrations of their applicability in various transportation fields. This thesis contributes to the literature of RSP problems in several aspects.

Firstly, a multi-criteria shortest path-finding model is proposed to tackle the non-additive difficulty involved in RSP problems. Several dominance conditions of the RSP problems are established to enable the use of generalized dynamic programming approaches for solving the RSP problems. In the proposed RSP model, the travel time spatial correlations among k-neighboring links are considered. This limited travel time spatial dependence can be interpreted as Tobler's First Law of Geography that "all things are related, but nearby things are more related than distant things". An efficient multi-criteria A* algorithm is proposed to determine the RSP without the requirement of generating all non-dominated paths in the entire network. A case study using traffic data from a real-world advanced traveller information system (ATIS) is provided to validate the proposed solution algorithm.

The proposed RSP model and solution algorithm are extended to incorporate travel time temporal correlations in those stochastic time-dependent (STD) networks where link travel time distributions vary by time intervals throughout the day. In the STD networks, travellers' experienced link travel time variation depends on the time instance vehicles entering the link; and the link travel time distribution is typically assumed to be fixed when these vehicles travelling on that link. This assumption, however, may violate the first in first out (FIFO) property, since traffic conditions cannot be updated when vehicles travelling on the link. To address this non-FIFO problem, a stochastic travel speed model (S-TSM) that can update travellers' experienced travel speeds during different time intervals on the link is proposed in this research. The proposed S-TSM can ensure the FIFO property of link travel times, so that the efficient multi-criteria A* algorithm can be adopted to solve the RSP problems in STD networks. Based on the proposed multi-criteria A* algorithm, a real-world ATIS-based routing system is developed to aid road users of Hong Kong making route choice decisions in road networks with travel time spatiotemporal correlations.

Secondly, the proposed RSP model is incorporated in reliability-based user equilibrium (RUE) problems for traffic assignment. In this research, an effective reliable shortest path algorithm is developed to determine RSP for all user classes in one search process so as to avoid the repeated path searching for each user class. The proposed reliable shortest path algorithm is then, further incorporated into a path-based RUE assignment algorithm using a column generation method. The proposed RUE assignment algorithm does not require path enumeration and can achieve highly accurate RUE results within reasonable computational time. A numerical example demonstrates that the proposed RUE assignment algorithm is capable for solving relevant problems in road networks with demand and / or supply uncertainties.

Thirdly, the proposed RSP and RUE algorithms are applied to identify critical links in large-scale road networks. The traditional method, to identify critical links, is to use a full scan approach to assess all possible link closure scenarios by means of traffic assignment methods. This full scan approach is not viable for identifying critical links in large-scale road networks, because of the large number of link closure scenarios and computational intensity of traffic assignment methods in these large-scale networks. An impact area vulnerability analysis approach is proposed in this research to evaluate the consequences of a link failure within a local impact area, rather than the entire network. Such vulnerability analysis approach reduces the problem size of the critical link identification so as to reduce the computational burden involved. Case studies on large-scale real-world networks are presented to illustrate the proposed impact area vulnerability approach and investigate the effects of stochastic demand and heterogeneous travellers' risk-taking behaviour.

Publications Arising From the Thesis

Journal papers:

1. Chen, B.Y., Lam, W.H.K., Sumalee, A. and Shao, H., 2011, An efficient solution algorithm for solving multi-class reliability-based traffic assignment problem. *Mathematical and Computer Modelling*, 54, pp. 1428-1439.
2. Chen, B.Y., Lam, W.H.K., Sumalee, A., Li, Q. and Li, Z.C., 2012, Vulnerability analysis for large-scale and congested road networks with demand uncertainty. *Transportation Research Part A-Policy and Practice*, 46, pp. 501-516.
3. Chen, B.Y., Lam, W.H.K., Sumalee, A. and Li, Z. L., 2011, Reliable shortest path finding in stochastic networks with spatial correlated link travel times. *International Journal of Geographical Information Science*. DOI: 10.1080/13658816.2011.598133. In press.
4. Chen, B.Y., Lam, W.H.K., Sumalee, A. Li, Q. and Tam, M. L., 2011, Reliable shortest path problems in stochastic time-dependent networks. *Journal of Intelligent Transportation Systems*. Accepted for publication.
5. Chen, B.Y., Lam, W.H.K., Sumalee, A., Li, Q. and Shao, H., 2011, Finding reliable shortest paths in road networks under uncertainty. *Networks & Spatial Economics*. Revised.
6. Chen, B.Y., Lam, W.H.K. and Tam, M. L., 2009, Modeling departure time and route choice problems in stochastic road networks for online ATIS applications. *Journal of the Eastern Asia Society for Transportation Studies*, 8, pp. 1768-1777.

Conference papers:

7. Chen, B.Y., Lam, W.H.K., Shao, H. and Tam, M.L., 2009, A novel solution algorithm for solving multi-class reliability-based path choice problems. In: *14th International Conference of Hong Kong Society for Transportation Studies*, D. Wang and S.M. Li (Eds.), Hong Kong, pp. 139-147.

8. Chen, B.Y., Lam, W.H.K., Tam, M.L. and Sumalee, A., 2009, Finding reliable shortest paths in the stochastic time-dependent road networks for online ATIS applications. In: *3th International Forum on Shipping, Ports and Airports. Hong Kong*, China, pp. 28-36.
9. Chen, B.Y., Lam, W.H.K. and Tam, M.L., 2008, Using a multi-criteria approach to solve path finding problem in road network with uncertainty. In: *13th International Conference of Hong Kong Society for Transportation Studies*, H.P. Lo, C.H. Leung and M.L. Tam (Eds.), Hong Kong, China, pp. 229-238.

Acknowledgments

I would like to express my deepest gratitude to my chief supervisor and mentor, Prof. William H. K. Lam, for his guidance, advice and patience throughout my study in Hong Kong. This thesis could not be completed without his encouragements and insightful inputs. Beside knowledge, he provides me self-motivated and structured working attitude that substantially influences the rest of my career.

I would like to express my sincere gratitude to my co-supervisor, Dr. Agachai Sumalee, for his guidance and constructive suggestions. I was moved and inspired every time when I read my papers on which his revisions and comments fill the margin of every single page. I also would like to thank Prof. Liping Fu and Prof. Janny Leung for serving in my thesis committee, and for their valuable suggestions and advices.

Special thanks must be given to Dr. Hu Shao and Prof. Zhi-Chun Li for guiding me into the world of transportation modelling. Special thanks should be given to Prof. Xiaoling Chen, Prof. Onyx Wai, Prof. Zhilin Li, Prof. Chung-lun Li, Prof. Michael Bell, Prof. Michael Taylor and Prof. Anthony Chen. Discussions with them have benefited my research. I would also like to express special thanks to Prof. Elaine Anson for perfecting my thesis to ‘elegant’ form.

Grateful appreciation should be extended to all research staffs in the transportation research group at PolyU not limited to: Dr. Karen Tam, Dr. Zhongyi Wu, Dr. Bin Yu, Dr. Jiancheng Long, Dr. Jisuan Zhu and Dr. Wenzhu Zhou. I also would like to acknowledge fellow research students: Dr. Renxin Zhong, Dr. Paramet Luathep, Lianqun OuYang, Ding Liu, Yuqing Zhang, DaoJun Ye, Yiliang Xiong, Xiao Fu, Hua Wang, Jianting Cong and Lingyun Yuan.

Above all, I am deeply thankful to my wife, Miss Huiping Chen, and my family for their endless love, care, and support in no matter what I choose to pursue in life.

Table of Contents

Certificate of Originality 2
Abstract 3
Publications Arising From the Thesis 6
Acknowledgments 8
Table of Contents 9
List of Tables 11
List of Figures 12
Notation 13
1. Introduction 18
    1.1. Need for the Study 18
    1.2. Objectives of the Research 20
    1.3. Structure of the Thesis 23
2. Backgrounds and Literature Review 26
    2.1. Advanced Traveller Information Systems 26
    2.2. Travel Time Variability and Travellers' Risk-taking Behaviour 29
    2.3. Stochastic Shortest Path Algorithms 33
    2.4. Reliability-based Traffic Assignment 36
    2.5. Network Vulnerability Analysis 38
    2.6. Summary 41
3. Reliable Shortest Path Problems 43
    3.1. Background 43
    3.2. Problem Definition 45
    3.3. Multi-criteria Shortest Path Approach 48
    3.4. Solution Algorithms for Solving RSPP 54
    3.5. Computational Performance 63
    3.6. Extension to Lognormal Distributions 68
    3.7. Summary 69
4. Spatial-dependent Reliable Shortest Path Problems 71
    4.1. Background 72
    4.2. Problem Statement 73
    4.3. Multi-criteria Shortest Path Approach for Solving SD-RSPP 76
    4.4. Two-level Hierarchical Network 79
    4.5. Solution Algorithm 84
    4.6. Case Study in Hong Kong 89
    4.7. Summary 95
5. Time-dependent Reliable Shortest Path Problems 97
    5.1. Background 98
    5.2. Problem Statement 101
    5.3. Stochastic Travel Speed Model 103
    5.4. Non-reversibility of TD-RSPP 111

5.5. Methods for Generating Path Travel Time Distribution 120
5.6. Case Study in Hong Kong 123
5.7. Computational Performance 125
5.8. Summary 129
6. Reliable Path Searching System: Case Study in Hong Kong 132
6.1 Background 132
6.2 Network Representation 133
6.3 Generalized Reliable Shortest Path Problem 137
6.4 System Architecture 142
6.5 Estimation of Travel Time Distributions 144
6.6 Experimental Results 146
6.7 Summary 148
7. An Efficient Solution Algorithm for Solving Multi-Class Reliability-Based Traffic Assignment Problem 150
7.1. Background 151
7.2. Definitions and Problem Statement 153
7.3. Solution Algorithm 156
7.4. Numerical Examples 162
7.5. Summary 169
8. Vulnerability Analysis for Large-scale and Congested Road Networks with Demand Uncertainty 171
8.1. Background 172
8.2. Network Efficiency and Vulnerability Index 173
8.3. Impact Area Vulnerability Analysis Approach 175
8.4. Case Studies on Real-world Road Networks 180
8.5. Summary 194
9. Conclusions and Further Studies 196
9.1. Summary of Research Findings 196
9.2. Recommendations for Further Studies 200
Appendix A 204
Appendix B 206
Appendix C 209
Appendix D 211
References 213

List of Tables

Table 1.1 Classification of reliable shortest path problems 20
Table 2.1 Classification of stochastic shortest path problems 33
Table 2.2 The characteristics of road network vulnerability indices 40
Table 3.1 Basic characteristics of testing networks 64
Table 3.2 Computational performance of algorithms for risk-averse scenarios 65
Table 3.3 Computational performance of algorithms for risk-seeking scenarios 68
Table 4.1 The sizes of impact area and hierarchical network under different *k* values 93
Table 4.2 Computation performance of *SDRSP-HA** algorithm 94
Table 5.1 The steps of the proposed *B-TDRSP* algorithm 119
Table 5.2 Path travel time distributions using both S-LTM and S-TSM 124
Table 5.3 Basic characteristics of testing networks 126
Table 5.4 Computational performance of algorithms for solving TD-RSPP 127
Table 5.5 Computational time of the *F-TDRSP-A** algorithm in seconds 128
Table 6.1 Departure time and route choices between RG and CBD 148
Table 7.1 Summary of solution algorithms for solving UE problems 151
Table 7.2 Basic characteristic of test networks 163
Table 7.3 The computational time of reliable shortest path algorithms (millisecond) 164
Table 7.4 Performance of *RUE-PB* algorithm under different network sizes 166
Table 7.5 Algorithm performance on Chicago network under different class numbers 168
Table 8.1 Impact area vulnerability analysis for various demand uncertainty 185
Table 8.2 Impact area vulnerability analysis for various impact area parameters 189
Table 8.3 The accuracy of partial network scan approach under different parameters 191
Table 8.4 Computational performance of the partial network scan approach 193
Table 9.1 Summary of reliable shortest path problems 196

List of Figures

Figure 1.1 The inter-relationships of research objectives 22
Figure 1.2 Structure of the thesis 24
Figure 2.1 Typical framework of ATIS application 28
Figure 2.2 Travel time information collected on a route in Hong Kong 29
Figure 2.3 A typical traffic flow pattern after the traffic incident 39
Figure 3.1 An illustration of non-additive property of RSPP 48
Figure 3.2 First-order stochastic dominance 50
Figure 3.3 Mean-variance dominance (a) risk-averse (b) risk-seeking 51
Figure 3.4 Mean-travel time budget dominance (a) risk-averse (b) risk-seeking 52
Figure 3.5 An illustration of update operation 62
Figure 3.6 Non-dominated paths ($\tilde{n}$) generated by three different algorithms 67
Figure 4.1 An illustrative example 76
Figure 4.2 An illustration of dominance conditions 78
Figure 4.3 An illustration of hierarchical network 80
Figure 4.4 Hong Kong RTIS network 89
Figure 4.5 Histogram of correlation coefficient between adjacent links 90
Figure 4.6 Travel time correlations among *k*-neighboring links 92
Figure 5.1 A simple illustrative example for the S-LTM and the S-TSM 106
Figure 5.2 An illustrative example of the non-reversibility property 112
Figure 5.3 Accuracy of path travel time estimation method 125
Figure 5.4 Comparison of computational results using α-discrete and mean-variance approximations 129
Figure 6.1 Turn restrictions at a simple street intersection 135
Figure 6.2 Data structure in the memory 136
Figure 6.3 The system framework of RPSS 142
Figure 6.4 User interface for the reliable routing service (a) input routing criteria (b) reliable path searching results 144
Figure 6.5 Results of reliable route searching from RG to CBD 146
Figure 7.1 Relative gap vs. computational time for the Chicago Sketch network 166
Figure 7.2 Computational time under different levels of congestions and demand variations 167
Figure 8.1 The Sioux Falls network 182
Figure 8.2 Results of vulnerability analysis under various demand multipliers and CV values 183
Figure 8.3 Correlation between global and impact area vulnerability 185
Figure 8.4 The road network of Hong Kong 186
Figure 8.5 The most critical links of the Hong Kong road network when CV=0.1 188
Figure 8.6 The most critical links of the Hong Kong road network when CV=0.5 188
Figure 8.7 Correlation between global and impact area vulnerability rankings when CV=0.1 190
Figure 8.8 Correlation between global and impact area vulnerability rankings when CV=0.5 192

Notation

A	A set of links
$\|A\|$	Number of links in the network
$\|\hat{A}\|$	Number of links in the top hierarchical network
a_{ij}	A link connecting node i and node j
a^m	The mth link along path p_u^{rs}
$\hat{a}_{iw,u}$	A top hierarchical link representing a primal path $p_u^{iw,k}$
b	Travel time budget
$BORDER(\overline{G}_i^g)$	Border nodes of directed-in-tree $\overline{G}_i^g$
c_a	Link capacity
$\text{cov}(T^m, T^n)$	Travel time covariance between links a^m and a^n
$\text{cov}(\hat{T}_{iw,u}, \hat{T}_{jq,u})$	Travel time covariance between top hierarchical links $\hat{a}_{iw,u}$ and $\hat{a}_{jq,u}$
$\text{cov}(V_{ij}^k, V_{ij}^w)$	Travel speed temporal correlations between two time intervals
cv^{rs}	Coefficient of variation of travel demand distribution between origin and destination
d_{is}	Network distance from node i to destination
d_{ij}	Link length of a_{ij}
d_{ij}^{qw}	Topological distance between links a_{ij} and a_{qw} (measured by number of links)
$D_{y_i}^{y_a}$	Travel distance distribution during time period (y_i, y_a)
$d_{y_i}^{y_a}$	Mean travel distance during time period (y_i, y_a)
e_{is}	Euclidean distance from node i to destination
$E(\rho^k)$	Mean absolute value (MAV) of correlation coefficients for all *k*-neighboring links
$\overline{E}(G)$	Efficiency of network G under normal condition
$\overline{E}_{ij}(G)$	Efficiency of network G after the closure of link a_{ij}
$F(p_u^{ri})$	Heuristic value function for path p_u^{rs}
$F(\hat{p}_u^{rj})$	Heuristic value function for top hierarchical path $\hat{p}_u^{rj}$

F_k^{rs}	Path travel flow distribution along path p_k^{rs}
f_k^{rs}	Mean travel flow along path p_k^{rs}
$f(\cdot)$	Probability density function (PDF) of a random variable
G	Directed network (or primal network)
G_{ij}^k	A local impact area of link a_{ij}
$\overline{G}_i^g$	A directed-in tree originated at node i in the ground hierarchy
$h(i)$	Estimation of travel time budget from node i to destination
HG	Hierarchical network
H^g	Ground hierarchy
H^t	Top hierarchy
$LOCAL(\overline{G}_i^g)$	Local nodes of directed-in-tree $\overline{G}_i^g$
$\|M\|$	Number of user classes
N	A set of nodes
$\overline{n}_u^{ij,\lambda}$	A ground hierarchical node representing a primal path $p_u^{ij,\lambda}$
$\|N\|$	Number of nodes in the network
P^{rs}	A set of paths from origin to destination
$\|P\|$	Number of non-dominated paths between origin and destination
$\|\hat{P}\|$	Number of non-dominated paths between origin and destination in top hierarchical network
p_u^{rs}	A path from origin to destination
$p_u^{ij,k}$	A path from node i to node j with k consecutive links
$\hat{p}_u^{rs,\lambda}$	A top hierarchical path with λ consecutive top hierarchical links
$PDS(i)$	A set of predecessor nodes
Q_{rs}	Travel demand distribution between origin and destination
q^{rs}	Mean travel demand between origin and destination
r	Origin node
$rank_a^g$	Ranking of link a based on global vulnerability index
$rank_a^l$	Ranking of link a based on impact area vulnerability index
$rank_a^p$	Ranking of link a amongst selected candidate links
s	Destination node
$SCS(i)$	A set of successor nodes of node i
T_{ij}	Travel time distribution of link a_{ij}

$T_{ij}(y)$	Travel time distribution of link a_{ij} at time instant y
$T_{iw,u}^{\hat{a}}$	Travel time distribution of top hierarchical link $\hat{a}_{iw,u}$
t_{ij}	Mean travel time of link a_{ij}
t_{ij}^{n}	Mean travel time of link a_{ij} during time interval (Δ_{n-1}, Δ_n)
$t_{iw,u}^{\hat{a}}$	Mean travel time of top hierarchical link $\hat{a}_{iw,u}$
t_a^0	Link free flow travel time
T_u^{rs}	Travel time distribution of path p_u^{rs}
$T_{rs,u}^{\hat{p}}$	Travel time distribution of top hierarchical path $\hat{p}_u^{rs,\lambda}$
$t_{rs,u}^{\hat{p}}$	Mean travel time of top hierarchical path $\hat{p}_u^{rs,\lambda}$
t_u^{rs}	Mean travel time of path p_u^{rs}
$T_u^{rs}(y_r)$	Travel time distribution of path p_u^{rs} at departure time y_r
$u^{rs,m}$	Proportion of user class m out of the total travel demand
$v_{\max}$	Maximum travel speed in the network
V_{ij}^{n}	Travel speed distribution of link a_{ij} during time interval (Δ_{n-1}, Δ_n)
V_a	Link travel flow distribution
v_a	Mean link travel flow
VOD	Parameters for converting travel distance into the petrol cost
VOT	Value of time
VUL_{ij}^{g}	Globe vulnerability index of link a_{ij} closure
VUL_{ij}^{l}	Impact area vulnerability index of link a_{ij} closure
Y_i	Arrival time distribution at node i
y_r	Departure time at origin
y_s	Preferred arrival time at destination
Z_α	Inverse cumulative distribution function for standard normal distribution at α confidence level
α	Probability of on-time arrival
ρ_{ij}^{qw}	Correlation coefficient for two links a_{ij} and a_{qw}
σ_{ij}	Travel time standard deviation of link a_{ij}
$\sigma_{iw,u}^{\hat{a}}$	Travel time standard deviation of top hierarchical link $\hat{a}_{iw,u}$

σ_u^{rs}	Travel time standard deviation of path p_u^{rs}
$\sigma_{rs,u}^{\hat{p}}$	Travel time standard deviation of top hierarchical path $\hat{p}_u^{rs,\lambda}$
$\sigma_{D_{y_i}^{\Delta_m}}$	Travel distance standard deviation
σ_q^{rs}	Travel demand standard deviation between origin and destination
$\sigma_{f,k}^{rs}$	Path travel flow standard deviation along path p_k^{rs}
σ_v^a	Link travel flow standard deviation
δ_{ij}^{rs}	Link-path incidence relationship
Ψ	A set of allowed movements in the network
ψ_{ijk}	A movement from link a_{ij} to link a_{jk}
$\hat{\psi}_{iq,u}$	A top hierarchical movement representing a primal path $p_u^{iq,k+1}$
$\Phi(\cdot)$	Cumulative distribution function (CDF) of a random variable
$\Phi^{-1}(\cdot)$	Inverse cumulative distribution function of a random variable
Ω	Period of interest
Δ	Time interval
τ_{ij}	Toll charge of link a_{ij}
ω_i^n	Probability mass function of Y_i during time interval (Δ_{n-1}, Δ_n)
$u(Y_i)$	Mean of natural logarithm of Y_i
$v(Y_i)$	Variance of natural logarithm of Y_i

The following acronyms are used throughout this thesis:

ATIS	Advanced traveller information systems
AVI	Automatic vehicle identification
A-LEPP	Adaptive-least expected time path problems
A-RSPP	Adaptive-reliable shortest path problems
BPR	Bureau of Public Road
B-TDRSP	Solution algorithm for solving the backward TD-RSPP
CBD	Central business district
CDF	Cumulative distribution function
CHT	Cross harbour tunnel
CV	Coefficient of variation
DRSS	Driving route searching service
EHC	Eastern harbour crossing
FIFO	First in first out

FSD	First order stochastic dominance
F-TDRSP-A*	Multi-criteria A* algorithm for solving the forward TD-RSPP
GPS	Global positioning system
LTM	Link travel time model
ITS	Intelligent transportation systems
M-B dominance	Mean – travel time budget dominance
M-V dominance	Mean – travel time variance dominance
P-LEPP	Priori-least expected path problems
P-RSPP	Priori-reliable shortest path problems
PDF	Probability density function
RGS	Route guidance application
RSUE	Reliability-based stochastic user equilibrium
RSP	Reliable shortest path
RSPP	Reliable shortest path problems
BSPP-LCA	Label-correcting algorithm for solving bi-criteria shortest path problems
RSPP-LA*	Label-selection reliable shortest path algorithm
RSPP-NA*	Node-selection reliable shortest path algorithm
RSPP-P	Parametric approach for solving reliable shortest path problems
RTIS	Real time traveller information system
RUE	Reliability-based user equilibrium
SD-RSPP	Spatial-dependent reliable shortest path problems
SDRSP-HA*	Hierarchical multi-criteria A* algorithm for solving SD-RSPP
SD	Standard deviation
SE	Scan eligible
STD	Stochastic time-dependent
SUE	Stochastic user equilibrium
S-FIFO	Stochastic first in first out
S-LTM	Stochastic link travel time model
S-TSM	Stochastic travel speed model
TD-RSPP	Time-dependent reliable shortest path problems
TD-SPP	Time-dependent shortest path problems
TSM	Travel speed model
UE	User equilibrium
WHC	Western harbour crossing

1. Introduction

1.1. Need for the Study

Shortest path problems have been intensively studied owing to their broad applications in various science and engineering disciplines. In the transportation field, substantial attention has been given to the development of efficient shortest path algorithms for routing systems (Zhan and Noon, 1998; Fu et al., 2006). The value of routing systems is most evident when incorporating real-time traffic information generated by advanced traveller information systems (ATIS).

Most existing ATIS-based routing systems assume that link travel times are static and deterministic. However, in congested urban road networks, link travel times are highly stochastic due to the traffic demand fluctuations and capacity degradations (Lee et al., 2009). The statistical distributions of travel times vary in accordance with time-of-day due to the temporal changes of network demand and supply. For instance, the travel time from home to work during a morning peak hour (8 AM) could be significantly different from that during an off-peak hour (11 PM). In addition, link travel times are spatially correlated due to flow propagation over time and space (Chan et al., 2009). For example, a traffic accident happening at a major urban road may also cause significant travel delays on that road's upstream links.

Many empirical studies have found that travel time uncertainties have a significant impact on travellers' departure time and route choice behaviour (Abdel-Aty et al., 1995; Lam and Small, 2001; Tam et al., 2008). These empirical studies revealed that travellers indeed consider travel time uncertainties as a risk, when planning for important events. Clearly large travel time variations may cause late arrivals and the subsequent imposition of high penalties (e.g. missed flights). As a result, travellers

tend to depart from their origin early and/or choose reliable shortest path for their travel, so that they can arrive at the destination with a given on-time arrival probability, termed ‘travel time reliability’ in the literature.

Hence, such travellers’ risk-taking behaviours under travel time uncertainties are necessary inclusions in sophisticated ATIS-based routing systems to aid road users to determine optimal choices on departure time and reliable shortest path in large-scale road networks. Therefore, there is a need to investigate reliable shortest path problems in the context of ATIS-based routing systems.

In addition, great strides have been made recently in traffic assignment models (Lo et al., 2006; Shao et al., 2006a; Taylor et al., 2006; Watling, 2006; Chen et al., 2007; Chen and Zhou, 2010) for transportation planning and network vulnerability analysis. New approaches have been developed to augment and improve existing ones by explicitly modelling travellers’ heterogeneous risk-taking behaviour under demand and supply uncertainties. It is well known that the shortest path problem is an essential sub-problem of such network assignment models. However, few shortest path algorithms have been developed to capture travellers’ heterogeneous risk-taking behaviour. Therefore, there is a need for developing reliable shortest path algorithms for development of such advanced traffic assignment models.

The reliable shortest path problems in both static and time-dependent networks are significant. Although travel times are time-dependent in nature, it is valuable to investigate the reliable shortest path problems in static networks, where link travel time distributions are assumed to be stable over the period of analysis (e.g. morning peak hours). This steady-state assumption of link travel times is commonly used in traffic assignment models (Sheffi, 1985) for long-term transportation planning. With respect to ATIS-based routing systems, the use of such static travel times could be sufficient to aid travellers make pre-trip planning decisions, given its computational

simplicity. The change of travel times within the analysis period can be captured as a source of travel time variations (Sen et al., 2001). In contrast, the reliable shortest path finding in stochastic time-dependent networks is computationally demanding but can provide more accurate reliable shortest path finding results for dynamic traffic assignment models and ATIS-based routing systems.

In summary, the reliable shortest path problems are critical for developing both ATIS-based routing systems and traffic assignment models in realistic road networks. According to their different application contexts, and for sake of the presentation, the reliable shortest path problems can be broadly classified into four variants shown in Table 1.1.

Table 1.1 Classification of reliable shortest path problems

Problems	Travel time characteristics		Consider travel time correlations		Transportation applications		
	Stochastic stationary	Stochastic time - dependent	Spatial	Temporal	ATIS-based routing systems	Static traffic assignment	Dynamic traffic assignment
RSPP	√				√	√	
SD-RSPP	√		√		√	√	
TD-RSPP		√		√	√		√
ST-RSPP		√	√	√	√		√

RSPP: Reliable shortest path problem

SD-RSPP: Spatially-dependent reliable shortest path problem

TD-RSPP: Time-dependent reliable shortest path problem

ST-RSPP: Spatiotemporal-dependent reliable shortest path problem

1.2. Objectives of the Research

The aim of this research is to develop efficient solution algorithms for solving the reliable shortest path problems, and to illustrate the applicability of such algorithms

in realistic road networks. The specific associated objectives, designed to achieve the above aim, are as follows:

Objective 1: to develop an efficient solution algorithm for solving the reliable shortest path problem (RSPP) in stochastic stationary networks with independent link travel times.

Objective 2: to develop an efficient solution algorithm for solving the spatially-dependent reliable shortest path problem (SD-RSPP) in stochastic stationary networks with spatial correlated link travel times.

Objective 3: to develop an efficient solution algorithm for solving the time-dependent reliable shortest path problem (TD-RSPP) in stochastic time-dependent networks where link travel time distributions vary by time intervals throughout the day.

Objective 4: to develop efficient solution algorithms for solving spatiotemporal-dependent reliable shortest path problem (ST-RSPP) and to illustrate their applicability in a real-world ATIS-based routing system.

Objective 5: to develop an efficient solution algorithm for solving multi-class reliability-based traffic assignment problems in large-scale road networks.

Objective 6: to investigate the effects of travellers' heterogeneous risk-taking behaviour on network vulnerability analysis and to then develop an efficient solution algorithm for identifying critical links in large-scale road networks.

The inter-relationships of these research objectives are depicted in the following Fig 1.1. As shown in the figure, this research consists of three components: ATIS

(advanced traveller information systems), reliable shortest path algorithms and illustrative applications. The ATIS utilizes real-time and historical travel information to estimate and/or predict link travel time distributions in congested road networks. Stochastic link travel times generated by ATIS can be represented as either static or time-dependent.

With link travel time distributions from the ATIS, solution algorithms (Objectives 1, 2, 3) are developed for finding reliable shortest paths in road networks. These reliable shortest path algorithms are then applied to the development of ATIS-based routing systems (Objective 4).

The algorithm for solving RSPP (Objective 1) is also utilized to develop an efficient solution algorithm for solving reliability-based traffic assignment models (Objective 5). The effects of travellers' heterogeneous risk-taking behaviour on network vulnerability analysis (Objective 6) are further investigated based on research outputs of Objectives 1 and 5.

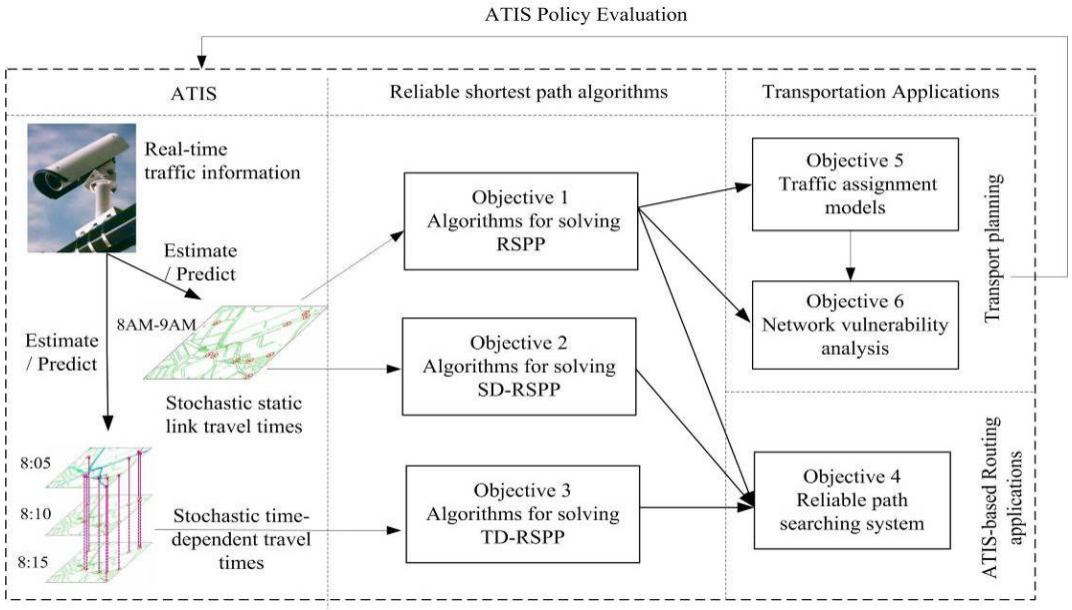


Figure 1.1 The inter-relationships of research objectives

1.3. Structure of the Thesis

The thesis comprises four basic parts. The first part (Chapters 1 and 2) gives a brief introduction and relevant literature review of travel time variability, travellers' risk-taking behaviour, stochastic shortest path problems, reliability-based traffic assignment and network vulnerability analysis.

The second part presents models and solution algorithms (Chapters 3, 4, and 5) for solving reliable shortest path problems. The third part illustrates the applicability of reliable shortest path algorithms in ATIS-based routing system (Chapter 6), reliability-based traffic assignment (Chapter 7) and network vulnerability analysis (Chapter 8). The fourth part (Chapter 9) gives a summary of this research and recommendations for further research. The relationship between these four parts in this thesis is illustrated in Fig. 1.2 as below.

Specifically, Chapter 3 describes solution algorithms for solving RSPP in stochastic stationary networks with independent link travel times. The dominance conditions of RSPP are established to enable the use of generalized dynamic programming approach to solve RSPP. Two efficient multi-criteria A* algorithms based on different labeling strategies are presented. The computational performance of proposed solution algorithm is then theoretically and computationally examined.

Chapter 4 addresses SD-RSPP in stochastic stationary networks with spatial correlated link travel times. A concept of impact area is introduced to represent travel time spatial correlations among neighboring links. A hierarchical multi-criteria A* algorithm to solve SD-RSPP on an equivalent two-level hierarchical network is proposed. A case study using traffic data from a real-world ATIS is provided to validate the proposed solution algorithms.

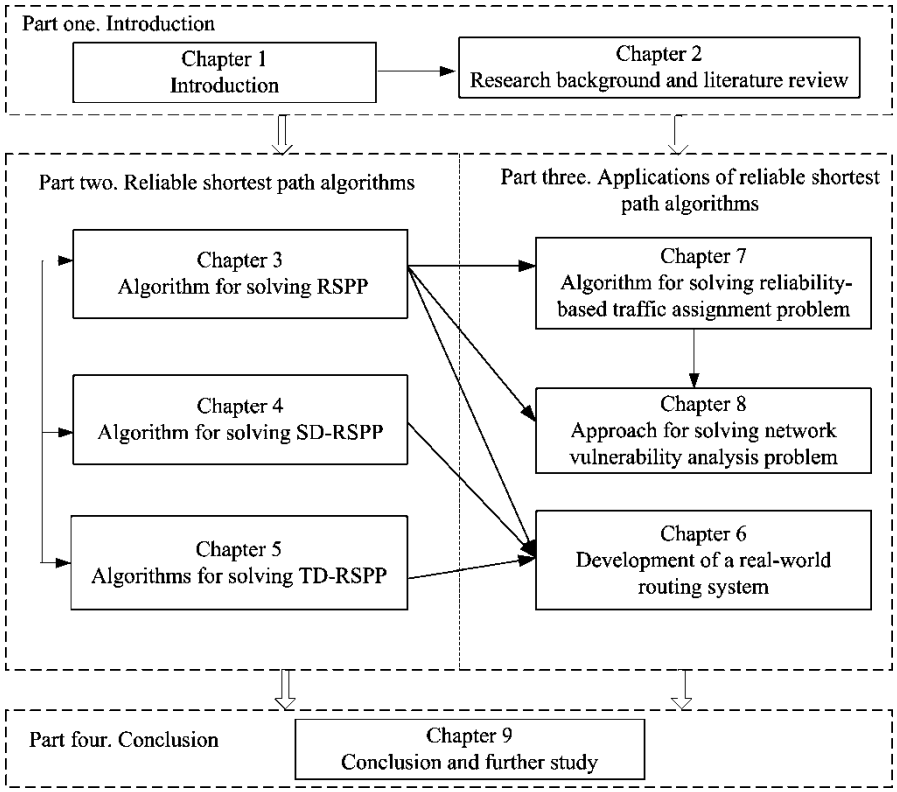


Figure 1.2 Structure of the thesis

Chapter 5 presents solution algorithms for solving TD-RSPP in stochastic time-depdent networks. Three key issues involved in TD-RSPP (a stochastic first in first out property, reversibility property and the generation of path travel time distribution) are investigated. Efficient multi-criteria A* algorithms are extended for solving TD-RSPP in two different routing scenarios.

Chapter 6 describes a real-world ATIS-based routing system. Travel time distributions generated from a real-world ATIS are incorporated. A practical reliable shortest algorithm to consider travel time spatiotemporal correlations and travellers' various routing preferences in real road networks is implememented.

In Chapter 7, a solution algorithm is proposed to solve the reliability-based traffic assignment problem. An effective reliable shortest path algorithm is developed to

determine, in one search process, reliable shortest paths from one origin to all network nodes for all user classes. Based on the proposed shortest path algorithm, a path-based traffic assignment algorithm, using a column generation method is then presented.

Chapter 8 addresses the network vulnerability analysis problem in large-scale road networks with demand uncertainty. An impact area network vulnerability analysis approach is proposed to identify critical links, the failure of which would have the most serious impact on the whole network. Case studies involving two real networks to illustrate the proposed approach are presented.

2. Backgrounds and Literature Review

This chapter introduces the research background and reviews related studies in the literature. The chapter is structured as follows. The advanced traveller information systems and associated routing systems are introduced in Section 2.1. A review of travel time variability and travellers' risk-taking behaviour is given in Section 2.2. Solution algorithms for solving stochastic shortest path problems are summarized in Section 2.3. The reliability-based traffic assignment models are reviewed in Section 2.4. The network vulnerability analysis problems are described in Section 2.5. A summary of the literature reviewed is given in Section 2.6.

2.1. Advanced Traveller Information Systems

Traffic congestion has been recognized as one of the most serious problems of modern society. Its negative effects include such as increased travel time, fuel consumption and air pollutions. For example, according to the 2009 Urban Mobility Report, traffic congestions in USA have accounted for a 100 billion loss in national productivity. Traffic congestions caused Americans to travel an additional 4.2 billion hours and to purchase an extra 2.8 billion gallons of fuel (Schrank and Lomax, 2009).

Traffic congestions can be classified into two categories: recurrent and non-recurrent congestion. Recurrent congestions are mainly due to peak hour traffic demand exceeding the available roadway capacity (Zhong, 2010). Non-recurrent congestions are typically caused by traffic incidents (Pal and Sinha, 2000), such as traffic accidents, adverse weather conditions, and man-made (e.g. terrorist attacks) and natural disasters (e.g. bridge collapses, landslides, earthquakes).

A traditional measure to remedy congestion is to build new transportation infrastructures so as to increase network capacity. This supply-side measurement is, however, politically, financially and environmentally constrained in densely urban areas (Gao and Chabini, 2006).

In recent years, to alleviate congestion, attention has increasingly been given to better uses of existing transportation infrastructures through the development of intelligent transportation systems (ITS) (Chowdhury and Sadek, 2003). With advanced information techniques, ITS enables elements within the transportation system (vehicles, roads, traffic lights, etc.) to become intelligent, leading to an efficient management of traffic demand and network supply. It has been regarded by practitioners from academic institutions and industry that ITS can significantly improve transportation network performance, including such as reduced congestion and emission**,** increased safety and traveller convenience.

Advanced traveller information systems (ATIS), as a major component of ITS, may be defined as all possible systems which provide travellers with updated traffic information to help them making better driving decisions. It is expected that informed drivers may be able to avoid problematic roads for their trips, so that a high level of efficiency, convenience and safety can be enjoyed by drivers. From the perspective of transportation authorities, ATIS can be regarded as a demand-side measurement for congestion alleviations. Researchers have found that ATIS can not only help drivers to make better route choice decisions regarding the avoidance of congested roads, but also significantly improve overall network traffic conditions (Avineri and Prashker, 2006; Ng et al., 2006; Toledo and Beinhaker, 2006; Li et al., 2010).

As shown in Fig. 2.1, ATIS typically consists of three components: data collection, data processing and ATIS-based routing systems. To collect real-time traffic data, ATIS utilizes various technological infrastructures deployed on road networks, such as loop detectors, close-circuit cameras, automatic vehicle identification (AVI)

detectors, probe vehicles equipped with GPS (global positioning system) devices. The collected real-time traffic data are then transmitted to a traffic management centre to estimate and/or predict traffic information (e.g. travel speed, travel time, detected traffic incidents) (Dion and Rakha, 2006; van Lint, 2008; Yue and Yeh, 2008; Lee et al., 2009; Vanajakshi et al., 2009). The traffic information is subsequently disseminated to road users through various media, so as to provide ATIS-based routing services.

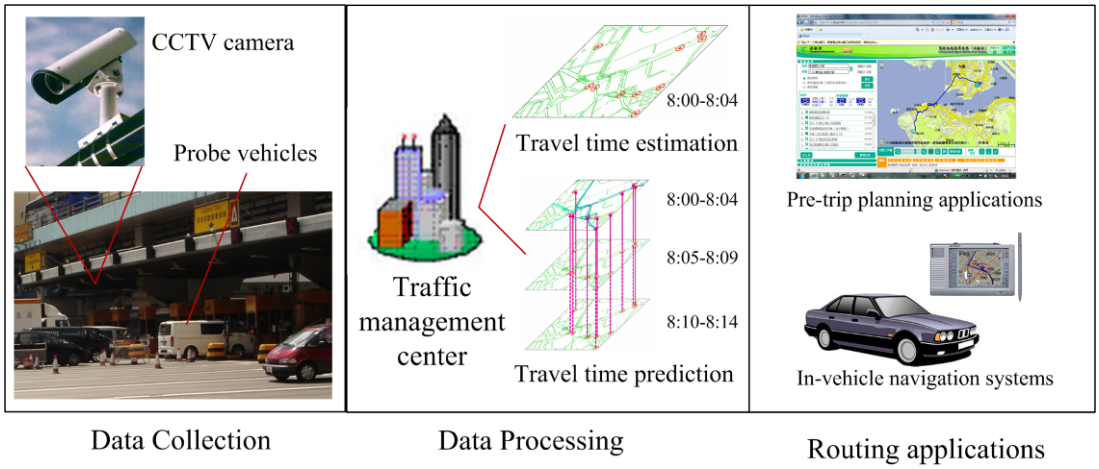


Figure 2.1 Typical framework of ATIS application

Specially, the ATIS-based routing systems can be classified as either an in-vehicle navigation system or a pre-trip planning system (Toledo and Beinhaker, 2006). The in-vehicle navigation system requires on-board microprocessors together with wireless communication devices for calculating the optimal route and update traffic information during the journey. Typically the recommended routes must be found and repeatedly updated for large-scale networks within a short time. A pre-trip planning system, however, is generally deployed to provide routing services for the public through website portals or 3-G cellular phones. On occasions, a large number of users may request the pre-trip planning service simultaneously. These requests require routes and schedules to be calculated within a reasonable time interval.

2.2. Travel Time Variability and Travellers' Risk-taking Behaviour

Fig. 2.2 shows travel times of a typical route in Hong Kong collected by AVI detectors. As shown in the figure, travel times in congested urban road networks are not constant but rather, highly stochastic.

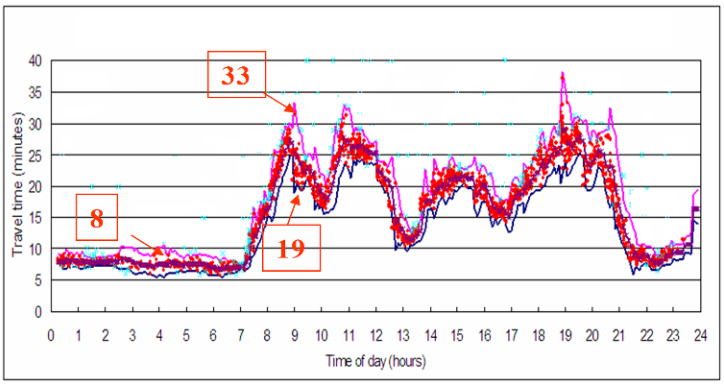


Figure 2.2 Travel time information collected on a route in Hong Kong

Travel time variability has been investigated in the literature for a long time. Early study conducted by Wardrop (1952) found that travel times follow a skewed distribution with a long 'tail' representing the few very slow vehicles. Later studies found that travel times indeed follow a skewed distribution and travel times could be modeled as a lognormal distribution (Polus, 1979; Dandy and McBean, 1984; Montgomery and May, 1987; Rakha et al., 2006). Rakha et al. (2006) pointed out that although travel time distributions do not follow normal distributions, and that the use of normal distributions appeared to reflect travel times for the most part except for some outlier observations at the right tails of travel time distributions. The normality assumption could be sufficient from a practical standpoint, given its computational simplicity (Rakha et al., 2006; Lim et al., 2008).

Travel time variability introduces uncertainties for travellers such that they do not know exactly when they will arrive at their destinations. Obviously, large travel time

variations may cause late arrivals and the subsequent imposition of high penalties for travellers when planning important events (e.g. job interview). As a result, the major concerns of travellers could be travel time reliability. Travel time reliability is defined as the on-time arrival probability that a trip can be successfully made within a desirable time interval (Bell and Iida, 1997; FHWA, 2006). Hence, travellers tend to allow extra travel time, referred to as safety margin (Knight, 1974), to achieve on-time arrival probability.

Studies of such traveller risk-taking behaviour have been well recognized in the literature. Jackson and Jucker (1981) empirically found that both mean and variance are important factors affecting travellers' route choice decisions. Abdel-Aty et al. (1995) identified that travel time reliability is either the most important or second most important factor for the commuters when choosing routes. Bates et al. (2001) also found that for most travellers a one minute reduction of travel time standard deviation is equally valued as two minutes on the actual travel time.

Travellers with heterogeneous degrees of risk-aversion are also recognized in many empirical studies (Lam and Small, 2001; Tam et al., 2008; Tilahun and Levinson, 2010), as travellers may value travel time reliability differently depending on their socio-economic characteristics and trip purposes. It was reported by Lam and Small (2001) that women and commuters with a higher income level are substantially more risk-averse to travel time variations. Based on the results of a stated-preference survey, Tilahun and Levinson (2010) pointed out that younger travellers are more likely to choose reliable routes. In the analysis of air passengers' travel behaviour, Tam et al. (2008) also found that business air passengers place a significantly higher value on both travel time and reliability than non-business air passengers.

As well as the above empirical studies, several theoretical models have been proposed to quantitatively model travellers' risk-taking behaviours. These theoretical

models include travel time budget model (Chen and Ji, 2005; Lo et al., 2006), mean-excess travel time model (Chen and Zhou, 2010) and scheduling model (Noland and Small, 1995). Such three models have some similarities and can be expressed as the same type of mathematical formulation as follows.

Chen and Ji (2005) and Lo et al. (2006) adopted the concept of travel time budget for modeling road users' risk-aversion behaviour under travel time uncertainties. Let T_u^{rs} be the path travel time between origin and destination. The travel time budget, denoted by b, can be defined by travel time reliability chance constraint at confidence level α as follows:

$$b = Min(b \mid \Pr(T_u^{rs} < b) \geq \alpha) \tag{2.1}$$

The value of α depends on travellers' socio-economic characteristics and trip purposes. The heterogeneous travellers with different degrees of risk-aversion can be modeled by different values of α ($\alpha > 0.5$, $\alpha = 0.5$ and $\alpha < 0.5$ represents risk-averse, risk-neutral and risk-seeking attitude respectively). The Eq. (2.1) can be re-written as the summation of the expected travel time and the safety margin as:

$$b = \Phi_{T_u^{rs}}^{-1}(\alpha) = E(T_u^{rs}) + Z_\alpha \sigma_u^{rs}, \quad \alpha \in (0,1) \tag{2.2}$$

where $\Phi_{T_u^{rs}}^{-1}(\alpha)$ is the inverse of cumulative distribution function (CDF) of path travel time at confidence level α; $E(T_u^{rs})$ is the expected path travel time; and $Z_\alpha \sigma_u^{rs}$ is the safety margin which is a function of travel time standard deviation σ_u^{rs}.

In a later study by Chen and Zhou (2010), the concept of travel time budget was extended to take account of both reliability and unreliability aspects of path travel time. Consequently, a new concept of mean-excess travel time was proposed as the conditional expectation of the travel time exceeding the corresponding travel time

budget:

$$\eta = E(T_u^{rs} \mid T_u^{rs} \geq b) = (\int_b^{+\infty} tf(t)dt)/(1-\alpha) = (\int_\alpha^1 \Phi_{T_u^{rs}}^{-1}(s)ds)/(1-\alpha) \tag{2.3}$$

As the inverse of CDF of path travel time, $\Phi_{T_u^{rs}}^{-1}(\cdot)$, is the monotonic increasing function with respect to confidence level, Eq. (2.3) can be written as:

$$\eta = \Phi_{T_u^{rs}}^{-1}(\beta) = E(T_u^{rs}) + Z_\beta \sigma_u^{rs}, \quad \beta \in (\alpha, 1) \tag{2.4}$$

Noland and Small (1995) proposed a scheduling model to explicitly consider travel time variability in a model of scheduling choice as:

$$E(U) = \lambda_1 E(T_u^{rs}) + \lambda_2 E(SDE) + \lambda_3 E(SDL) + \lambda_4 P_L \tag{2.5}$$

In this model, the expected utility $E(U)$ for the journey is dependent on expected travel time $E(T_u^{rs})$, expected schedule delay-early $E(SDE)$, expected schedule delay-late $E(SDL)$, and the probability of late arrival P_L. According to Fosgerau and Karlstr (2010), scheduling model in Eq. (2.5) can be expressed as

$$E(U) = \lambda_1 E(T_u^{rs}) + \sigma(\lambda_2 + \lambda_3)\int_{\lambda_3/(\lambda_2+\lambda_3)}^1 \Phi^{-1}(s)ds \tag{2.6}$$

Dividing Eq. (2.6) by λ_1, the scheduling model can be written as travel time budget model as

$$E(U) = E(T_u^{rs}) + \sigma\frac{(\lambda_2 + \lambda_3)}{\lambda_1}\int_{\lambda_3/(\lambda_2+\lambda_3)}^1 \Phi_{T_u^{rs}}^{-1}(s)ds \tag{2.7}$$

$$E(U) = \Phi^{-1}(\gamma) = E(T_u^{rs}) + Z_\gamma \sigma_u^{rs}, \quad \gamma \in (0,1) \tag{2.8}$$

As formulated, the above three models can be expressed as the same travel time budget model but with different model parameters. Hence, the travel time budget model is adopted in the reliable shortest path problems throughout this study. It should be noted that solution algorithms developed in this study can also be applied to the reliable shortest path problems based on the other two models (i.e.

mean-excess travel time model and scheduling model).

2.3. Stochastic Shortest Path Algorithms

As travel time in congested urban road networks is highly stochastic, in the literature, substantial attention has been given to stochastic shortest path algorithms for finding the optimal path in such stochastic networks. In general, stochastic shortest path algorithms can be grouped into the following two factors: (1) the en-route routing strategy (2) the inclusion of travellers' risk-taking behaviour. In relation to these two factors, the stochastic shortest path problems can be broadly divided into four categories as shown in Table 2.1.

Table 2.1 Classification of stochastic shortest path problems

		En-route routing strategies	
		Adaptive	Priori
Consider travel time reliability?	No	A-LEPP	P-LEPP
	Yes	A-RSPP	P-RSPP

A-LEPP: adaptive least expected time path problems
P-LEPP: priori least expected time path problems
A-RSPP: adaptive reliable shortest path problems
P-RSPP: priori reliable shortest path problems

The priori least expected time path problems (P-LEPP) have been intensively studied in the literature. Under P-LEPP, a priori optimal path with the least expected travel time is determined for risk-neutral travellers based on current estimated (or predicted) travel time distributions. When link travel times do not vary with time-of-day, the problem then reduces to a traditional shortest path problem in a deterministic network where the random link travel times are replaced by their expected values. In this scenario, efficient classical shortest path algorithms (Dijkstra, 1959; Hart et al., 1968; Fu et al., 2006; Chan and Lim, 2007; Ziliaskopoulos et al., 2009) can be employed to exactly solve P-LEPP.

P-LEPP is non-trivial when link travel times are stochastic time-dependent (STD). Due to the existence of temporal correlations in STD networks, the expected path travel time is non-additive and thus cannot be simply calculated by the summation of expected travel time of individual links (Fu and Rilett, 1998). Therefore, the efficient classical shortest path algorithms, built on the additive property, are no longer applicable for solving P-LEPP in STD networks. To address this difficulty, Fu and Rilett (1998) introduced an approximation method to estimate expected path travel time in STD networks and proposed a heuristic solution algorithm using the concept of *K*-shortest path algorithms (Yen, 1971). Miller-Hooks and Mahmassani (2000), on the basis of a stochastic dominance condition, proposed a non-polynomial label-correcting algorithm to find the least expected time paths from all network nodes to the origin for all possible departure times.

Another area of stochastic shortest path problems, receiving much attention in the literature, is the adaptive least expected time path problems (A-LEPP) (Hall, 1986; Polychronopoulos and Tsitsiklis, 1996; Fu, 2001). It is assumed in A-LEPP that travellers would realize actual (deterministic) travel time of a link when the arrival of that link's tail node; and travellers would then re-optimize their route choice decisions at each intermediate node, based on such realized link travel times, with the result that the expected path travel time would be minimized. Under these two assumptions, the optimal solution of A-LEPP is not a single complete path but a routing policy (Gao and Chabini, 2006).

Hall (1986) found that the routing policy generated by A-LEPP typically has a lower expected travel time than the priori path generated by P-LEPP. This is because the routing policy takes into account future availability of travel time information and future opportunities to divert to different paths (Fu, 2001). Most existing solution algorithms for solving A-LEPP used the label-correcting approach (Hall, 1986;

Polychronopoulos and Tsitsiklis, 1996; Cheung, 1998; Miller-Hooks and Mahmassani, 2000; Fu, 2001; Opasanon and Miller-Hooks, 2006). Spatial and/or temporal correlations among link travel times have also been explored by several authors (Waller and Ziliaskopoulos, 2002; Fan et al., 2005b; Gao and Chabini, 2006). However, in worst case these solution algorithms for solving A-LEPP can be computational intractable, because the optimal routing policy of A-LEPP may contain an infinite number of cycles.

Fan et al. (2005a) extended the above A-LEPP into the adaptive reliable shortest path problems (A-RSPP). Unlike A-LEPP, travellers under A-RSPP are assumed to choose the next node visit by maximizing the on-time arrival probability (or travel time reliability concept in Section 2.2) rather than by minimizing the expected path travel time. Successive approximation algorithms were proposed to solve A-RSPP (Fan et al., 2005a; Fan and Nie, 2006; Nie and Fan, 2006). These successive approximation methods can also be computationally intractable due to the existence of cyclic paths in the optimal routing policy of A-RSPP.

The priori reliable shortest path problems (P-RSPP) are another area that has received much attention in recent years. In his seminal work, Frank (1969) introduced the optimal priori path as the one which maximizes travellers' on-time arrival probability within a certain travel time budget. Chen and Ji (2005) proposed a similar concept to find the optimal path which minimizes the travel time budget required to ensure a pre-specified on-time arrival probability. Based on these two definitions, substantial solution algorithms have been proposed for solving P-RSPP, including multi-criteria label-correcting algorithms (Nie and Wu, 2009b; Wu and Nie, 2009), parametric algorithms (Nikolova, 2009), and genetic solution algorithms (Chen and Ji, 2005). These solution algorithms, however, either are pure heuristic (genetic solution algorithms) or have a non-polynomial computational complexity (multi-criteria label-correcting algorithms and parametric algorithms). In addition,

only a few studies have considered the travel time spatial and temporal correlations among network links (Nie and Wu, 2009a; Ji et al., 2011).

This study focuses on the above P-RSPP rather than A-RSPP. From a modeling point of view, solution algorithms for finding priori reliable shortest paths are an essential part of reliability-based traffic assignment models (referred to Section 2.4). From an ATIS-based routing system point of view, travellers usually make a specified route choice decision before the journey or near their origin (Abdel-Aty and Abdalla, 2004); and en-route re-routing occurs only if the travel time on the pre-planned route exceeds a certain threshold or if traffic incidents happened on the pre-planned route (Chorus et al., 2006).

2.4. Reliability-based Traffic Assignment

Traffic assignment, which models route choice behaviour of all travellers in the road networks, is one of the most essential tasks involved in urban transportation planning. Traditional traffic assignment models are mainly based on Wardrop's user equilibrium (UE) principle (Sheffi, 1985). In this UE modeling framework, travel demands and network capacities are assumed to be deterministic, and all travellers are assumed to have perfect knowledge of travel costs and thus choose their optimal paths with minimum travel costs. The travellers' perception errors of travel costs can also be incorporated into the UE modeling framework resulting in a stochastic user equilibrium (SUE) principle, in which all travellers minimize their own perceived travel costs (Dial, 1971; Sheffi, 1985; Bell, 1995).

The above deterministic assumptions in UE and SUE, however, are not held in real road networks. In real-world road networks, travel demands and network capacities are often exposed to various uncertainties. Several exogenous factors may contribute to network capacity degradations, including such as traffic incidents, traffic management and control, work zones, adverse weather conditions, man-made and

natural disasters (Chen and Zhou, 2010). In addition, travel demands are varying due to endogenous factors, such as temporal factor (time-of-day and day-of-week), travellers' characteristics and special events.

As indicated in Section 2.2, the random demand fluctuations and network capacity degradations result in travel time uncertainty in congested urban road networks. In this case, travellers consider travel time uncertainty as a risk and reveal various risk-taking attitudes based on their own socio-economic characteristics and trip purposes. Therefore, the UE (or SUE) traffic assignment models, built on deterministic assumptions, cannot fully capture travellers' risk-taking route choice behaviour in congested urban road networks.

To better represent travellers' route choice behaviour, Lo et al. (2006) adopted the travel time budget model (described in Section 2.2) in traffic assignment problems. Analogous with the Wardrop's user equilibrium (UE) principle (Sheffi, 1985), a notion of reliability-based user equilibrium (RUE) was introduced. In such RUE condition, all travellers in the network will choose the reliable shortest path so as to minimize their travel time budget instead of expected travel time. The heterogeneous users with different degrees of risk-aversion are modeled by assuming a discrete set of reliability thresholds for several distinct user classes. Subsequently, Watling (2006) and Chen and Zhou (2010) also developed similar RUE models based on the scheduling model and the mean-excess travel time model.

Following this RUE modeling framework, Lo et al. (2006) proposed a model considering the link capacity degradation subject to traffic incidents. Shao et al. (2006a) presented a demand driven RUE model to consider the effects of daily demand fluctuations. As an extension of these two studies, research studies were conducted to model travellers' route choice behaviour under travel time variability due to both demand fluctuation and link capacity degradation (Siu and Lo, 2008; Zhou and Chen, 2008; Chen and Zhou, 2010). Subsequently, reliability-based stochastic user equilibrium (RSUE) models were further developed to take into account travellers' perception errors under various conditions (Shao et al., 2006b; Lam et al., 2008; Chen and Zhou, 2009; Connors and Sumalee, 2009).

Although RUE traffic assignment models are significant, few efficient solution algorithms are available for solving the RUE problems in large-scale road networks. This is mainly due to the lack of efficient solution algorithms for solving reliable shortest path sub-problems. Most existing algorithms for solving RUE problems hedge this difficulty by enumerating paths or by defining a fixed path set (Lo et al., 2006; Shao et al., 2006a; Siu and Lo, 2008). It should be noted that the path enumeration is time consuming for a large-scale problem and as such is appropriate only for small-size networks.

2.5. Network Vulnerability Analysis

The investigation of network vulnerability to various disruptive incidents has received much attention in recent years, due to widespread man-made and natural disasters (e.g. 9/11 in New York City and Japan's earthquake in 2011). In the literature, the vulnerability of the road network is defined as a susceptibility to incidents that can result in considerable reductions in road network performance (Berdica, 2002). Based on this definition, network vulnerability analysis encompasses three components: traffic incidents, traveller behavioural responses to incidents, and a vulnerability index to access the consequences of incidents.

A road network may suffer two different types of traffic incidents: 1. Minor events such as vehicle breakdown and slight accidents which can occur quite frequently, and during which the network is not severely damaged. 2. Major events (such as bridge collapse, flooding, landslide and serious accident) which could seriously damage network infrastructures. Recognized is the low probability of such latter occurrences, yet in the realm of network vulnerability analyses, these major events may be more important considerations than minor events. As discussed by Taylor (2008), even if the probabilities of the major events are low, the adverse consequences of major events on particular network infrastructures could be sufficiently large to indicate a

major problem that warrants remedial actions.

Major network events, such as those indicated above, may cause different levels of disruption. During a major event, one or more network links may be involved, causing the affected links to be partially or completely closed. The combinations of different disruptive scenarios are extreme large. A common theme to most network vulnerability analysis approaches, however, is the consideration of single link completed closure (Jenelius et al., 2006; Taylor et al., 2006; Chen et al., 2007).

The impacts of traffic incidents strongly depend on traveller behavioural responses. Empirical studies have indicated that travellers tend to have time-related reactions to traffic incidents (Chang and Nojima, 2001; Hunt et al., 2002; Danczyk and Liu, 2009). Fig. 2.3 plots a typical traffic flow pattern in an affected area to illustrate the time-related reactions. During the immediate period after a disruptive incident, an instability interval in traffic patterns occurs, such as a noticeable reduction in traffic flows in the affected area immediately after the incident. A period of adjustment follows. Travellers' behavioural responses such as changing route, switching mode, changing destination, and cancelling or postponing trips, during this period are evident.

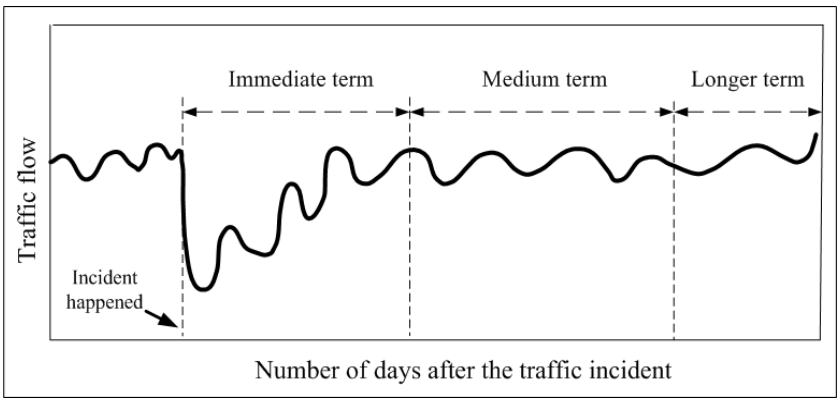


Figure 2.3 A typical traffic flow pattern after the traffic incident

After the period of adjustment, a new relatively stable "equilibrium" travel pattern may emerge. In the medium term, the network travel demand pattern adapts to the new conditions, and the traffic conditions become stable and more predictable. The

travel demand is still subject to normal day-to-day and seasonal demand variations. In the longer term, other factors such as changes in transportation policies and economic growth start to influence the traffic condition after the medium term period.

In the literature, various vulnerability indices have been proposed to evaluate the consequences of link closures. The characteristics of these indices are briefly discussed and summarized in Table 2.2. For the immediate term, Kurauchi et al. (2009) evaluated network vulnerability by considering the number of distinct paths connecting each origin-destination pair. For the medium term, Jenelius et al. (2006) used the increase of the generalized cost, weighted by travel demand, as a link closure vulnerability measure. Taylor et al. (2006) adopted Hansen for assessing the vulnerability of the national road system of Australia, and the ARIA (Accessibility/Remoteness Index of Australia) index for the rural or remote area of Australia. Chen et al. (2007) introduced the utility-based accessibility index to take account of travellers' behavioural responses to the link closure.

Table 2.2 The characteristics of road network vulnerability indices

Source	Network type	Network performance measure	Time period
Kurauchi et al. (2009)	Urban / Rural	Network connectivity	Immediate term
Taylor et al. (2006)	Urban	Hansen integral accessibility	Medium term
Taylor et al. (2006)	Rural	Accessibility / Remoteness Index of Australia	Medium term
Jenelius et al. (2006)	Urban / Rural	Generalized cost weighted by travel demand	Medium term
Chen et al. (2007)	Urban / Rural	Behavioural-utility-based accessibility	Medium term

A key issue in the vulnerability analysis is to identify the critical infrastructures (links/nodes) of a network, where the failure of those infrastructures would have the most serious impacts on the whole network. A common technique for identifying critical links is the full network scan approach (Jenelius et al., 2006; Taylor et al., 2006). In this approach, each link is iteratively removed from the network and the

consequences of its closure are measured using a form of traffic assignment (e.g. RUE traffic assignment in Section 2.4). The critical links are then identified by evaluating all possible link closures. This approach can be computationally burdensome and may not be viable for identifying critical links in large-scale networks, since traffic assignment itself is computational demand in large-scale networks.

2.6. Summary

Link travel times in congested urban road networks are highly stochastic due to random demand fluctuations and capacity degradations. Many empirical studies found that travellers indeed consider travel time uncertainty as a risk for their trips and reveal heterogeneous degrees of risk-aversion based on their socio-economic characteristics and trip purposes. Three behavioural models (namely travel time budget model, mean-excess travel time model and scheduling model) have been proposed to quantitatively model travellers' risk-taking behaviour. It was found that these three behavioural models can be formulated as the same type of mathematical expression. In this case, reliable shortest path problems based on such behavioural models can be solved by the same solution algorithms.

Two variants of the reliable shortest path problems, using different en-route routing strategies have been investigated in the literature. The priori reliable shortest path algorithms are to determine a complete reliable shortest path according to priori known travel time distributions. The adaptive reliable shortest path algorithms are to find an optimal routing policy that help travellers making route choice decisions at each intermediate node based on realized actual traffic condition.

The priori reliable shortest path problems are more practical than the adaptive variant. The priori reliable shortest path problems are an essential sub-problem of the reliability-based traffic assignment problems; and are suitable for the development of

ATIS-based routing systems.

A key issue in the vulnerability analysis is to identify the critical infrastructures of a network, where the failure of the above infrastructures would have the most serious impacts on the whole network. A common technique for identifying critical links is the full network scan approach. In this approach, each link is iteratively removed from the network and the consequences of its closure are measured using a reliability-based traffic assignment.

Through the review of reliable shortest path problems, it was noted that efficient reliable shortest path algorithm to support real-time operations of ATIS-based routing systems is yet to be developed. In addition, travel time spatial and temporal correlations are not explicitly considered in the reliable shortest path problems. With respect to reliability traffic assignment problems, most existing solution algorithms employ a path enumeration technique and thus cannot be used for solving the large-scale reliability-based traffic assignment problems. In connection with network vulnerability analysis, a traditional network scan approach is computational intractable for identifying the critical links in large-scale real transportation networks.

Based on this previous related work, efficient reliable shortest path algorithms for supporting the real-time operations of ATIS-based routing systems are proposed in Chapter 3. An extension of proposed reliable shortest path algorithms to explicitly consider the travel time spatial and temporal correlations is presented in Chapters 4 and 5. Based on the developed reliable shortest path algorithms, a real-world ATIS-based routing system is described in Chapter 6. An incorporation of reliable shortest path algorithms into the large-scale reliable-based traffic assignment problems and network vulnerability analysis problems are discussed in Chapters 7 and 8.

3. Reliable Shortest Path Problems

The problem of finding reliable shortest paths in stochastic networks with independent normal distributed link travel times is investigated and described in this chapter. It is shown that this reliable shortest path problem (RSPP) is non-additive and thus cannot be solved by classical shortest path algorithms (e.g. Dijkstra's algorithm). In this chapter, the non-additive difficulty of RSPP is tackled using a multiple-criteria shortest path approach. Several dominance conditions of RSPP are established to enable the use of generalized dynamic programming approaches to solve RSPP. Two new multi-criteria A* algorithms are proposed to solve RSPP for risk-averse, risk-neural and risk-seeking scenarios, exactly. Computational results using large scale networks demonstrated that the proposed multi-criteria A* algorithms have a significant computational advantage over existing methods. The extension of above proposed algorithms to lognormal distributions is also discussed in this chapter.

This chapter is organized as follows. Previous studies relating to RSPP are briefly reviewed in Section 3.1. The definition of RSPP in relation to stochastic networks, together with the model formulation is presented in Section 3.2. The dominance conditions of RSPP are introduced in Section 3.3. Solution algorithms to solve RSPP are presented in Section 3.4. The computational performance of the proposed solution algorithms is reported in Section 3.5. The extension of proposed solution algorithms to lognormal distributions are discussed in Section 3.6. Finally, a summary of this chapter is given in Section 3.7.

3.1. Background

The problem of finding reliable shortest paths in stochastic networks has been intensively studied. Frank (1969) introduced the concept of finding the most reliable path which maximizes the probability of travellers arriving at their destination within

a given travel time budget. Chen and Ji (2005) proposed a similar concept to find the alpha-shortest path which minimizes the travel time budget required to ensure a pre-specified on-time arrival probability. Based on these two definitions, substantial solution algorithms have been proposed in the literature to solve RSPP in stochastic networks.

In the literature, focus is on RSPP solution algorithms based on the definition of most reliable paths. Mirchandani (1976) presented a recursive algorithm to solve the discrete version of Frank's problem. However, the algorithm requires path enumeration and thus cannot be applied to large-scale problems. Shao et al. (2004) proposed a heuristic method for solving this problem, based on the relationship between the link travel time mean and standard deviation. Nonetheless, this method cannot provide a solution for risk-seeking travellers, whose travel time budget is less than the least expected travel time between origin-destination (O-D) nodes. Nikolova (2006) and Lim et al. (2008) proposed parametric approaches to determine the most reliable shortest path. However, such parametric approaches also cannot determine the solution for risk-seeking travellers.

Nie and Wu (2009b) proposed a label-correcting algorithm to find the most reliable path by generating all non-dominated paths between O-D nodes under first-order stochastic dominance. The proposed algorithm, notwithstanding, has a non-deterministic polynomial complexity, since the number of non-dominated paths grows exponentially with network size. To reduce the number of generated non-dominated paths, an approximate approach was proposed. Nevertheless, the approximation may miss the optimal paths completely, and there is no precise notion of how good that approximation is.

A few efficient solution algorithms have been proposed to solve the path finding problem based on the definition of alpha-shortest paths. Chen and Ji (2005) presented a simulation-based genetic algorithm for finding both the alpha-shortest and the most reliable paths. It should be noted that the simulation-based methods are computationally expensive and the precision of results depends on the number of

simulations. Nikolova (2010) proposed a parametric approach to determine the alpha-shortest path. This algorithm, however, cannot determine a solution for risk-seeking travellers.

The alpha-shortest path definition is adopted in this chapter to address RSPP in the context of online pre-trip planning applications. The definition of alpha-shortest path is shown to be equivalent to that of Frank (1969), but it can better reflect travellers' route choice processes in pre-trip planning applications.

3.2. Problem Definition

Let $\mathrm{G}=(\mathrm{N},\mathrm{A})$ be a directed network, where N and A are the sets of nodes and links respectively. Suppose that the nodes $r \in N$ and $s \in N$ represent the O-D nodes of G. Each node i has a set of successor nodes $SCS(i)=\{j: a_{ij} \in A\}$ and a set of predecessor nodes $PDS(i)=\{k: a_{wi} \in A\}$. Assume that each link a_{ij} has a random travel time T_{ij} with a given probability density function (PDF). The mean and standard deviation (SD) of link travel time are denoted by t_{ij} and σ_{ij} respectively. Let $P^{rs}=\{p_1^{rs},...,p_n^{rs}\}$ be a set of paths from the origin r to the destination s. Let $\delta_{ij}^{rs,u}$ be the link-path incidence relationship, where $\delta_{ij}^{rs,u}=1$ means that the link a_{ij} is on the path p_u^{rs}, and $\delta_{ij}^{rs,u}=0$, otherwise. The path travel time, denoted by T_u^{rs}, is the sum of related link travel times along the path as

$$T_u^{rs}=\sum_{a_{ij}\in A} T_{ij}\delta_{ij}^{rs,u} \tag{3.1}$$

Obviously, the path travel time T_u^{rs} is also a random variable whose distribution is the joint PDF of all links along the path. The mean and SD of path travel time are denoted by t_u^{rs} and σ_u^{rs} respectively. Let $\Phi_{T_u^{rs}}(\cdot)$ be the cumulative distribution function (CDF) of path travel time and $\Phi_{T_u^{rs}}^{-1}(\cdot)$ be the inverse CDF of path travel time.

The reliable shortest path defined by Frank (1969) can be expressed with respect to the pre-specified travel time budget b as below:

Definition 3.1 (Most reliable path). Given a travel time budget b, a path $p_u^{rs} \in P^{rs}$ is the most reliable path if $\Phi_{T_u^{rs}}(b) > \Phi_{T_v^{rs}}(b)$ for any other path $p_v^{rs} \in P^{rs}$.

As an alternative, the alpha-shortest path can be defined with respect to a pre-specified on-time arrival probability α (Chen and Ji, 2005) as follows:

Definition 3.2 (Alpha-shortest path). Given an on-time arrival probability α, a path $p_u^{rs} \in P^{rs}$ is the alpha-shortest path if $\Phi_{T_u^{rs}}^{-1}(\alpha) < \Phi_{T_v^{rs}}^{-1}(\alpha)$ for any other path $p_v^{rs} \in P^{rs}$.

Travellers' different risk attitudes towards travel time uncertainty according to on-time arrival probability can be defined as follows:

- If $\alpha > 0.5$ (or $\Phi_{T_u^{rs}}(b) > 0.5$) then the traveller is 'risk-averse for on-time arrival';
- If $\alpha = 0.5$ (or $\Phi_{T_u^{rs}}(b) = 0.5$) the traveller is then 'risk-neutral for on-time arrival';
- If $\alpha < 0.5$ (or $\Phi_{T_u^{rs}}(b) < 0.5$) the traveller is then 'risk-seeking for on-time arrival'.

Even though the reliable shortest path can be determined using either Definition 3.1 or 3.2, it should be noted that these two definitions are different to some extent. To find the most reliable path, an underlying assumption is that travellers can determine their travel time budgets prior to their trips. However, the travel time budget depends largely on the distance between O-D nodes and the congestion level in the network. The travellers may be thus unable to define an appropriate travel time budget prior to their trips. Definition 3.2, however, requires the travellers to express their desired risk attitudes toward travel time uncertainty. It is postulated in this research that in

the context of pre-trip planning applications, travellers can identify their desired risk attitudes based on trip purposes rather than defining an appropriate travel time budget in the stochastic network. Hence, the alpha-shortest path definition is adopted for determining the optimal path in stochastic networks.

In this chapter, it is assumed that link travel times follow normal distributions and all link travel times are statistically independent. These two assumptions are commonly used in studies of stochastic shortest path problems (Wijeratne et al., 1993; Shao et al., 2004; Chang et al., 2005). As indicated in Chapter 2, the normality assumption could be sufficient from a practical standpoint, given its computational simplicity. Under the normality assumption, RSPP can be formally defined as follows:

$$Min \quad \Phi_{T_u^{rs}}^{-1}(\alpha) = \sum_{a_{ij} \in A} t_{ij} \delta_{ij}^{rs,u} + Z_\alpha * \sqrt{\sum_{a_{ij} \in A} \sigma_{ij}^2 \delta_{ij}^{rs,u}} \tag{3.2}$$

Subject to

$$\sum_{j \in SCS(i)} \delta_{ij}^{rs,u} - \sum_{w \in PDS(i)} \delta_{wi}^{rs,u} = \begin{cases} 1 & \forall\, i = r \\ 0, & \forall\, i \neq r; i \neq s, \\ -1 & \forall\, i = s \end{cases} \tag{3.3}$$

$$\delta_{ij}^{rs,u} \in \{0, 1\}, \quad \forall a_{ij} \in A \tag{3.4}$$

Eq. (3.2) is the travel time budget that travellers want to minimize. Eq. (3.3) ensures that the links on the reliable shortest path are feasible. Eq. (3.4) is concerned with the link-path incidence variables which are binary in nature.

RSPP can be regarded as one of the non-additive shortest path problems, since the path cost ($\Phi_{T_u^{rs}}^{-1}(\alpha)$) cannot be calculated as the sum of related links costs. This leads to a violation of Bellman's Principle of Optimality (Bellman, 1958), which states that a sub-path between any pair of nodes on the shortest path is itself, the shortest path. An illustration of this non-additive property can be found in Fig. 3.1, where $\oplus$ is a path concatenation operator (e.g. $p_1^{13} = a_{12} \oplus a_{23}$ means that p_1^{13} goes through a_{12} and a_{23}).

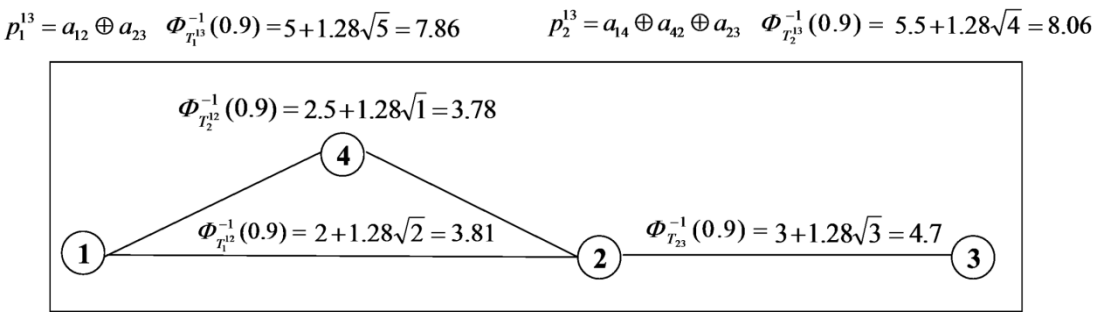


Figure 3.1 An illustration of non-additive property of RSPP

As shown in the figure, consider a risk-averse traveller ($\alpha = 0.9, Z_a = 1.28$) the cost of path p_1^{13} is not equal to the aggregate cost of links a_{12} and a_{23} ($\Phi_{T_1^{13}}^{-1}(0.9) = 7.86 < \Phi_{T_1^{12}}^{-1}(0.9) + \Phi_{T_{23}}^{-1}(0.9) = 8.51$). Note that the optimal path between nodes 1 and 3 uses the path p_1^{13} going through links a_{12} and a_{23}. Based on Bellman's Principle of Optimality, one may expect that the optimal path from node 1 to 2 should be link a_{12}, but it actually goes through sub-path $p_2^{12} = a_{14} \oplus a_{42}$ ($\Phi_{T_2^{12}}^{-1}(0.9) = 3.78 < \Phi_{T_1^{12}}^{-1}(0.9) = 3.81$). The violation of Bellman's Principle of Optimality disallows the application of classical dynamic programming approaches (e.g Dijkstra's algorithm) to solve RSPP.

In the following sections, multi-criteria shortest path approach is proposed to solve RSPP.

3.3. Multi-criteria Shortest Path Approach

In this section, RSPP is formulated as a multi-criteria shortest path problem and the principle of Bellman's optimality is extended. The multi-criteria shortest path problem typically relies on a number of specific dominance conditions to determine dominated paths. When considering two paths $p_u^{ri} \neq p_v^{ri} \in P^{ri}$ from origin r to a node i, the following dominated path definitions for RSPP exist:

Let $p_u^{rw} = p_u^{ri} \oplus p^{iw}$ and $p_v^{rw} = p_v^{ri} \oplus p^{iw}$ be two paths from origin r to node w

going through the same sub-path p^{iw}. T_u^{rw} and T_v^{rw} denote travel time for these two paths respectively.

Definition 3.3. A path $p_u^{ri} \in P^{ri}$ dominates another path $p_v^{ri} \in P^{ri}$ (denoted by $p_u^{ri} \succ p_v^{ri}$), if and only if $\Phi_{T_u^{rw}}^{-1}(\alpha) < \Phi_{T_v^{rw}}^{-1}(\alpha)$ for any path $p^{iw} \in P^{iw}$ and any node $w \in N$.

Definition 3.4. A path $p_u^{ri} \in P^{ri}$ is a non-dominated path, if and only if p_u^{ri} is not dominated by any path $p_v^{ri} \in P^{ri}$.

Based on Definitions 3.3 and 3.4, Bellman's Principle of Optimality is extended as follows.

Theorem 3.1. A sub-path of any non-dominated path must be a non-dominated path itself.

Proof. Suppose $p_v^{ri} \in P^{ri}$ is a sub-path of a non-dominant path $p_v^{rw} = p_v^{ri} \oplus p_v^{iw} \in P^{rw}$, and it is dominated by another path $p_u^{ri} \in P^{ri}$. According to Definition 3.3, we have $\Phi_{T_u^{rs}}^{-1}(\alpha) < \Phi_{T_v^{rs}}^{-1}(\alpha)$ for any path $p^{is} \in P^{is}$. Therefore, there exists at least one path $p_u^{rw} = p_u^{ri} \oplus p_v^{iw} \in P^{rw}$ satisfying $\Phi_{T_u^{rs}}^{-1}(\alpha) < \Phi_{T_v^{rs}}^{-1}(\alpha)$ for any $p^{ws} \in P^{ws}$, since any $p_v^{iw} \oplus p^{ws} \in P^{is}$. Thus, $p_u^{ri} \succ p_v^{ri}$. This contradicts the fact that p_v^{ri} is a non-dominant path. □

The principle of optimality in Theorem 3.1 can be utilized to search for the reliable shortest path using a generalized dynamic programming method. Compared with the conventional method, several non-dominated paths may have to be stored at each node in RSPP. The dominated paths at each node, however, can be discarded without further consideration since they cannot be parts of the reliable shortest path between O-D nodes.

To determine and discard dominated paths in the stochastic network, Miller-Hooks

and Mahmassani (2003) suggested a first-order stochastic dominant (FSD) condition to determine the dominated paths under the most reliable path definition. As shown in Fig. 3.2, the FSD condition can also be expressed under the alpha-shortest path definition as follows:

Proposition 3.1 Given two paths $p_u^{ri} \neq p_v^{ri} \in P^{ri}$, $p_u^{ri} \succ p_v^{ri}$ if p_u^{ri} and p_v^{ri} satisfy $\Phi_{T_u^{ri}}^{-1}(\lambda) < \Phi_{T_v^{ri}}^{-1}(\lambda)$ for any confidence level $0 < \lambda < 1$.

Proof. See Appendix A.

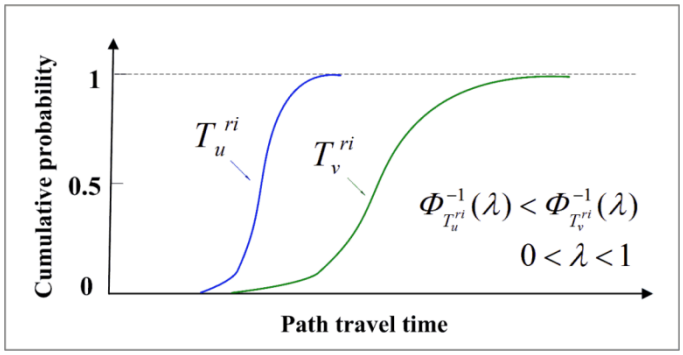


Figure 3.2 First-order stochastic dominance

From the objective function, RSPP can be formulated as a bi-criterion shortest path problem with respect to the mean and variance of the path travel time, and mean-variance (M-V) dominance defined as follows:

Proposition 3.2 (M-V dominance). Given an on-time arrival probability α and two paths $p_u^{ri} \neq p_v^{ri} \in P^{ri}$, $p_u^{ri} \succ p_v^{ri}$ if p_u^{ri} and p_v^{ri} satisfy either

(i) $t_u^{ri} \leq t_v^{ri}$ and $Z_\alpha \sigma_u^{ri} < Z_\alpha \sigma_v^{ri}$ or

(ii) $t_u^{ri} < t_v^{ri}$ and $Z_\alpha \sigma_u^{ri} \leq Z_\alpha \sigma_v^{ri}$

Proof. see Appendix A. □

Fig. 3.3 graphically illustrates the M-V dominance. As shown in this figure, the M-V dominance is stronger than FSD. Using this M-V dominance condition, risk-averse travellers can discard dominated paths satisfying $\Phi_{T_u^{ri}}^{-1}(\lambda) < \Phi_{T_v^{ri}}^{-1}(\lambda)$ for any

confidence level $0.5 \le \lambda < 1$; and risk-seeking travellers can determine dominated paths satisfying $\Phi^{-1}_{T_u^{ri}}(\lambda) < \Phi^{-1}_{T_v^{ri}}(\lambda)$ for any confidence level $0 < \lambda \le 0.5$.

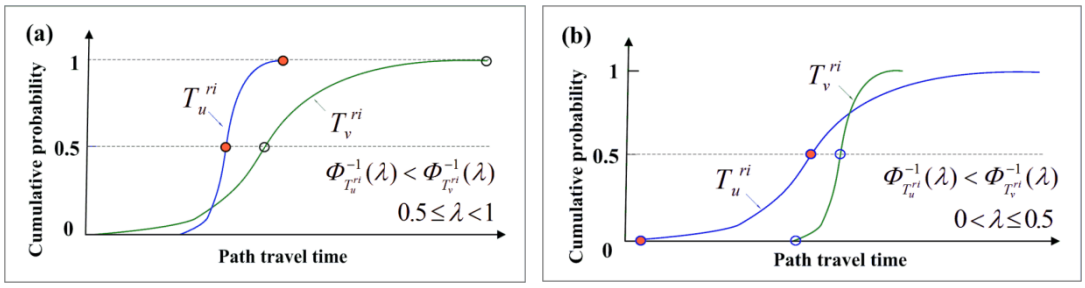


Figure 3.3 Mean-variance dominance (a) risk-averse (b) risk-seeking

In addition to the above two dominance conditions, (Hutson and Shier, 2009) introduced mean-travel time budget (M-B) dominance to determine dominated paths for only risk-averse scenarios. It can be proved that the M-B dominance can also be applied to risk-seeking and risk-neutral scenarios as follows.

Proposition 3.3 (M-B dominance). Given an on-time arrival probability α and two paths $p_u^{ri} \ne p_v^{ri} \in P^{ri}$, $p_u^{ri} \succ p_v^{ri}$ if p_u^{ri} and p_v^{ri} satisfy $t_u^{ri} \le t_v^{ri}$ and $\Phi^{-1}_{T_u^{ri}}(\alpha) < \Phi^{-1}_{T_v^{ri}}(\alpha)$.

Proof. see Appendix A. □

As shown in Fig. 3.4, the M-B dominance can help identify potential dominated paths which may not be identified under the M-V dominance condition. Depending on the criterion, risk-averse travellers can determine and discard dominated paths that satisfy $\Phi^{-1}_{T_u^{ri}}(\lambda) < \Phi^{-1}_{T_v^{ri}}(\lambda)$ for any confidence level $0.5 \le \lambda < \alpha$. For risk-seeking travellers, the dominated paths satisfying $\Phi^{-1}_{T_u^{ri}}(\lambda) < \Phi^{-1}_{T_v^{ri}}(\lambda)$ for any confidence level $0.5 \le \lambda < \alpha$ could be determined and discarded.

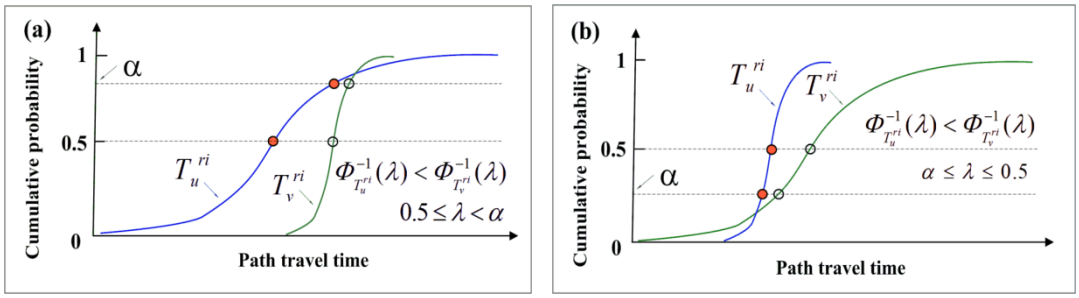


Figure 3.4 Mean-travel time budget dominance (a) risk-averse (b) risk-seeking

Based on the above dominance conditions, it can be observed that the alpha-shortest path is more suitable than the most reliable path for developing efficient solution algorithms for pre-trip planning applications. Using the definition of the most reliable path, one cannot determine on-time arrival probability until the destination node has been reached. Thus, all non-dominated paths under the FSD condition have to be kept in the reliable shortest path search process. The definition of alpha-shortest path, however, has an explicit on-time arrival probability reflecting travellers' risk attitude toward travel time uncertainty. It can lead to more strict dominance conditions (M-V dominance and M-B dominance) reducing the number of generated non-dominated paths thereby improving the efficiency of the search process.

The dominance conditions, based on the use of the alpha-shortest path definition, can also be established for two sets of paths. Let $P^{ri} = \{p_1^{ri}, ..., p_n^{ri}\}$ and $\hat{P}^{ri} = \{\hat{p}_1^{ri}, ..., \hat{p}_m^{ri}\}$ be two sets of non-dominated paths from origin r to node i. To determine their dominant relationships, a straightforward approach is to use established dominance conditions between two paths to check whether any $p_u^{ri} \in P^{ri}$ dominates any $\hat{p}_v^{ri} \in \hat{P}^{ri}$, and vice versa. By arranging P^{ri} and $\hat{P}^{ri}$ as ordered sets, the following dominance conditions can be used to determine and discard a set of dominated paths in a more efficient way.

Proposition 3.4. Let P^{ri} be a set of non-dominated paths ordered by increasing value of t_u^{ri}. If a path $p_u^{ri} \in P^{ri}$ is ordered in front of another path $p_v^{ri} \in P^{ri}$ $(u < v)$ then

(i) $\Phi^{-1}_{T_u^{ri}}(\lambda) \ge \Phi^{-1}_{T_v^{ri}}(\lambda)$ and

(ii) $Z_\alpha \sigma_u^{ri} \geq Z_\alpha \sigma_v^{ri}$

Proof. It can be easily followed by Propositions 3.2 and 3.3. □

Given two sets of non-dominated paths P^{ri} and $\hat{P}^{ri}$, if each path of $\hat{P}^{ri}$ is dominated by any path of P^{ri}, $\hat{P}^{ri}$ is then dominated by P^{ri}, denoted as $P^{ri} \succ \hat{P}^{ri}$. In such case, all paths of $\hat{P}^{ri}$ can be discarded without further consideration in the search process.

Proposition 3.5. Given two sets of non-dominated paths $P^{ri} = \{p_1^{ri},..., p_n^{ri}\}$ and $\hat{P}^{ri} = \{\hat{p}_1^{ri},..., \hat{p}_m^{ri}\}$ ordered by increasing value of t_u^{ri}, then $P^{ri} \succ \hat{P}^{ri}$ if either

(i) $t_n^{ri} \leq \hat{t}_1^{ri}$ and $\Phi_{T_n^{ri}}^{-1}(\alpha) < \Phi_{\hat{T}_m^{ri}}^{-1}(\alpha)$ or

(ii) $t_1^{ri} \leq \hat{t}_1^{ri}$ and $\Phi_{T_1^{ri}}^{-1}(\alpha) < \Phi_{\hat{T}_m^{ri}}^{-1}(\alpha)$

Proof. (i) If $t_n^{ri} \leq \hat{t}_1^{ri}$, then $t_n^{ri} \leq \hat{t}_1^{ri} \leq ... \leq \hat{t}_m^{ri}$. According to Proposition 3.4(i), if $\Phi_{T_n^{ri}}^{-1}(\alpha) < \Phi_{\hat{T}_m^{ri}}^{-1}(\alpha)$, then $\Phi_{T_n^{ri}}^{-1}(\alpha) < \Phi_{\hat{T}_m^{ri}}^{-1}(\alpha) \leq ... \leq \Phi_{\hat{T}_1^{ri}}^{-1}(\alpha)$. Therefore, $p_n^{ri} \in P^{ri}$ dominates every path $\hat{p}_v^{ri} \in \hat{P}^{ri}$ according to Proposition 3.3.

(ii) The proof is similar to (i). □

Inspired by Skriver and Andersen (2000), the following proposition is introduced to discard all paths of $\hat{P}^{ri}$ before the path extension operations, $\hat{P}^{rj} = \hat{P}^{ri} \oplus a_{ij}$, are carried out.

Proposition 3.6. Given two ordered sets of non-dominated paths $P^{rj} = \{p_1^{rj},..., p_n^{rj}\}$ and $\hat{P}^{ri} = \{\hat{p}_1^{ri},..., \hat{p}_m^{ri}\}$, then $P^{rj} \succ \hat{P}^{rj} = \hat{P}^{ri} \oplus a_{ij}$ if either

(i) $t_n^{rj} \leq \hat{t}_1^{ri} + t_{ij}$ and $Z_\alpha(\sigma_n^{rj})^2 < Z_\alpha(\hat{\sigma}_m^{ri})^2 + Z_\alpha(\sigma_{ij})^2$ or

(ii) $t_1^{rj} < \hat{t}_1^{ri} + t_{ij}$ and $Z_\alpha(\sigma_1^{rj})^2 \leq Z_\alpha(\hat{\sigma}_m^{ri})^2 + Z_\alpha(\sigma_{ij})^2$

Proof. (i) If $t_n^{rj} \leq \hat{t}_1^{ri} + t_{ij}$, then $t_n^{rj} \leq \hat{t}_1^{ri} + t_{ij} \leq ... \leq \hat{t}_m^{ri} + t_{ij}$. According to Proposition 3.4(ii), if $Z_\alpha(\sigma_n^{rj})^2 < Z_\alpha(\hat{\sigma}_m^{ri})^2 + Z_\alpha(\sigma_{ij})^2$, then

$Z_\alpha(\sigma_n^{rj})^2 < Z_\alpha(\hat{\sigma}_m^{ri})^2 + Z_\alpha(\sigma_{ij})^2 \le ... \le Z_\alpha(\hat{\sigma}_1^{ri})^2 + Z_\alpha(\sigma_{ij})^2$. Therefore, $p_n^{ri} \in P^{ri}$ dominates every path $\hat{p}_i^{ri} \oplus a_{ij} \in \hat{P}^{rj}$ according to Proposition 3.2.

(ii) The proof is similar to (i). □

The non-dominated paths of RSPP also have the following properties:

Proposition 3.7. Given two paths p_u^{ri} and $p_u^{rj} = p_u^{ri} \oplus a_{ij}$, the relationship $\Phi_{T_u^{rj}}^{-1}(\lambda) > \Phi_{T_u^{ri}}^{-1}(\lambda),\ \forall \lambda \in (0,1)$ always holds .

Proof. $\Phi_{T_u^{rj}}(b) - \Phi_{T_u^{ri}}(b) = \int_0^b \Phi_{T_{ij}}(b-t) f_{T_u^{ri}}(t)dt - \int_0^b f_{T_u^{ri}}(t)dt$

$= \int_0^b \left(\Phi_{T_{ij}}(b-t) - 1\right) f_{T_u^{ri}}(t)dt < 0,\ \forall b \in R^+$

$\Rightarrow \Phi_{T_u^{rj}}^{-1}(\lambda) > \Phi_{T_u^{ri}}^{-1}(\lambda),\ \forall \lambda \in (0,1)$. □

Proposition 3.8. A non-dominated path is acyclic.

Proof. Suppose $p_u^{ri} \ne p_v^{ri} \in P^{ri}$ are two non-dominated paths; p_u^{ri} is acyclic; and p_v^{ri} is the same path as p_u^{ri} except for containing one cycle p_v^{kk} starting and ending at node k. According to Proposition 3.7, $\Phi_{T_v^{ri}}^{-1}(\lambda) > \Phi_{T_u^{ri}}^{-1}(\lambda),\ \forall \lambda \in (0,1)$ always holds. Therefore, $p_u^{ri} \succ p_v^{ri}$ according to the FSD condition, contradicting the assumption that p_u^{ri} and p_v^{ri} are two non-dominated paths. □

3.4. Solution Algorithms for Solving RSPP

In this section, efficient multi-criteria A* algorithms are presented for solving RSPP. Similar to traditional A* algorithm (Hart et al., 1968), a heuristic valuation function $F(p_u^{ri}) = \Phi_{T_u^{ri}}^{-1}(\alpha) + h(i)$ is adopted to represent the path cost of p_u^{ri}, where $h(i)$ is an estimate of travel time budget from node i to destination s, and $h(s) = 0$ at the destination. $F(p_u^{ri})$ reflects the likelihood of the sub-path p_u^{ri} on the reliable shortest path. The heuristic function is admissible if the following inequality is satisfied

$$F(p_u^{rj}) = \Phi^{-1}_{T_u^{rj}}(\alpha) + h(j) \geq F(p_u^{ri}) = \Phi^{-1}_{T_u^{ri}}(\alpha) + h(i) \tag{3.5}$$

Eq. (3.5) indicates that the heuristic function value of $F(p_u^{ri})$ should monotonically increase with path extensions. By using this heuristic valuation function, a higher priority can be assigned to the nodes closer to the destination, so as to reduce the number of examined nodes and speed up the search process.

Based on this idea, two multi-criteria A* algorithms using different labeling techniques are proposed in this chapter. The first solution algorithm (namely *RSPP-LA**) is a label-selection multi-criteria shortest path approach. This solution algorithm is easy to implement. At each iteration, only one non-dominated path (or one label) is selected for path extension and evaluated according to the M-B dominance. To further improve the computational performance, the second solution algorithm (named *RSPP-NA**) is proposed by using node-selection multi-criteria shortest path approach. In the developed *RSPP-NA** algorithm, at each iteration, all non-dominated paths at the same node are simultaneously selected for path extensions and evaluated using established dominance conditions for two sets of paths. The details of *RSPP-LA** and *RSPP-NA** algorithms are given in following sections.

3.4.1 Label-based multi-criteria A* algorithm

Let $P^{ri} = \{p_1^{ri}, ..., p_m^{ri}\}$ be a set of non-dominated paths maintained at each node i. Paths in P^{ri} are sorted in ascending order by mean travel time t_u^{ri}. In the *RSPP-LA** algorithm, non-dominated paths from all nodes are maintained in a scan eligible set, denoted by $SE = \{p_u^{ri}, ..., p_v^{rj}\}$. The non-dominated paths in SE are ordered by increasing the value of the heuristic function, $F(p_u^{ri})$. At each iteration, only one non-dominated path p_u^{ri} at the top of SE (with minimum $F(p_u^{ri})$) is selected from SE for path extensions. A temporary acyclic path without passing the same node twice is constructed by extending the selected path p_u^{ri} to its successor link a_{ij}, denoted by $p_u^{rj} := p_u^{ri} \oplus a_{ij}$. The dominant relationship between the newly

generated path p_u^{rj} and the set of non-dominated paths P^{rj} at node j is determined according to the M-B dominance (Proposition 3.3). If p_u^{rj} is a non-dominated path at node j, it is then inserted into P^{rj} and SE. The newly generated path p_u^{rj} may also dominate a set of paths in P^{rj}, denoted by P_D^{rj}. These dominated paths in P_D^{rj} can be eliminated from P^{rj} and SE. The algorithm continues this path search process until the destination is reached or SE becomes empty. When the algorithm terminates, the reliable shortest path can be determined as the first selected path at the destination node. The steps of the proposed *RSPP-LA** algorithm are given as follows.

Algorithm: *RSPP-LA**

Inputs: O-D nodes (r, s) and on-time arrival probability α

Returns: the reliable shortest path

Step 1. Initialization:

Create a path p^{rr} from r to itself and set $P^{rr} := \{p^{rr}\}$.

Calculate $h(r)$ and $F(p^{rr})$, and set $SE := \{p^{rr}\}$.

Step 2. Label selection:

If $SE = \phi$, then Stop; otherwise, continue.

Select p_u^{ri} at the top of SE and set $SE := SE \setminus \{p_u^{ri}\}$.

If $i = s$, then Stop; otherwise continue.

Step 3. Path extension:

For each successor node $j \in SCS(i)$

If $j \in p_u^{ri}$, then scan next successor node; otherwise, continue.

Generate a new path $p_u^{rj} := p_u^{ri} \oplus a_{ij}$ and calculate $h(j)$ and $F(p_u^{rj})$.

Call procedure $P_D^{rj} :=$ CheckDominance(p_u^{rj}, P^{rj}).

If p_u^{rj} is a non-dominated path, then set $SE := SE \cup \{p_u^{rj}\}$ and $SE := SE \setminus P_D^{rj}$.

End for

Goto Step 2.

Procedure: *CheckDominance*

Inputs: A path p_u^{rj} and a set of non-dominated path P^{rj}

Returns: P_D^{rj} storing the set of paths dominated by p_u^{rj}, and update P^{rj}

Step 1: Initialization

Set $P_D^{rj} := \phi$ and $n := 1$.

Step 2: Dominant relationship determination

While $n \le |P^{rj}|$ and $t_u^{rj} > t_n^{rj}$ ($|P^{rj}|$ is the number of paths in P^{rj})

If $\Phi_{T_u^{rj}}^{-1}(\alpha) > \Phi_{T_n^{rj}}^{-1}(\alpha)$, then return P_D^{rj}.

Set $n := n + 1$.

End while

If $t_u^{rj} = t_n^{rj}$ and $\Phi_{T_u^{rj}}^{-1}(\alpha) > \Phi_{T_n^{rj}}^{-1}(\alpha)$, then return P_D^{rj}.

Insert p_u^{rj} into P^{rj} at n th position and set $n := n + 1$ (by default $|P^{rj}| := |P^{rj}| + 1$).

While $n \le |P^{rj}|$ and $\Phi_{T_u^{rj}}^{-1}(\alpha) < \Phi_{T_n^{rj}}^{-1}(\alpha)$

Set $P^{rj} := P^{rj} \setminus \{p_n^{rj}\}$ and $P_D^{rj} := P_D^{rj} \cup \{p_n^{rj}\}$.

Set $n := n + 1$.

End while

Return P_D^{rj}.

Proposition 3.9. The *RSPP-LA** algorithm can determine the reliable shortest path when the destination node is reached if the heuristic function used is admissible.

Proof. Let $\overline{P}^{rs}$ be the set of paths containing all non-dominated paths between O-D nodes. When the destination node is reached, the path p_u^{rs} is selected from SE. Thus, the heuristic function value of p_u^{rs} ($F(p_u^{rs})$) is the minimum heuristic function value among the paths in SE. Since all paths in $P'^{rs} = \overline{P}^{rs} \setminus \{p_u^{rs}\}$ are extended from SE and the heuristic function value monotonically increase with path extensions, the heuristic function value of p_u^{rs} is less than that of any path in

$P^{'rs}$. As $h(s)=0$, $F(p_u^{rs})=\Phi_{T_u^{rs}}^{-1}(\alpha)$ is the minimum travel time budget in $\overline{P}^{rs}$ and thus p_u^{rs} is the reliable shortest path between O-D nodes. □

The actual performance of the *RSPP-LA** algorithm depends on the quality of heuristic function. A common admissible heuristic function is the Euclidean distance function $h(i)=e_{is}/v_{\max}$; where e_{is} is the Euclidean distance from node i to destination s and $v_{\max}$ is maximum travel speed (or design speed) of the network. However, it should be noted that the higher the $v_{\max}$ used, the greater the computational efforts required. When $v_{\max}$ approach infinite, and $h(i)$ approaches zero; and the A* algorithm becomes a label-setting approach which uses objective function value $\Phi_{T_u^{ri}}^{-1}(\alpha)$ instead of $F(p_u^{ri})$.

Another possible admissible heuristic function is the shortest distance $h(i)=d_{is}/v_{\max}$, where d_{is} is the network distance from node i to destination s. The network distance d_{is} can be calculated by using Dijkstra's algorithm from the reverse direction (from the destination to the origin). This shortest distance heuristic function can provide a better estimation of $h(i)$ but it requires additional computation burden on the shortest distance path finding.

In the worst case, label selection step (Step 2) requires $O(|N||P|Log(|N||P|))$ with the implementation of SE using an F-heap data structure (Fredman and Tarjan, 1987), where $|N|$ is the number of network nodes and $|P|$ is the maximum number of non-dominated paths at one network node. As *CheckDominance* procedure runs in $O(|P|)$, the path extension step (Step 3) requires $O(|A||P|^2)$, where $|A|$ is number of network links. Therefore, the *RSPP-LA** algorithm runs in a time $O(|A||P|^2+|N||P|Log(|N||P|))$.

3.4.2 Node-based multi-criteria A* algorithm

Similar to the above *RSPP-LA** algorithm, non-dominated paths $P^{ri} = \{p_1^{ri}, ..., p_m^{ri}\}$ at the same node i are sorted in ascending order by mean travel time t_u^{ri}. Unlike the *RSPP-LA** algorithm, in the *RSPP-NA** algorithm, all non-dominated in P^{ri} are simultaneously selected and evaluated at each iteration. For this purpose, P^{ri} as a whole (or called node i) is maintained in the scan eligible set, $SE = \{P^{ri}, ..., P^{rj}\}$. The nodes in SE are in ascending order, based on their heuristic function $F(i)$, which is defined as the minimum $F(p_u^{ri})$ for $\forall p_u^{ri} \in P^{ri}$.

At each iteration, a node i at the top of the priority queue (with minimum $F(i)$) is selected from SE and moved to the set of the selected queue, denoted by Q. A temporary set of paths is constructed from the selected path set P^{ri} (associated at node i) to each successor node j, denoted by $\hat{P}^{rj} = P^{ri} \oplus a_{ij}$. The newly generated paths in $\hat{P}^{rj}$ are then merged into the set of existing non-dominated paths P^{rj}. The newly generated non-dominated paths from $\hat{P}^{rj}$ may also dominate existing paths in P^{rj} which should be discarded. As the algorithm progresses, each node is updated in such a way that all paths at each node are always non-dominated paths (according to the *Update* procedure). The algorithm continues the path extension process until the destination is reached or SE becomes empty. When the algorithm terminates, the reliable shortest path can be determined (the last path in P^{rs} according to Proposition 3.4). The steps of the proposed *RSPP-NA** algorithm are described below.

Algorithm *RSPP-NA**

Inputs: O-D pair, On-time arrival probability α

Returns: the reliable shortest path

Step 1. Initialization:

Create a path p^{rr} from r to itself and set $P^{rr} := \{p^{rr}\}$.

Set $F(r) := h(r)$, $SE := \{P^{rr}\}$ and $Q := \phi$.

Step 2. Node selection:

If $SE = \phi$, stop; otherwise, continue.

Select P^{ri} at the top of SE.

Set $SE := SE \setminus \{P^{ri}\}$ and $Q := Q \cup \{P^{ri}\}$.

If $i = s$, stop; otherwise, continue.

Step 3. Path extension:

For each successor node $j \in SCS(i)$

Call procedure $P_N^{rj} := Update(P^{ri}, a_{ij}, P^{rj})$.

If $P^{rj} \notin SE$ and $P^{rj} \notin Q$, calculate $F(j)$ and set $SE := SE \cup \{P^{rj}\}$.

If $P^{rj} \in SE$, calculate $F(j)$. If $F(j)$ decreased, update the order of P^{rj} in SE.

If $P^{rj} \in Q$ and $P_N^{rj} \neq \phi$, call procedure $Extend(P_N^{rj})$.

End for

Go back to Step 2.

Procedure *Update*

Inputs: A set of non-dominated paths P^{ri} at selected node i, a set of existing non-dominated paths P^{rj} at successor node j, and the link a_{ij}

Returns: A set of newly generated non-dominated paths P_N^{rj} at node j

Step 1. Fast dominant check (based on Proposition 3.6):

If $t_n^{rj} \leq t_1^{ri} + t_{ij}$ and $Z_\alpha(\sigma_n^{rj})^2 < Z_\alpha(\hat{\sigma}_m^{ri})^2 + Z_\alpha(\sigma_{ij})^2$, return $P_N^{rj} := \phi$.

If $t_1^{rj} \leq t_1^{ri} + t_{ij}$ and $Z_\alpha(\sigma_1^{rj})^2 < Z_\alpha(\sigma_m^{ri})^2 + Z_\alpha(\sigma_{ij})^2$, return $P_N^{rj} := \phi$.

Step 2. Path extension operation:

Set $\hat{p}_1^{rj} := p_1^{ri} \oplus a_{ij}$ and $F(j) := \Phi_{\hat{T}_1^{rj}}^{-1}(\alpha)$.

For each $p_u^{ri} \in P^{ri}$

Set $\hat{p}_u^{rj} := p_u^{ri} \oplus a_{ij}$.

If $\Phi_{\hat{T}_u^{rj}}^{-1}(\alpha) \leq F(j)$, set $F(j) := \Phi_{\hat{T}_u^{rj}}^{-1}(\alpha)$ and $\hat{P}^{rj} := \hat{P}^{rj} \cup \{\hat{p}_u^{rj}\}$.

End for

Step 3. Fast dominant check (based on Proposition 3.5):

If $t_n^{rj} \leq \hat{t}_1^{rj}$ and $\Phi_{T_n^{rj}}^{-1}(\alpha) < \Phi_{\hat{T}_m^{rj}}^{-1}(\alpha)$, return $P_N^{rj} := \phi$.

If $t_1^{rj} \le \hat{t}_1^{rj}$ and $\Phi_{T_1^{rj}}^{-1}(\alpha) < \Phi_{\hat{T}_m^{rj}}^{-1}(\alpha)$, return $P_N^{rj} := \phi$.

If $\hat{t}_m^{rj} \le t_1^{rj}$ and $\Phi_{\hat{T}_m^{rj}}^{-1}(\alpha) < \Phi_{T_n^{rj}}^{-1}(\alpha)$, set $P^{rj} := \hat{P}^{rj}$ and return $P_N^{rj} := \hat{P}^{rj}$.

If $\hat{t}_1^{rj} \le t_1^{rj}$ and $\Phi_{\hat{T}_1^{rj}}^{-1}(\alpha) < \Phi_{T_n^{rj}}^{-1}(\alpha)$, set $P^{rj} := \hat{P}^{rj}$ and return $P_N^{rj} := \hat{P}^{rj}$.

Step 4. Merge operation:

Call procedure $P_N^{rj} := ModifiedMerge(P^{rj}, \hat{P}^{rj})$ and return P_N^{rj}.

Procedure *Extend*

Inputs: a set of non-dominated paths P_N^{rj} at node j

For each successor node $w \in SCS(j)$

Call procedure $P_N^{rw} := Update(P_N^{rj}, a_{jw}, P^{rw})$.

If $P^{rw} \in Q$ and $P_N^{rw} \neq \phi$, call procedure $Extend(P_N^{rw})$.

End for

The following is a description of the *Update* procedure. In step 1, a fast dominant check is carried out using Proposition 3.6 before the extension and merging of operations. If Proposition 3.6 is satisfied, all paths in P^{ri} at selected node i are dominated and can be discarded. A temporary set of paths, $\hat{P}^{rj} = P^{ri} \oplus a_{ij}$, will then be constructed in Step 2. Note that, in this path extension step, some paths may be dominated by the paths ahead of them, according to Proposition 3.4(i). In Step 3, there still exists an opportunity to discard a set of dominated paths using Proposition 3.5 before the merge operation is carried out. In the final step, the merge operation is carried out to combine all newly generated paths in $\hat{P}^{rj}$ and existing paths in P^{rj}. The merge operation returns a set of newly generated non-dominated paths in $\hat{P}^{rj}$, denoted as P_N^{rj}. The merge operation is implemented using the 'modified merge algorithm' in Brumbaugh-Smith and Shier (1989).

In the conventional label-setting algorithm (Dijkstra's algorithm), paths in the selected queue Q are not updated and extended. In the proposed algorithm, newly

generated non-dominated paths (P_N^{rj} return from the *Update* procedure) may need to be inserted into Q and extended to its successor nodes according to the *Extend* procedure. An example illustrating the *Extend* procedure can be found in Fig. 3.5. As shown in the figure, Node 1 and Node 2 (P^{11} and P^{12}) have been selected from SE and a non-dominated path a_{12} has been identified at Node 2. In the following iteration, Node 4 (P^{14}) is selected from SE, and a new non-dominated path at Node 2, $a_{14} \oplus a_{42}$, is generated. The generated non-dominated path, $a_{14} \oplus a_{42}$, should then be inserted into the set of non-dominated paths at Node 2 in Q, and further extended to Node 3, because it may result in a non-dominated path $a_{14} \oplus a_{42} \oplus a_{23}$ at Node 3.

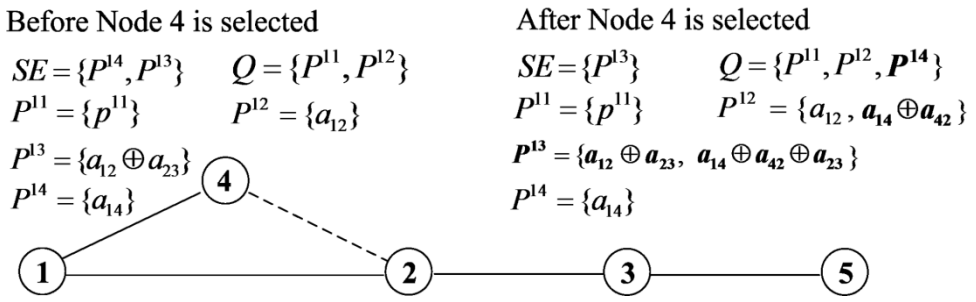


Figure 3.5 An illustration of update operation

Proposition 3.10. The *RSPP-NA** algorithm can determine the reliable shortest path when the destination node is selected from SE, and if the heuristic function is admissible.

Proof. Suppose that $\overline{P}^{rs}$ containing all non-dominated paths between O-D nodes, and P^{rs} is the set of non-dominated paths when destination s is selected from SE. If the heuristic function $F(i)$ is admissible, the selected heuristic function from SE monotonically increases with path extensions. As P^{rs} has been selected from SE, $F(s)$ of P^{rs} is the minimum value in SE. Because all paths in $P^{'rs} = \overline{P}^{rs} - P^{rs}$ are extended from SE, and the $F(s)$ of P^{rs} is less than the that of $P^{'rs}$. As $h(s) = 0$, $F(s)$ of P^{rs} is the minimum $\Phi_{T_u^{rs}}^{-1}(\alpha)$ of $\overline{P}^{rs}$, and thus p_u^{rs} can be determined. □

Compared with the *RSPP-LA** algorithm, the *RSPP-NA** algorithm has a better worst case performance. In the *RSPP-NA** algorithm, the node selection step (Step 2) requires $O(|N|\log|N|)$ with the implementation of *SE* using an F-heap data structure (Fredman and Tarjan, 1987). As *Update* procedure runs in $O(|P|)$, the path extension step (Step 3) requires $O(|A||P|)$. Therefore, the *RSPP-NA** algorithm runs in a time $O\left(|A||P|+|N|\log|N|\right)$.

Theoretically, both the *RSPP-LA** and *RSPP-NA** algorithms have a non-polynomial complexity, because $|P|$ grows exponentially with the network size. However, a number of authors in their numerical experiments (Brumbaugh-Smith and Shier, 1989; Müller-Hannemann and Weihe, 2004; Nie and Wu, 2009b) found that $|P|$ is much smaller than the maximum possible size, especially for sparse networks in transportation applications.

3.5. Computational Performance

The computational performance of the *RSPP-LA** and *RSPP-NA** algorithms on several large-scale networks is examined in this section. The *RSPP-LA** and *RSPP-NA** algorithms were coded in the Visual C# programming language. The priority queue was implemented using the F-heap data structure (Fredman and Tarjan, 1987).

To comparatively evaluate and benchmark the proposed algorithms, an exact parametric approach for solving RSPP under risk-averse scenarios (*RSPP-P*) (Nikolova, 2010) was implemented. Following the recommendations of Nikolova (2010), Dijkstra's algorithm (using the same F-heap data structure) was employed as the deterministic shortest path algorithm for the implementation of *RSPP-P*.

As RSPP can be considered as a bi-criterion shortest path problem (BSPP) with respect to the mean travel time and its variance, the label-correcting algorithm

(*BSPP-LCA*) (Skriver and Andersen, 2000), which is known to be the most efficient for solving BSPP, was also implemented for the purpose of comparison. These two additional solution algorithms were coded in the same environment and tested on the same networks. All experiments were conducted on a ThinkPad X61 laptop with an Intel dual-core 1.6 GHz CPU (only a single processor was used) and 2G RAM.

Table 3.1 Basic characteristics of testing networks

Road networks			Grid networks			Random networks		
Network	$\|N\|$	$\|A\|$	Network	$\|N\|$	$\|A\|$	Network	$\|N\|$	$\|A\|$
RTIS	1,367	3,655	G1 (40*50)	2,000	7,820	R1	3,000	24,541
			G2 (50*100)	5,000	19,700	R2	7,000	57,369
Chicago	12,982	39,018	G3(100*100)	10,000	39,600	R3	14,000	114,532

$|N|$: Number of nodes in the network; $|A|$: Number of links in the network

As shown in Table 3.1, the computational tests were conducted on three different types of network: road networks, grid networks and random networks. NetMaker (Skriver and Andersen, 2000) was used to generate random networks without the mean and variance of link travel times. To test the A* algorithm, the x and y coordinates of nodes were uniformly distributed from [0m, 10,000m], respectively. Based on the node coordinates, the link lengths were calculated as the Euclidean distance between the two nodes of a link. Link speeds were then, uniformly distributed from [10 km/hour, 100 km/hour] to generate the mean travel time by calculating the ratio of link length to link speed. The SD of link travel times was generated by randomly selecting the coefficient of variation (the ratio of SD to mean) from a uniform distribution in the range of [0.1, 1]. This method, to randomly generate the mean and variance of link travel times, was also adopted for the grid networks and the Chicago regional network.

Tables 3.2 and 3.3 report the computational performance of the algorithms. The computational performance was evaluated in terms of computational time (denoted by $\tilde{t}$ in milliseconds) and the number of generated non-dominated paths between O-D nodes (denoted by $\tilde{n}$). Note that $\tilde{n}$ can be interpreted as the number of extreme points in the work of Nikolova (2010). All the reported $\tilde{t}$ and $\tilde{n}$ values

were an average of 100 runs, using different O-D nodes for each run. The 100 O-D nodes were randomly selected for each network and the same set of O-D nodes was used for every test performed on a given network.

Table 3.2 shows the reliable shortest path finding results for risk-averse scenarios. The on-time arrival probability was set as $\alpha = 90\%$. It can be seen from Table 3.2 that the *RSPP-NA** algorithm runs significantly faster than the *RSPP-LA** algorithm for all networks. For example, in the Chicago regional network, the computational time required by the *RSPP-NA** algorithm was 2.36 (167.3/70.71) times less than that consumed by the *RSPP-LA** algorithm. This was mainly due to different labeling techniques used by *RSPP-LA** and *RSPP-NA** algorithms. The *RSPP-LA** algorithm adopts a label-selection method, in which only one non-dominated path (label) with minimum travel time budget is selected for path extension and evaluated according to the M-B dominance. The *RSPP-NA** algorithm is a node-selection method. At each iteration, all non-dominated paths associated with the same node can be simultaneously selected for path extension. This can reduce the computational burden required for maintaining, sorting and selecting elements in the scan eligible (*SE*). In addition, effective dominance conditions for two sets of paths used in the *RSPP-NA** algorithm can reduce computational burden associated with non-dominated path evaluations.

Table 3.2 Computational performance of algorithms for risk-averse scenarios

Network	A*	RSPP-NA*		RSPP-LA*		RSPP-P		BSPP-LCA	
	$\tilde{t}$	$\tilde{t}$	$\tilde{n}$	$\tilde{t}$	$\tilde{n}$	$\tilde{t}$	$\tilde{n}$	$\tilde{t}$	$\tilde{n}$
RTIS	3.51	4.73	1.26	5.42	1.2	67.21	10.82	2,382	34.31
Chicago	**64.41**	**70.71**	2.52	***167.3***	1.95	**2,353**	19.51	*	195.2
G1	5.36	5.60	1.73	8.60	1.55	79.63	7.48	773.33	23.3
G2	19.37	21.69	2.15	46.83	1.95	470.20	12.05	18,533	51.81
G3	36.87	44.91	2.35	112.66	1.98	1,443	15.04	78,481	76.91
R1	61.47	69. 23	1.44	96.72	1.26	620.61	4.96	382.22	9.58
R2	263.2	292.05	1.66	438.38	1.43	3,792	5.09	1,488	10.06
R3	814.8	885.95	1.43	1293.6	1.33	**13,452**	4.84	**3,166**	9.79

$\tilde{t}$: Average computational times in milliseconds (using 100 runs)

$\tilde{n}$: Average numbers of non-dominated paths between O-D nodes (using 100 runs)

*: Computation time > 20minutes

The computational performance of traditional A* algorithm was provided in Table 3.2 as a benchmark of the proposed solution algorithm. It can be observed from Table 3.2 that the proposed *RSPP-NA** algorithm runs slightly slower than the traditional A* algorithm. This result is due to the fact that, during the path searching process of *RSPP-NA** algorithm, several non-dominated paths have to be generated, maintained and evaluated at each network nodes. In the traditional A* algorithm, only one path is required to be maintained.

It is also be observed from Table 3.2 that the proposed *RSPP-NA** algorithm performed substantially better than the *RSPP-P* and *BSPP-LCA* algorithms for all types of networks. For instance, the *RSPP-NA** required 70.71 milliseconds to determine the reliable shortest path in the Chicago regional network. This was 33.3 (2,353/70.71) times faster than the *RSPP-P* algorithm and 19,309 (1,365,337/70.71) times faster than the *BSPP-LCA* algorithm.

The computational advantage of the *RSPP-NA** algorithm was expected since it generated fewer non-dominated paths than those generated by *RSPP-P* and *BSPP-LCA*. Fig. 3.6 illustrates the different ways in which these three solution algorithms generate non-dominated paths. The *BSPP-LCA* algorithm first determines all of the non-dominated paths between the O-D nodes (all of the points in the Fig. 3.6), and then chose the reliable shortest path with minimum travel time budget. Therefore, this solution algorithm generated $\tilde{n} = 195.2$ non-dominated paths between O-D nodes for the Chicago regional network.

As RSPP in risk-averse scenarios can be formulated as a parametric of mean travel time and its variance, the *RSPP-P* algorithm only determined the non-dominated paths on the convex hull, without consideration of non-dominated paths in the shaded areas shown in Fig. 3.6. Each non-dominated path on the convex hull was obtained by a search of the shortest path in a deterministic network using Dijkstra's algorithm. As such, fewer non-dominated paths, $\tilde{n} = 19.51$, are generated for the Chicago regional network. The proposed *RSPP-NA** algorithm, however, can determine the reliable shortest path as soon as destination node was reached. Thus, only a few non-dominated paths ($\tilde{n} = 2.52$ for the Chicago regional network) were generated.

Therefore, the proposed *RSPP-NA** algorithm performed better than the other two algorithms, especially for networks with a large number of non-dominated paths.

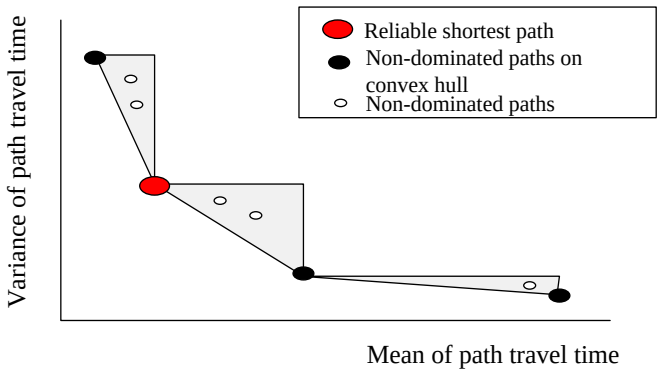


Figure 3.6 Non-dominated paths ($\tilde{n}$) generated by three different algorithms

It can also be seen from Table 3.2 that the *RSPP-P* algorithm result was superior to that of the *BSPP-LCA* algorithm for road networks and grid networks, where the number of non-dominated paths is large and link density small ($|A|/|N|$ was about 3 to 4). However, the performance of *RSPP-P* may even be inferior to *BSPP-LCA* for random networks with a few non-dominated paths and a large density ($|A|/|N|$ was about 8). It should be noted that *RSPP-NA** was still the best performer even for these random networks.

Table 3.3 reports the results when solving RSPP under risk-seeking scenarios ($\alpha = 10\%$). Under risk-seeking scenarios, the *RSPP-P* and *BSPP-LCA* algorithms cannot be used to solve RSPP. This is because risk-seeking travellers tend to choose optimal paths with smaller means but larger variances. In this case, reliable shortest paths are not guaranteed on the convex hull and thus the *RSPP-P* algorithm cannot be used for solving the problem. In addition, when travellers are risk-seeking, the non-dominated paths may contain an infinite number of cycles under the M-V dominance condition. As a result, the *BSPP-LCA* algorithm is computationally intractable for solving RSPP under risk-seeking scenarios. In contrast to these two algorithms, the *RSPP-LA** and *RSPP-NA** algorithms, based on M-B dominance can solve the RSPP in the same way for both risk-averse and risk-seeking scenarios. Therefore, the proposed *RSPP-LA** and *RSPP-NA** algorithms can determine the

reliable shortest path for risk-seeking travellers as efficiently as for the risk-averse travellers.

Table 3.3 Computational performance of algorithms for risk-seeking scenarios

Network	RSPP-NA*		RSPP-LA*	
	$\tilde{t}$	$\tilde{n}$	$\tilde{t}$	$\tilde{n}$
Hong Kong RTIS	4.17	1.21	5.25	1.21
Chicago Region	65.77	2.50	295.58	2.04
G1	4.53	1.74	12.09	1.98
G2	17.13	1.92	53.96	1.76
G3	40.15	1.95	133.96	1.87
R1	57.98	1.54	119.76	1.38
R2	247.81	1.58	423.10	1.30
R3	725.25	1.19	1240.8	1.22

$\tilde{t}$: Average computational times in milliseconds (100 runs)

$\tilde{n}$: Average numbers of non-dominated paths between O-D nodes (100 runs)

3.6. Extension to Lognormal Distributions

As indicated in Chapter 2, many empirical studies have found that both normal and lognormal distributions can be considered to be reasonable assumptions from a practical standpoint, but link travel times in congested urban road networks may fit lognormal distributions better. In this section, the proposed multi-criteria A* algorithms are extended to lognormal distributions.

When path travel times follow lognormal distributions, the path travel time distributions can also be generated by calculating their mean and variance in the same way as the normal distribution case, since the lognormal distribution is a two-parameter distribution. The established M-V and M-B dominance conditions (Propositions 3.2 and 3.3), however, cannot be used, the FSD condition (Proposition 3.1) only, can be adopted to determine dominated paths as shown below.

To determine dominated paths under lognormal distributions, the natural logarithm of path travel times can be used. Let t_u^{ri} and σ_u^{ri} be the mean and SD of path travel time T_u^{ri}. The mean and variance of natural logarithm of T_u^{ri}, respectively denoted

by $u(T_u^{ri})$ and $v(T_u^{ri})$, can be expressed as

$$v(T_u^{ri}) = \ln(1 + (\sigma_u^{ri} / t_u^{ri})^2) \tag{3.6}$$

$$u(T_u^{ri}) = \ln(t_u^{ri}) - 0.5 * v(T_u^{ri}) \tag{3.7}$$

The inverse CDF of path travel time can then be calculated by

$$\Phi_{T_u^{ri}}^{-1}(\alpha) = \exp\left(u(T_u^{ri}) + Z_\alpha v(T_u^{ri})\right) \tag{3.8}$$

Consequently, given two paths $p_u^{ri} \neq p_v^{ri} \in P^{ri}$, p_u^{ri} dominates p_v^{ri} under the FSD condition, if $\Phi_{T_u^{ri}}^{-1}(0.001) < \Phi_{T_v^{ri}}^{-1}(0.001)$ and $\Phi_{T_u^{ri}}^{-1}(0.999) < \Phi_{T_v^{ri}}^{-1}(0.999)$ are satisfied. The parameters $Z_{0.001} = -3.0801$ and $Z_{0.999} = 3.0801$ can be used in Eq. (3.8) to calculate $\Phi_{T_u^{ri}}^{-1}(0.001)$ and $\Phi_{T_u^{ri}}^{-1}(0.999)$ respectively.

By using the FSD condition, the proposed label-selection multi-criteria A* algorithm (*RSPP-LA**), in Section 3.4.1, can be easily modified to solve RSPP under lognormal distributions. In the modified *RSPP-LA** algorithm, all non-dominated paths $P^{ri} = \{p_1^{ri}, ..., p_m^{ri}\}$ maintained at each node i are sorted in ascending order instead of mean travel time, by $\Phi_{T_u^{ri}}^{-1}(0.001)$. The *CheckDominance* procedure can also be modified by using $\Phi_{T_u^{ri}}^{-1}(0.001)$ and $\Phi_{T_u^{ri}}^{-1}(0.999)$ instead of t_u^{ri} and $\Phi_{T_u^{ri}}^{-1}(\alpha)$. After these simple modifications, the *RSPP-LA** algorithm can be used for solving RSPP under lognormal distributions.

3.7. Summary

In this chapter, the reliable shortest path problem (RSPP) has been investigated. RSPP is formulated and solved using a multi-criteria shortest path approach. Several dominance conditions, first order stochastic dominance (FSD), M-V dominance and M-B dominance, have been established to determine dominated paths for RSPP. The FSD condition can be used to determine dominated paths for any types of travel time distributions. Compared with the FSD condition, the M-V and M-B dominance conditions can help reduce the number of generated non-dominated paths when

travel times follow normal distributions.

Based on above established dominance conditions, two new multi-criteria A* algorithms (*RSPP-LA** and *RSPP-NA**) were proposed to solve RSPP. The *RSPP-LA** algorithm is a label-selection approach. This solution algorithm is easy to implement. At each iteration, only one non-dominated path is selected for path extension and evaluated according to M-B dominance. The *RSPP-NA** algorithm is a node-selection approach. At each iteration, all non-dominated paths at the same node are simultaneously selected for path extensions and evaluated using established dominance conditions for two sets of paths. Compared with *RSPP-LA** algorithm, this *RSPP-NA** algorithm has better computational performance in both average and worst cases.

It is assumed in this chapter that link travel times are statistically independent. Therefore, the spatial correlations of link travel times in this part of the complete study have not been considered and modeled explicitly. In order to obtain more accurate reliable shortest path results, the proposed solution algorithms are extended to take account travel time spatial correlations and further described in Chapter 4.

4. Spatial-dependent Reliable Shortest Path Problems

The development of new efficient solution algorithms was described in the last chapter. These algorithms aim at solving the reliable shortest path problem (RSPP) in stochastic networks, where link travel times are statistically independent. The spatial correlations of link travel times have yet to be considered in last chapter. As mentioned in Chapter 1, many empirical studies have found that travel times are strongly correlated among neighboring links (Chan et al., 2009), largely due to traffic flow propagations over time and space. For example, a traffic accident on a major urban road may also cause significant travel delays on that road's upstream links. The spatial correlations have also been measured (in terms of variance-covariance matrices) in advanced traveller information systems (ATIS) as an important data source for travel time estimation and prediction (Tam and Lam 2008; Chan et al., 2009; El Esawey and Sayed, 2011). Therefore, travel time correlations are necessary inclusions in RSPP. In this study, RSPP with spatial correlations is referred to, hereafter, as the spatial-dependent reliable shortest path problem (SD-RSPP).

In this chapter, the efficient multi-criteria A* algorithm presented in Chapter 3 is extended to solve SD-RSPP based on a *k* limited spatial dependence assumption. In this assumption, the travel time of a link is assumed to be spatially correlated only with its neighboring links within a local 'impact area'. Dominance conditions established in Chapter 3 are extended for SD-RSPP under this *k* limited spatial dependence assumption. An equivalent two-level hierarchical network is proposed to represent travel time correlations among neighboring links, and facilitate reliable shortest path findings based on established dominance conditions. Using this hierarchical network, an efficient multi-criteria A* algorithm is proposed to solve SD-RSPP, exactly.

This chapter is organized as follows. Section 4.1 briefly reviews related studies. Section 4.2 presents the definition of SD-RSPP. Section 4.3 introduces the dominance conditions for SD-RSPP. Section 4.4 presents the two-level hierarchical

network. The solution algorithm for solving SD-RSPP is described in Section 4.5. The numerical examples using data from a real-world ATIS in Hong Kong is reported in Section 4.6. Finally, a summary of this chapter is given in Section 4.7.

4.1. Background

To writer's knowledge, SD-RSPP has not received much attention in the literature. Nie and Wu (2009a) studied SD-RSPP by assuming that the probability density function (PDF) of the travel time of a link is conditional on the state of travellers arriving at the tail node of that link. The FSD condition and the dynamic programming approach can be employed, based on this assumption, to solve SD-RSPP. However, it is a formidable task for ATIS to generate such probabilities for a large number of links in real road networks.

As an alternative approach, Ji et al. (2011) formulated the spatial correlations as variance-covariance matrices which can be directly obtained from ATIS. Based on this formulation, a simulation-based method was proposed to solve SD-RSPP. Nevertheless, the simulation-based method is computationally expensive and the precision of results is dependent on the maximum number of simulations. Using the same formulation of variance-covariance matrix, Nikolova (2009) proposed a network transformation technique to solve SD-RSPP using the parametric approach. However, this parametric method cannot provide a solution for risk-seeking travellers; and such method may encounter infinite negative cycles in the transformed network due to the negative travel time covariance.

The aim of this chapter is to investigate SD-RSPP in the context of ATIS-based routing systems, so as to aid various travellers (including risk-averse, risk-neutral and risk-seeking travellers) make their route choice decisions under travel time uncertainties. The link travel time correlations are thus represented by variance-covariance matrices as the work of Ji et al. (2011).

As indicated above, an efficient multi-criteria A* algorithm is proposed to solve SD-RSPP based on a *k* limited spatial dependence assumption. As also indicated, in

this assumption, the travel time of a link is assumed to be spatially correlated only with its neighboring links within a local impact area, in which the topological distance (measured by number of links) between any two links is less than or equal to *k*. The consideration of this k limited spatial dependence in SD-RSPP extends the work of Nie and Wu (2009a) and Nikolova (2009), which considers only travel time correlations on the adjacent links ($k = 1$).

To some extent, such *k* limited spatial dependence can be interpreted as Tobler's First Law of Geography that 'all things are related, but nearby things are more related than distant things' (Tobler, 1970). Empirical studies based on field observations also found travel times, among neighboring links, to be strongly correlated (Gajewski and Rilett, 2003). The correlation is usually very low for links that are spatially distant, even on the same street (El Esawey and Sayed, 2011).

4.2. Problem Statement

Let $G = (N, A, \Psi)$ be a directed network consisting of a set of nodes N, a set of links A, and a set of movements Ψ. Each link $a_{ij} \in A$ has a tail node $i \in N$, a head node $j \in N$ and a random travel time T_{ij}. The mean and standard deviation (SD) of link travel time are denoted by t_{ij} and σ_{ij} respectively. Each node i has a set of successor nodes $SCS(i) = \{j : a_{ij} \in A\}$ and a set of predecessor nodes $PDS(i) = \{w : a_{wi} \in A\}$. The movement $\psi_{wij} = (a_{wi}, a_{ij}) \in \Psi$ represents an allowed movement (e.g. through-movement or right-turn) at node i.

Suppose that the nodes $r \in N$ and $s \in N$ represent the O-D nodes. Let $p_u^{rs} = \{a^1, ..., a^m, ..., a^\lambda\}$ be a path from origin r to destination s, consisting of λ consecutive links. The path travel time, denoted by T_u^{rs}, is the sum of the related link travel times along the path as

$$T_u^{rs} = \sum_{m=1}^{\lambda} T_{ij}^m \tag{4.1}$$

where T_{ij}^{m} is the travel time distribution of a^{m} (the m[th] link along path p_u^{rs}).

As previously indicated, the travel time of link a_{ij} in this chapter is assumed to be spatially correlated only within a local impact area, denoted by $G_{ij}^{k} = \left(N_{ij}^{k}, A_{ij}^{k}, \Psi_{ij}^{k}\right)$. Let d_{ij}^{qw} be the topological distance (measured by number of links) between links a_{ij} and a_{qw}. A link a_{qw} is said to be a *k*-neighboring link of link a_{ij} if and only if $d_{ij}^{qw} = k$. With this concept, the impact area G_{ij}^{k} can be formally defined as a sub-network of G, satisfying $d_{ij}^{qw} \le k$, $\forall a_{qw} \in A_{ij}^{k}$.

As indicated in Chapter 3, it is considered reasonable for link travel times to follow either normal or lognormal distributions. To facilitate the presentation of the essential ideas, link travel time distributions are assumed to follow normal distributions in this chapter. It should be noted that the proposed algorithm can also be used to solve SD-RSPP with lognormal distributions.

Under these two assumptions, the path travel time T_u^{rs} follows a multivariate normal distribution. Its mean and SD, respectively denoted by t_u^{rs} and σ_u^{rs}, can be calculated as

$$t_u^{rs} = \sum_{m=1}^{\lambda} t_{ij}^{m} \tag{4.2}$$

$$\sigma_u^{rs} = \sqrt{\sum_{m=1}^{\lambda} (\sigma^{m})^2 + \sum_{n=1}^{k} \sum_{m=1}^{\lambda-n} 2\,\mathrm{cov}(T^{m}, T^{m+n})} \tag{4.3}$$

where $\mathrm{cov}(T^{m}, T^{m+n})$ is the travel time covariance between links a^{m} and a^{m+n}.

Let $\Phi_{T_u^{rs}}^{-1}(\alpha)$ be the inverse of the cumulative distribution function (CDF) of path travel time T_u^{rs} at α confidence level. As formulated and shown in Chapter 3, it can be expressed as

$$\Phi_{T_u^{rs}}^{-1}(\alpha) = t_u^{rs} + z_{\alpha}\sigma_u^{rs} \tag{4.4}$$

where z_{α} is the inverse CDF of standard normal distribution at α confidence level.

The on-time arrival probability α represents travellers' attitudes towards risks of being late ($\alpha > 0.5$, $\alpha = 0.5$ and $\alpha < 0.5$ for risk-averse, risk-neutral and risk-seeking attitudes, respectively). The value of α can be pre-determined based on travellers' trip purposes. Therefore, SD-RSPP can be formally expressed as the following optimization problem

$$Min \quad \Phi^{-1}_{T_u^{rs}}(\alpha) \tag{4.5}$$

Subject to

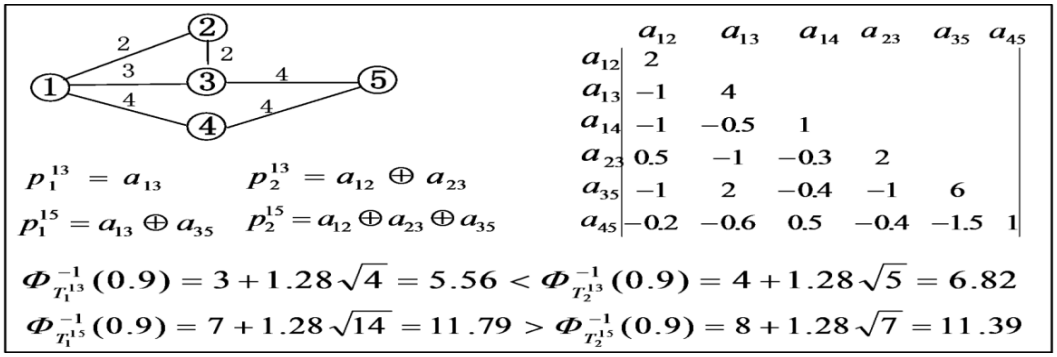
$$T_u^{rs} = \sum_{a_{ij} \in A} T_{ij} \delta_{ij}^{rs,u} \tag{4.6}$$

$$\sum_{j \in SCS(i)} \delta_{ij}^{rs,u} - \sum_{w \in PDS(i)} \delta_{wi}^{rs,u} = \begin{cases} 1 & \forall\, i = r \\ 0, & \forall\, i \neq r; i \neq s \\ -1 & \forall\, i = s \end{cases} \tag{4.7}$$

$$\delta_{ij}^{rs,u} \in \{0,1\}, \quad \forall a_{ij} \in A \tag{4.8}$$

where $\delta_{ij}^{rs,u}$ be the decision variable regarding the link-path incidence relationship; $\delta_{ij}^{rs,u} = 1$ means that the link a_{ij} is on the path p_u^{rs}, and otherwise $\delta_{ij}^{rs,u} = 0$. Eq. (4.5) represents the travel time budget which travellers want to minimize. Eq. (4.6) defines the path travel time as mentioned in Eqs. (4.1-4.3). Eq. (4.7) ensures that the reliable shortest path is feasible. Eq. (4.8) is concerned with the link-path incidence variables which should be binary in nature.

Fig. 4.1 illustrates the above concept, by means of a small network. In Fig. 4.1, all link travel times follow normal distributions. The mean link travel times are shown on the links while the link travel time variance and covariance are given in the matrix. In the variance-covariance matrix, elements along the diagonal are the variance of link travel times and off-diagonal elements are the covariance between two links. As the matrix is symmetric, only a lower triangular matrix is shown in the figure. It should be noted that when all off-diagonal elements are zero, SD-RSPP reduces to RSPP as discussed in Chapter 3.



	a_{12}	a_{13}	a_{14}	a_{23}	a_{35}	a_{45}
a_{12}	2					
a_{13}	−1	4				
a_{14}	−1	−0.5	1			
a_{23}	0.5	−1	−0.3	2		
a_{35}	−1	2	−0.4	−1	6	
a_{45}	−0.2	−0.6	0.5	−0.4	−1.5	1

Figure 4.1 An illustrative example

With spatial correlations on link travel times, the dominance conditions established in Chapter 3 cannot be used to determine dominated paths for SD-RSPP. As illustrated in Fig. 4.1, $p_1^{13} = a_{13}$ and $p_2^{13} = a_{12} \oplus a_{23}$ are two paths from Node 1 to Node 3. According to M-V dominance (Proposition 3.2), p_1^{13} dominates p_2^{13}, since $t_1^{13} = 3 < t_2^{13} = 4$ and $\sigma_1^{13} = 4 < \sigma_2^{13} = 5$. According to Definition 3.3, $p_1^{15} = p_1^{13} \oplus a_{35}$ should dominate $p_2^{15} = p_2^{13} \oplus a_{35}$ and $\Phi_{T_1^{15}}^{-1}(0.9) < \Phi_{T_2^{15}}^{-1}(0.9)$ should hold. However, in SD-RSPP, $\Phi_{T_1^{15}}^{-1}(0.9) = 11.79 > \Phi_{T_2^{15}}^{-1}(0.9) = 11.39$, due to the effects of travel time spatial correlations (i.e. $\text{cov}(T_1^{13}, T_{35}) = \text{cov}(T_{13}, T_{35}) = 2$, while $\text{cov}(T_2^{13}, T_{35}) = \text{cov}(T_{12}, T_{35}) + \text{cov}(T_{23}, T_{35}) = -2$). Therefore, the established dominance conditions and solution algorithms, presented in Chapter 3, cannot be used to solve SD-RSPP.

4.3. Multi-criteria Shortest Path Approach for Solving SD-RSPP

In this section, the dominance conditions, established in Chapter 3, are extended to solve SD-RSPP when the *k* limited spatial dependence assumption is used. Let $p^{ij,k} = \{a^1, ..., a^k\}$ be a path from node i to node j consisting of k consecutive links, and $p_u^{rj,k+\lambda} = p_u^{ri,\lambda} \oplus p^{ij,k}$ be a path from origin r to node j going through sub-path $p^{ij,k}$. The first-order stochastic dominant (FSD) condition for SD-RSPP can be formally defined as below.

Proposition 4.1 (FSD condition). Given two paths $p_u^{rj,\lambda+k} \neq p_v^{rj,\eta+k} \in P^{rj}$, $p_u^{rj,\lambda+k} \succ p_v^{rj,\eta+k}$ if $p_u^{rj,k+\lambda}$ and $p_v^{rj,k+\eta}$ satisfy $\Phi_{T_u^{rj}}^{-1}(y) < \Phi_{T_v^{rj}}^{-1}(y), \forall y \in (0,1)$.

Proof. See Appendix B. □

In addition to the FSD condition, following mean-variance (M-V) dominance exists:

Proposition 4.2. (M-V dominance) Given an on-time arrival probability α and two paths $p_u^{rj,\lambda+k} \neq p_v^{rj,\eta+k} \in P^{rj}$, $p_u^{rj,\lambda+k} \succ p_v^{rj,\eta+k}$ if $p_u^{rj,k+\lambda}$ and $p_u^{rj,k+\eta}$ satisfy either

(i) $t_u^{rj} \leq t_v^{rj}$ and $Z_\alpha \sigma_u^{rj} < Z_\alpha \sigma_v^{rj}$ or

(ii) $t_u^{rj} < t_v^{rj}$ and $Z_\alpha \sigma_u^{rj} \leq Z_\alpha \sigma_v^{rj}$

Proof. See Appendix B. □

Fig. 4.2 illustrates the above two established dominance conditions in a simple network when *k* = 3 is considered. As shown in Fig. 4.2, four paths from Node 1 to Node 8 go through the same sub-path $p^{58,3} = a_{56} \oplus a_{67} \oplus a_{78}$ with three links. The mean and travel time standard deviation of these four paths are given in Fig. 4.2a, and the CDF of four path travel time distributions are illustrated in Fig. 4.2b.

It is observed from Fig. 4.2b that the path $p_4^{18,5}$ is FSD dominated by the path $p_1^{18,5}$ since $\Phi_{T_1^{18}}^{-1}(y) < \Phi_{T_4^{18}}^{-1}(y), \forall y \in (0,1)$. Thus, for all travellers with different on-time arrival probabilities, $p_4^{18,5}$ can be discarded in the path search process. The other three paths ($p_1^{18,5}$, $p_2^{18,5}$ and $p_3^{18,6}$) should be maintained as FSD non-dominated paths. If a traveller is risk-averse, $p_3^{18,6}$ can be further eliminated as an M-V dominated path since $\Phi_{T_1^{18}}^{-1}(y) < \Phi_{T_3^{18}}^{-1}(y), \forall y \in [0.31,1)$. Similarly, according to the M-V dominance condition, the path $p_2^{18,5}$ can be also determined as the M-V dominated path for a risk-seeking traveller, since $\Phi_{T_1^{18}}^{-1}(y) < \Phi_{T_2^{18}}^{-1}(y), \forall y \in (0, 0.68]$. Therefore, with the given on-time arrival probability, the M-V dominance condition can help determine potential dominated paths, which may not be identified under the

FSD condition.

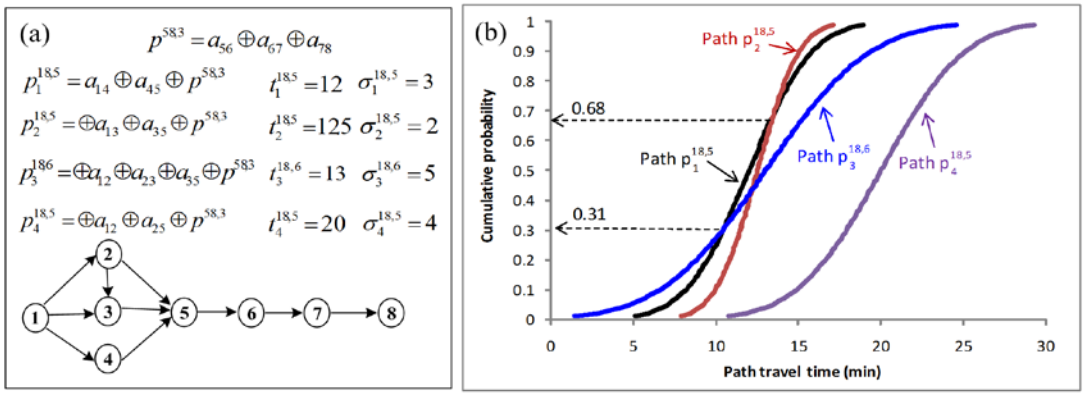


Figure 4.2 An illustration of dominance conditions

In addition, the monotonic property is also satisfied in the SD-RSPP property as follows:

Proposition 4.3. Given two paths p_u^{rj} and $p_u^{r\ell} = p_u^{rj} \oplus a_{j\ell}$, the relationship $\Phi_{T_u^{r\ell}}^{-1}(\alpha) > \Phi_{T_u^{rj}}^{-1}(\alpha) \quad \forall \alpha \in (0,1)$ always holds .

Proof. $\Phi_{T_u^{r\ell}}^{-1}(\alpha) - \Phi_{T_u^{rj}}^{-1}(\alpha) = t_{j\ell} + Z_\alpha \left(\sqrt{(\sigma_u^{rj})^2 + 2\,\mathrm{cov}(T_u^{rj}, T_{j\ell}) + \sigma_{j\ell}^2} - \sigma_u^{rj} \right)$. When $\alpha \geq 0.5$, we have $\Phi_{T_u^{r\ell}}^{-1}(\alpha) - \Phi_{T_u^{rj}}^{-1}(\alpha) \geq t_{j\ell} + Z_\alpha \left(\sqrt{(\sigma_u^{rj} - \sigma_{j\ell})^2} - \sigma_u^{rj} \right) = t_{j\ell} - Z_\alpha \sigma_{j\ell}$. Due to the non-negative property of travel time of link $a_{j\ell}$, we have $t_{j\ell} - Z_\alpha \sigma_{j\ell} > 0$ and thus $\Phi_{T_u^{r\ell}}^{-1}(\alpha) - \Phi_{T_u^{rj}}^{-1}(\alpha) > 0$. Similarly, $\Phi_{T_u^{r\ell}}^{-1}(\alpha) - \Phi_{T_u^{rj}}^{-1}(\alpha) > 0$ can be proved when $\alpha < 0.5$. □

As indicated in Chapter 3, the reliable shortest path in RSPP (with independent link travel times) is acyclic without passing the same node more than once. Such a property may not be satisfied by the reliable shortest path in SD-RSPP; but it should not pass the same link twice, as follows.

Proposition 4.4. The reliable shortest path must not pass the same link more than once.

Proof. Suppose $p_u^{rs} = p_u^{ri} \oplus a_{ij} \oplus \ldots \oplus a_{ij} \oplus p_u^{js}$ is the reliable shortest path passing

the link a_{ij} twice. There exists a path $p_v^{rs} = p_u^{ri} \oplus a_{ij} \oplus p_u^{js}$ passing the link a_{ij} only once. Since p_u^{rs} passes more links in sub-path $a_{ij} \oplus ... \oplus a_{ij}$, we have $\Phi_{T_u^{rs}}^{-1}(\alpha) > \Phi_{T_v^{rs}}^{-1}(\alpha) \quad \forall \alpha \in (0,1)$ according to Proposition 4.3. Therefore, p_u^{rs} is not the reliable shortest path according to Definition 3.3, contradicting the assumption that p_u^{rs} is the reliable shortest path. □

For convenience, in SD-RSPP, a path which does not pass the same link twice is hereafter referred to as the acyclic path.

4.4. Two-level Hierarchical Network

In this section, a two-level hierarchical network to represent road networks with spatial correlated link travel times, is proposed. For clarity, the network G presented in Section 4.2 is hereafter referred to as the primal network.

As previously indicated, SD-RSPP can be formulated using a multi-criteria shortest path approach, and a set of non-dominated paths is maintained and evaluated at the same sub-path with k consecutive links in the primal network. This generalized dynamic programming approach, however, may not be easily implemented in the primal network, because the sub-path with k consecutive links is not explicitly represented as a basis network entity.

To facilitate such a path search approach using k consecutive primal links as basis network entities, a two-level hierarchical network, denoted by $HG = \left(H^g, H^t\right)$, is proposed. The proposed hierarchical network has two hierarchies. The ground hierarchy H^g consists of $|\mathrm{N}|$ directed-in-trees, where $|\mathrm{N}|$ is the number of nodes in the primal network G. For each primal node $i \in G$, a directed-in-tree $\overline{G}_i^g = (\overline{N}_i, \overline{A}_i)$ is constructed rooting at this primal node. In each directed-in-tree $\overline{G}_i^g$, a ground hierarchical node $\overline{n}_u^{ij,\lambda} \in \overline{G}_i^g$ represents a primal path $p_u^{ij,\lambda} \in G$ with λ consecutive primal links ($\lambda \le k-1$ always holds). In this way, all primal paths

$\forall p_u^{ij,\lambda} \in G$ with λ links ($\lambda \le k-1$) can be represented in the ground hierarchy.

Fig. 4.3b illustrates the construction of the ground hierarchy, when $k=3$, from the primal network, shown in Fig. 4.3a. As illustrated in Fig. 4.3b, there are nine directed-in-trees constructed for all nodes in the primal network. In Fig. 4.3b, a node in these trees corresponds to a path in the primal network. For example, node $\overline{n}_{AB}$ in Fig. 4.3b represents the path $p_1^{15,2} = a_A \oplus a_B$ in the primal network.

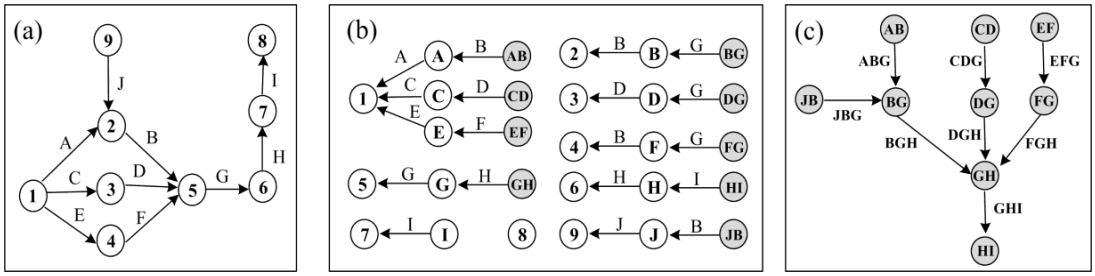


Figure 4.3 An illustration of hierarchical network

(a) Primal network G (b) Ground hierarchy H^g (c) Top hierarchy H^t

The nodes in the ground hierarchy can be classified into two categories: border nodes and local nodes. The border nodes for a tree $\overline{G}_i^g$, denoted by $BORDER(\overline{G}_i^g) = \{\forall \overline{n}_u^{ij,\lambda}, \lambda = k-1\}$, are defined as those nodes responding to $k-1$ consecutive primal links. The other nodes in the tree $\overline{G}_i^g$ can be defined as local nodes denoted by $LOCAL(\overline{G}_i^g) = \overline{N}_i \,/\, BORDER(\overline{G}_i^g)$. In the example shown in Fig. 3b, nine nodes (denoted in gray) are classified as border nodes, since they correspond to primal paths in Fig. 4.3a with two links ($\lambda = k-1 = 2$). These nine border nodes can be pushed into the top hierarchy as nodes in Fig. 4.3c. The other nodes ($\lambda = 0,1$) in Fig. 4.3b are classified as local nodes without consideration in the top hierarchy.

The top hierarchy $H^t \in HG$ has only one network, denoted by $\hat{G}^t = \left(\hat{\mathrm{N}}, \hat{\mathrm{A}}, \hat{\Psi}\right)$. As mentioned, all top hierarchical nodes are the border nodes from the ground hierarchy, denoted as $\hat{\mathrm{N}} = \{BORDER(\overline{G}_i^g), \forall \overline{G}_i^g\}$. Each top hierarchical link $\hat{a}_{iw,u} \in \hat{A}$ represents a primal path $p_u^{iw,k} = \{a_{ij}^1, ..., a_{\ell w}^k\}$ with k consecutive links. For the top

hierarchical link $\hat{a}_{iw,u}$, its tail node $\overline{n}_u^{i\ell,k-1}$ and head node $\overline{n}_u^{jw,k-1}$ respectively represent the first and last $k-1$ consecutive primal links of $p_u^{iw,k}$ (i.e. $p_u^{i\ell,k-1}$ and $p_u^{jw,k-1}$). Each top hierarchical movement $\hat{\psi}_{iq,u}=\{\hat{a}_{iw,u},\hat{a}_{jq,u}\}\in\hat{\Psi}$ corresponds to a primal path $p_u^{iq,k+1}=\{a_{ij}^1,...a_{\ell w}^k,a_{wq}^{k+1}\}$ passing $k+1$ consecutive links. Similarly, this top hierarchical movement's tail link $\hat{a}_{iw,u}$ and head link $\hat{a}_{jq,u}$ respectively represent the first and last k consecutive primal links of $p_u^{iq,k+1}$ (i.e. $p_u^{iw,k}$ and $p_u^{jq,k}$).

Fig. 4.3c depicts the construction of the top hierarchy, where $k=3$, from the same primal network in Fig. 4.3a. As shown in Fig. 4.3c, all nodes of top hierarchical networks are from the border nodes in the ground hierarchy. It can also be found from Fig. 4.3 that the top hierarchical network consists of links and movements which respectively represent the paths in Fig. 4.3a with three and four primal links. For instance, a top hierarchical movement $\hat{\psi}_{ABGH}=\{\hat{a}_{ABG},\hat{a}_{BGH}\}$ in Fig. 4.3c, corresponds to the primal path $p_1^{17,4}=a_A\oplus a_B\oplus a_G\oplus a_H$ in Fig. 4.3a. This movement's tail link $\hat{a}_{ABG}$ represents the primal path $p_1^{16,3}=a_A\oplus a_B\oplus a_G$ in Fig. 4.3a. It can be observed from Fig. 4.3 that all acyclic primal paths ($k\le 4$) in Fig. 4.3a are represented in the hierarchical network (Fig. 4.3b and Fig. 4.3c). Accordingly, this hierarchical network can be said to be a $k=3$ complete dual network of the primal network in Fig. 4.3a. The concept of the k complete dual network can be formally defined as follows.

Definition 4.1. A two-level hierarchical network $HG=\left(H^g,H^t\right)$ is a k complete dual network of the primal network G if and only if: (i) any acyclic path $p_u^{ij,\lambda}\in G$ with $\lambda\le k-1$ links has a corresponding node $\overline{n}_u^{ij,\lambda}\in H^g$; (ii) any acyclic path $p_u^{iw,k}\in G$ with k links has a corresponding link $\hat{a}_{iw,u}\in H^t$; and (iii) any acyclic path $p_u^{iq,k+1}\in G$ with $k+1$ links has a corresponding movement $\hat{\psi}_{iq,u}\in H^t$.

In the proposed hierarchical network, travel times and their correlations are stored in both ground and top hierarchies. In the ground hierarchy $H^g \in HG$, each node $\bar{n}_u^{ij,\lambda}$ maintains a travel time distribution $T_u^{ij,\lambda}$ for its corresponding primal path $p_u^{ij,\lambda} \in G$. In the top hierarchy $H^t \in HG$, each link $\hat{a}_{iw,u} \in H^t$ has a link travel time distribution $T_{iw,u}^{\hat{a}}$. Let $t_{iw,u}^{\hat{a}}$ and $(\sigma_{iw,u}^{\hat{a}})^2$ be mean and variance of $T_{iw,u}^{\hat{a}}$ respectively. They can be calculated from the primal network as

$$t_{iw,u}^{\hat{a}} = t^k \tag{4.9}$$

$$(\sigma_{iw,u}^{\hat{a}})^2 = (\sigma^k)^2 + \sum_{n=1}^{k-1} 2\,\mathrm{cov}(a^n, a^k) \tag{4.10}$$

where a^n and a^k are n$^{\text{th}}$ and k$^{\text{th}}$ links in the corresponding primal path $p_u^{iw,k} \in G$ respectively. The travel time covariance between two adjacent top hierarchical links $\hat{a}_{iw,u}$ and $\hat{a}_{jq,u}$ is stored as an attribute of the corresponding top hierarchical movement $\hat{\psi}_{iq,u} = \{\hat{a}_{iw,u}, \hat{a}_{jq,u}\}$. This travel time covariance $\mathrm{cov}(\hat{T}_{iw,u}, \hat{T}_{jq,u})$ can be expressed as

$$\mathrm{cov}(\hat{T}_{iw,u}, \hat{T}_{jq,u}) = \mathrm{cov}(T^1, T^{k+1}) \tag{4.11}$$

where T^1 and T^{k+1} travel times of the first link a^1 and the last link a^{k+1} of the movement's corresponding primal path $p_u^{iq,k+1} \in G$ respectively.

Let $p_u^{rs,\lambda+k-1} = \{a_{ri}^1, ..., a_{qs}^{\lambda+k-1}\} \in G$ be a primal path passing $\lambda + k - 1$ consecutive links. Obviously, the primal path $p_u^{rs,\lambda+k-1}$ contains λ primal sub-paths $\{p_u^{ri,k}, ..., p_u^{jw,k}, p_u^{\ell q,k}, ..., p_u^{\theta s,k}\}$ with consecutive k links. These λ primal sub-paths correspond to λ top hierarchical links $\{\hat{a}_{ri,u}^1, ..., \hat{a}_{jw,u}^{m-1}, \hat{a}_{\ell q,u}^m, ..., \hat{a}_{\theta s,u}^{\lambda}\}$ in the top hierarchy. Let $\hat{p}_u^{rs,\lambda}$ be the top hierarchical path consisting of these λ top hierarchical links. Its mean and travel time variance, denoted by $t_{rs,u}^{\hat{p}}$ and $(\sigma_{rs,u}^{\hat{p}})^2$, respectively are defined in this chapter as

$$t_{rs,u}^{\hat{p}} = t_u^{ri,k-1} + \sum_{m=1}^{\lambda} t_{\ell q,u}^{\hat{a},m} \tag{4.12}$$

$$(\sigma_{rs,u}^{\hat{p}})^2 = (\sigma_u^{ri,k-1})^2 + \sum_{m=1}^{\lambda}(\sigma_{\ell q,u}^{\hat{a},m})^2 + \sum_{m=2}^{\lambda} 2\,\mathrm{cov}(\hat{T}_{jw,u}^{m-1}, \hat{T}_{\ell q,u}^{m}) \tag{4.13}$$

where $t_u^{ri,k-1}$ and $(\sigma_u^{ri,k-1})^2$ respectively are mean and travel time variance stored in the first node $\bar{n}_u^{ri,\lambda}$ of the top hierarchical path $\hat{p}_u^{rs,\lambda}$; and $\hat{a}_{jw,u}^{m-1}$ and $\hat{a}_{\ell q,u}^{m}$ are two adjacent links along the path $\hat{p}_u^{rs,\lambda}$. Using this setting, it can be proved that the top hierarchical path $\hat{p}_u^{rs,\lambda}$ and the primal path $p_u^{rs,\lambda+k-1}$ have an identical travel time distribution as below.

Proposition 4.5. Given a top hierarchical path $\forall \hat{p}_u^{rs,\lambda} \in H^t$, its travel time distribution is equivalent to that of the corresponding primal path $p_u^{rs,\lambda+k-1} \in G$.

Proof. See Appendix B. □

A simple path in Fig. 4.3 can be used to illustrate the Proposition 4.5. A top hierarchical path $\hat{p}_1^{18,3} = \hat{a}_{ABG} \oplus \hat{a}_{BGH} \oplus \hat{a}_{GHI} \in H^t$ in Fig. 4.3c corresponds to a primal path $p_1^{18,6} = a_A \oplus a_B \oplus a_G \oplus a_H \oplus a_I \in G$ in Fig. 4.3a. The mean travel time of $\hat{p}_1^{18,3}$ is equivalent to that of $p_1^{18,6}$ as $t_{18,1}^{\hat{p}} = t_1^{15,2} + (t_{16,1}^{\hat{a},1} + t_{27,1}^{\hat{a},2} + t_{58,1}^{\hat{a},3}) = t_A + t_B + (t_G + t_H + t_I) = t_1^{18,5}$. The paths $\hat{p}_1^{18,3}$ and $p_1^{18,6}$ also have the same travel time variance as follows:

$$\begin{aligned}
(\sigma_{18,1}^{\hat{p}})^2 &= (\sigma_1^{15,2})^2 + (\sigma_{16,1}^{\hat{a},1})^2 + (\sigma_{27,1}^{\hat{a},2})^2 + (\sigma_{58,1}^{\hat{a},3})^2 + 2\,\mathrm{cov}(\hat{T}_{16,1}, \hat{T}_{27,1}) + 2\,\mathrm{cov}(\hat{T}_{17,1}, \hat{T}_{58,1}) \\
&= (\sigma_A^2 + \sigma_B^2 + 2\,\mathrm{cov}(T_A, T_B)) + (\sigma_G^2 + 2\,\mathrm{cov}(T_A, T_G) + 2\,\mathrm{cov}(T_B, T_G)) \\
&\quad + (\sigma_H^2 + 2\,\mathrm{cov}(T_B, T_H) + 2\,\mathrm{cov}(T_G, T_H)) + (\sigma_I^2 + 2\,\mathrm{cov}(T_G, T_I) + 2\,\mathrm{cov}(T_H, T_I)) \\
&\quad + (2\,\mathrm{cov}(T_A, T_H)) + (2\,\mathrm{cov}(T_B, T_I)) \\
&= (\sigma_1^{18,5})^2.
\end{aligned}$$

With Proposition 4.5, following two important lemmas hold:

Lemma 4.1. A path extension $\hat{p}_u^{rq,m} = \hat{p}_u^{rw,m-1} \oplus \hat{a}_{\ell q,u}$ in top hierarchy is equivalent to the path extension $p_u^{rq,m+k-1} = p_u^{rw,m+k-2} \oplus a_{wq}$ in the primal network, where top hierarchical path $\hat{p}_u^{rw,m-1} \in H^t$ corresponds to the primal path $p_u^{rw,m+k-2} \in G$; top

hierarchical link $\hat{a}_{\ell q,u} \in H^t$ corresponds to the primal path $p_u^{\ell q,k} \in G$; and primal link $a_{wq} \in G$ is the last link of the primal path $p_u^{\ell q,k}$.

Proof. It can be easily followed by Proposition 4.5.

Lemma 4.2. Given a path $\forall p_u^{rs,\lambda} \in G$, it can be determined either a node $\overline{n}_u^{rs,\lambda} \in H^g$ or a path $\hat{p}_u^{rs,\lambda-k+1} \in H^t$ with the same travel time distribution as $p_u^{rs,\lambda} \in G$, if the hierarchical network $HG = \left(H^g, H^t\right)$ is a k complete dual network of the primal network G.

Proof. When $\lambda \le k-1$, according to Definition 4.1, there exist a node $\overline{n}_u^{rs,\lambda} \in H^g$ representing the path $p_u^{rs,\lambda}$. According to the definition of a hierarchical network, the attributes stored at the node $\overline{n}_u^{rs,\lambda}$ are equivalent to travel time of $p_u^{rs,\lambda}$.

When $\lambda > k-1$, $p_u^{rs,\lambda}$ contains $\lambda-k+1$ sub-paths with consecutive k links. According to Definition 4.1, all these sub-paths, with consecutive *k* links, in the primal network are represented as links in the top hierarchy. Such top hierarchical links form a path $\hat{p}_u^{rs,\lambda-k+1}$. According to Proposition 4.5, $\hat{p}_u^{rs,\lambda-k+1}$ and $p_u^{rs,\lambda}$ have an identical path travel time distribution. □

With the above two lemmas, SD-RSPP in the primal network can likewise be solved in the proposed two-level hierarchical network using the generalized dynamic programming approach. As each top hierarchical link represents a primal path with *k* consecutive links, the non-dominated paths passing the same *k* consecutive links (according to Propositions 4.2 and 4.3) can be directly maintained and evaluated at each top hierarchical link. In addition, as the path extension in the top hierarchy is equivalent to the path extension in the primal network, the monotonic increasing property (Proposition 4.3) and the acyclic property (Proposition 4.4) are also satisfied in the top hierarchical network. These properties contribute to the development of efficient solution algorithms given in the next section.

4.5. Solution Algorithm

In this section, the label-selection multi-criteria A* algorithm, presented in Chapter 3, is extended to solve SD-RSPP in the proposed hierarchical network. For convenience, the algorithm is hereafter referred to as *SDRSP-HA**. Unless otherwise stated, the hierarchical network $HG = \left(H^g, H^t\right)$ used hereafter is a *k* complete dual network of the primal network. Similar to multi-criteria A* algorithm presented in Chapter 3, the *SDRSP-HA** algorithm uses a heuristic valuation function $F(\hat{p}_u^{rj}) = \Phi^{-1}_{\hat{T}_u^{rj}}(\alpha) + h(j)$ as a label for the top hierarchical path $\hat{p}_u^{rj} \in H^t$, where $h(j)$ is a travel time budget estimate from node $j \in G$ to destination $s \in G$, and $h(s) = 0$ at the destination.

As indicted above, SD-RSPP in the primal network can equally be solved in the proposed two-level hierarchical network. Each top hierarchical link $\hat{a}_{ij} \in H^t$ represents a primal path $p^{ij,k} \in G$ with k consecutive links. Thus, M-V non-dominated paths $p_u^{rj} = p_u^{ri} \oplus p_u^{ij,k} \in G$ passing the same $p_u^{ij,k}$ can be represented as corresponding top hierarchical paths $\hat{p}_u^{rj} = \hat{p}_u^{ri} \oplus \hat{a}_{ij} \in H^t$. Let $\hat{P}^{rj} = \{\hat{p}_u^{rj}, ..., \hat{p}_v^{rj}\}$ be a set of non-dominated paths maintained at the top hierarchical link $\hat{a}^{ij}$. The non-dominated paths in $\hat{P}^{rj}$ are sorted in ascending order by mean travel time $t_{rj,u}^{\hat{p}}$. Non-dominated paths from all top hierarchical links are maintained in a scan eligible set, denoted by $SE = \{\hat{p}_u^{rj}, ..., \hat{p}_v^{rw}\}$. The non-dominated paths in SE are ordered by increasing value of the heuristic function, $F(\hat{p}_u^{rj})$.

At each iteration, non-dominated path $\hat{p}_u^{rj}$ at the top of *SE* (with minimum $F(\hat{p}_u^{rj})$) is selected from *SE* for path extensions. A temporary acyclic path is constructed by extending the selected path $\hat{p}_u^{rj}$ to its successor link $\hat{a}_{qw} \in H^t$, denoted by $\hat{p}_u^{rw} := \hat{p}_u^{rj} \oplus \hat{a}_{qw}$. The dominant relationship between the newly generated path $\hat{p}_u^{rw}$ and the set of non-dominated paths $\hat{P}^{rw}$ at link $\hat{a}_{qw}$ is determined using the M-V dominance condition (Proposition 4.2). If $\hat{p}_u^{rw}$ is a M-V

non-dominated path at link $\hat{a}_{qw}$, it is then inserted into $\hat{P}^{rw}$ and SE. The newly generated path $\hat{p}_u^{rw}$ may also dominate a set of paths in $\hat{P}^{rw}$, denoted by $\hat{P}_D^{rw}$. These dominated paths in $\hat{P}_D^{rw}$ can be eliminated from $\hat{P}^{rw}$ and SE. The algorithm continues this path search process until the destination is reached or SE becomes empty. The steps of *SDRSP-HA** algorithm are given below.

Algorithm: SDRSP-HA*

Inputs: O-D nodes (r, s) and on-time arrival probability α

Returns: the reliable shortest path

Step 1. Initialization:

For each border node $\overline{n}_u^{ri,k-1} \in \overline{G}_r$ ($\overline{G}_r \in H^g$ denotes the tree rooted at origin r)

For each top hierarchical link $\hat{a}_{ij}$ emanating from node $\overline{n}^{ri,k-1}$

Generate a new path $\hat{p}_u^{rj} := \overline{n}_u^{ri,k-1} \oplus \hat{a}_{ij}$ and calculate $h(j)$ and $F(\hat{p}_u^{rj})$.

Set $\hat{P}^{rj} := \{\hat{p}_u^{rj}\}$ and $SE := SE \cup \{\hat{p}_u^{rj}\}$.

End for

End for

If destination $s \in \overline{G}_r$, then, insert all paths $p_u^{rs} \in \overline{G}_r$ into SE.

Step 2. Path selection:

If $SE = \phi$, then Stop; otherwise, continue.

Select $\hat{p}_u^{rj}$ at the top of SE and set $SE := SE \setminus \{\hat{p}_u^{rj}\}$.

If $j = s$, then Stop; otherwise continue.

Step 3. Path extension:

For every movement $\hat{\psi}_{ijw} = \{\hat{a}_{ij,u}, \hat{a}_{qw,u}\}$ ($\hat{a}_{ij,u}$ denotes the last link of $\hat{p}_u^{rj}$)

Generate a new path $\hat{p}_u^{rw} := \hat{p}_u^{rj} \oplus \hat{a}_{qw}$ and calculate $h(w)$ and $F(\hat{p}_u^{rw})$.

If p_u^{rw} is acyclic, then continue; otherwise, scan next movement.

Call procedure $\hat{P}_D^{rw} := CheckDominance(\hat{p}_u^{rw}, \hat{P}^{rw})$.

If $\hat{p}_u^{rw}$ is a non-dominated path, then set $SE := SE \cup \{\hat{p}_u^{rw}\}$ and $SE := SE \setminus \hat{P}_D^{rw}$.

End for

Go to Step 2.

Procedure: *CheckDominance*

Inputs: A path $\hat{p}_u^{rj}$ and a set of non-dominated path $\hat{P}^{rj}$

Returns: $\hat{P}_D^{rj}$ storing the set of paths dominated by $\hat{p}_u^{rj}$, and updated $\hat{P}^{rj}$

Step 1: Initialization

If $\alpha > 0.5$, then $\beta := 0.999$.

If $\alpha = 0.5$, then $\beta := 0.5$.

If $\alpha < 0.5$, then $\beta := 0.001$.

Set $\hat{P}_D^{rj} := \phi$ and $v := 1$.

Step 2: Dominant relationship determination

While $v \leq | \hat{P}_D^{rj} |$ and $t_u^{rj} > t_v^{rj}$ ($| \hat{P}_D^{rj} |$ is the number of paths in $\hat{P}_D^{rj}$)

If $\Phi_{\hat{T}_u^{rj}}^{-1}(\beta) > \Phi_{\hat{T}_v^{rj}}^{-1}(\beta)$, then return $\hat{P}_D^{rj}$.

Set $v := v + 1$.

End while

If $t_u^{rj} = t_v^{rj}$ and $\Phi_{\hat{T}_u^{rj}}^{-1}(\beta) > \Phi_{\hat{T}_v^{rj}}^{-1}(\beta)$, then return $\hat{P}_D^{rj}$.

Insert $\hat{p}_u^{rj}$ into $\hat{P}^{rj}$ at v th position and set $v := v + 1$ (by default $| \hat{P}_D^{rj} | := | \hat{P}_D^{rj} | + 1$).

While $v \leq | \hat{P}_D^{rj} |$ and $\Phi_{\hat{T}_u^{rj}}^{-1}(\beta) \leq \Phi_{\hat{T}_v^{rj}}^{-1}(\beta)$

Set $\hat{P}^{rj} := \hat{P}^{rj} \setminus \{\hat{p}_v^{rj}\}$ and $\hat{P}_D^{rj} := \hat{P}_D^{rj} \cup \{\hat{p}_v^{rj}\}$.

Set $v := v + 1$.

End while

Return $\hat{P}_D^{rj}$.

The heuristic function $F(\hat{p}_u^{rj})$ is admissible if the following inequality is satisfied

$$F(\hat{p}_u^{rj} \oplus \hat{a}_{qw,u}) = \Phi_{\hat{T}_u^{rw}}^{-1}(\beta) + h(w) \geq F(\hat{p}_u^{rj}) = \Phi_{\hat{T}_u^{rj}}^{-1}(\alpha) + h(j) \tag{4.14}$$

Eq. (4.14) indicates that the heuristic function value of $F(\hat{p}_u^{rj})$, should monotonically increase with path extensions. If the heuristic function is admissible, it

can be proved that the *SDRSP-HA** algorithm can obtain the optimal solution for the SD-RSPP as follows.

Proposition 4.6. If the heuristic function used is admissible, the *SDRSP-HA** algorithm can determine the reliable shortest path when the destination node is reached.

Proof. Let $P^{rs} \in G$ be the set of paths containing all non-dominated paths between O-D nodes. When destination node is reached, the path $\hat{p}_*^{rs}$ is selected from SE. The selected path $\hat{p}_*^{rs}$ can be either a node in the ground hierarchy or a path $\hat{p}_u^{rs} \in H^t$ in the hierarchical network. As at each iteration the path with minimum $F(\hat{p}_u^{rj})$ was selected from SE, the heuristic function value of $\hat{p}_*^{rs}$ ($F(\hat{p}_*^{rs})$) is the minimum heuristic function value among all paths in SE. Since all paths in $\tilde{P}^{rs} = P^{rs} \setminus \{\hat{p}_*^{rs}\}$ are extended from SE and the heuristic function value monotonically increases with path extensions, the heuristic function value of $\hat{p}_*^{rs}$ is less than that of any path in $\tilde{P}^{rs}$. As $h(s) = 0$, $F(\hat{p}_*^{rs}) = \Phi_{\hat{T}_u^{rs}}^{-1}(\beta)$ is the minimum travel time budget in P^{rs} and thus $\hat{p}_*^{rs}$ is the reliable shortest path between O-D nodes. □

The performance of *SDRSP-HA** algorithm depends on the quality of $h(j)$ used. The better the travel time budget $h(j)$ estimates, the better the computational performance of the *SDRSP-HA** algorithm. When $h(j) = 0$, the *SDRSP-HA** algorithm reduces to the label-setting algorithm which uses travel time budget $\Phi_{\hat{T}_u^{rj}}^{-1}(\alpha)$ as the heuristic function value for path $\hat{p}_u^{rj}$ instead of $F(\hat{p}_u^{rj})$. With the implementation of SE using F-heap data structure (Fredman and Tarjan, 1987), in worse case the label-setting algorithm requires $O\left(|\hat{\Psi}||\hat{P}|^2 + |\hat{A}||\hat{P}|Log(|\hat{A}||\hat{P}|)\right)$, where $|\hat{A}|$ and $|\hat{\Psi}|$ are the number of links and movements in the top hierarchical network, and $|\hat{P}|$ is maximum number of non-dominated paths associated with a top hierarchical link.

4.6. Case Study in Hong Kong

A real-world case study, to demonstrate the applicability of the proposed solution algorithm, is described in this section. In Hong Kong, real-time traffic information on major urban roads is provided by a Real-time Travel Information System (RTIS) (http://tis.td.gov.hk/rtis/ttis/index/main_partial.jsp) (Tam and Lam, 2008). In RTIS, Radio-frequency identification technology is adopted to collect real-time traffic data. Offline link travel times and variance-covariance matrices, generated by traffic flow simulators (Lam et al., 2002), are also adopted for RTIS. With the use of real-time and offline traffic data, RTIS can provide travel time estimates every five minutes for both links, either with or without real-time data.

As shown in Fig. 4.4, the RTIS network consists of 1,367 nodes, 3,655 links and 11,849 movements at road intersections. In this study, the RTIS data (including mean and variance-covariance matrix of link travel times) were collected at an off-peak hour (05:00-06:00) and a morning peak hour (08:00-09:00) on 23 Sep 2010 (Thursday).

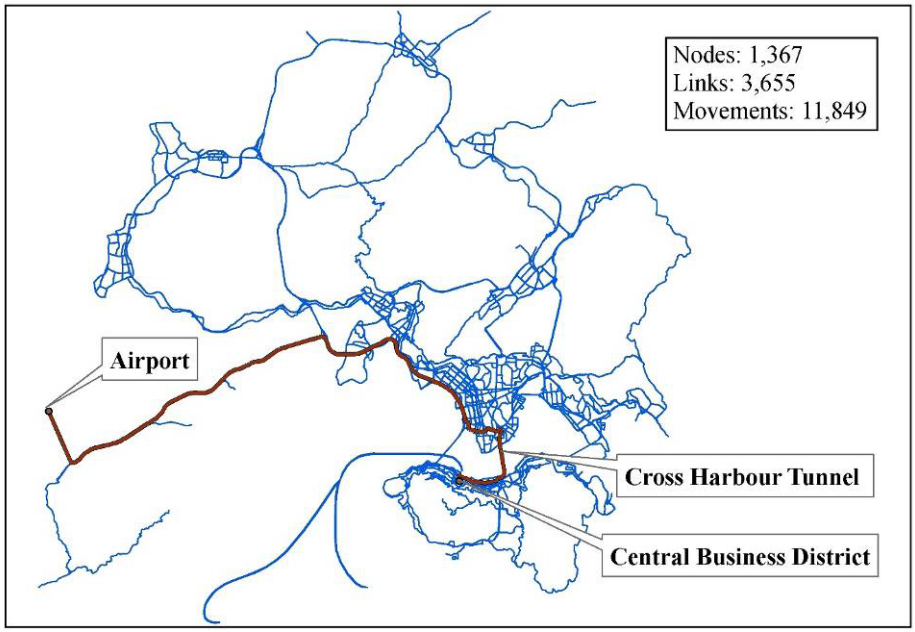


Figure 4.4 Hong Kong RTIS network

To normalize the link travel time covariance in the whole network, a correlation

coefficient for every two links , a_{ij} and a_{qw}, was calculated as

$$\rho_{ij}^{qw} = \mathrm{cov}(a_{ij}, a_{qw}) / \sigma_{ij}\sigma_{qw} \tag{4.15}$$

The value of ρ_{ij}^{qw} is between -1 and $+1$; $\rho_{ij}^{qw} = +1$ is the case of perfect positive correlation and $\rho_{ij}^{qw} = -1$ is the case of perfect negative correlation.

Fig. 4.5 gives the correlation coefficients of all adjacent RTIS links (k = 1) for both peak and off-peak hours. As shown in the figure, travel times between adjacent links are correlated either positively or negatively. It can be observed from the figure that about two thirds of link travel time was positively correlated. It is also seen from the same figure that a considerable number of links were strongly correlated. As shown, 68.8% and 48.3% of absolute values of these correlation coefficients were larger than 0.1 for peak and off-peak hours respectively. Therefore, the common assumption that link travel times are independently distributed may be erroneous for both peak and off-peak hours.

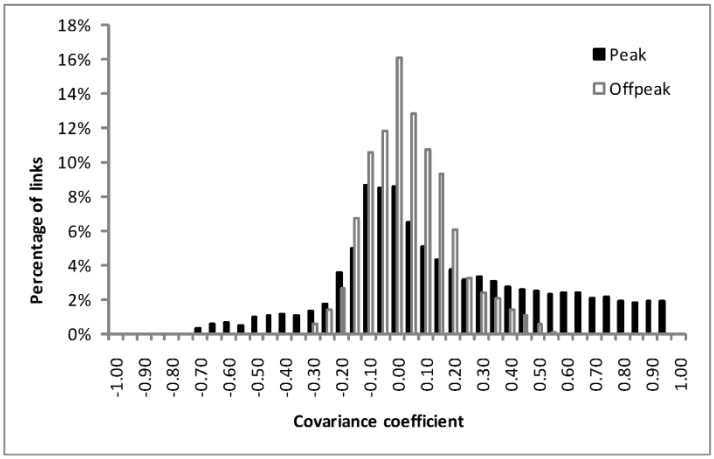


Figure 4.5 Histogram of correlation coefficient between adjacent links

The spatial correlations among k-neighboring links are also investigated. Let NL_{ij}^{k} the set of k-neighboring links for link a_{ij}. The mean absolute value (MAV) of correlation coefficients for all k-neighboring links, denoted by $E(\rho^{k})$, can be calculated as

$$E(\rho^k) = \sum_{a_{ij} \in A} abs(p_{ij}^{qw}) / \sum_{a_{ij} \in A} | NL_{ij}^k |, \quad \forall a_{qw} \in NL_{ij}^k \tag{4.16}$$

where $abs(p_{ij}^{qw})$ is the absolute value of ρ_{ij}^{qw} for links a_{ij} and a_{qw}; and $| NL_{ij}^k |$ is the number of *k*-neighboring links for link a_{ij}. The value of $E(\rho^k)$ is between 0 and +1. The larger the $E(\rho^k)$ value, the stronger the travel time correlations among *k*-neighboring links. The $E(\rho^k)$ value can be adopted as an indicator to measure travel time correlations among *k*-neighboring links.

Fig. 4.6 shows such $E(\rho^k)$ values for the RTIS network during the peak hour. In this figure, the x-axis refers to k values; and the y-axis at the left hand side represents $E(\rho^k)$ values. It is seen from Fig. 4.6 that with the increase of the *k* value the travel time correlations significantly decrease. For instance, when the *k* value increases from 1 to 4, $E(\rho^k)$ decreases from 0.29 to 0.04, a reduction of about 86.2%. This observation is consistent with the empirical findings of previous studies (Gajewski and Rilett, 2003): the travel time correlation is usually very low for links that are spatially distant.

From the above observation, the *k* limited spatial dependence assumption, that the travel time of a link correlates only with its neighboring links within a local impact area, seems valid. To justify this assumption, the approximation accuracy of path travel time standard deviation (SD) was examined using a typical cross harbour journey from the central business district (CBD) to Hong Kong International airport (HKIA). As shown in Fig. 4.4, the route passing the cross harbour tunnel (CHT) was selected for this case study, because CHT proved to be the most frequently used tunnel in Hong Kong and also, and not surprisingly, had a large travel time variation.

Fig. 4.6 illustrates this SD approximation accuracy under different k values during the peak hour (refers to the y-axis at right hand side). The actual path travel time SD was 11.64 minutes. It can be found from Fig. 4.6 that the path travel time SD can be underestimated by 26.5% when link travel time correlations are ignored (k = 0). This

SD approximation accuracy can be improved by increasing the *k* value. For example, the approximation accuracy of path travel time SD can be significantly improved to 87.1%, when travel time correlations among adjacent links (*k* = 1) are considered. This SD approximation accuracy can be further improved to 99.1% when the *k* value increases to 4. Therefore, for this case study, the path travel time SD can be well approximated by using the k limited spatial dependence assumption with a sufficiently large k value (e.g. *k* = 4).

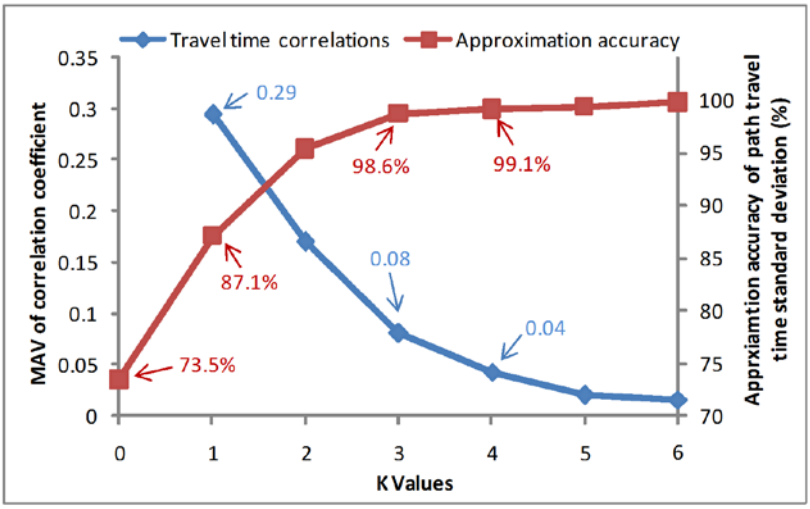


Figure 4.6 Travel time correlations among *k*-neighboring links

Table 4.1 gives the sizes of the impact area under different *k* values. The impact area size is measured by the average number of links, denoted by $E(|A_{ij}^k|)$. It can be calculated as

$$E(|A_{ij}^k|) = \sum_{a_{ij} \in A} |A_{ij}^k| / |A| \tag{18}$$

where $|A_{ij}^k|$ is the number of links of the impact area G_{ij}^k for a primal link $a_{ij} \in G$; and $|A|$ is the number of links in the primal network G. It can be observed from Table 4.1 that the size of the impact area exponentially increases with *k* values. For instance, when *k* increases from 1 to 4, the $|A_{ij}^k|$ value grows by 27 times, from 3.24 to 87.53. This result can be used to interpret the validation of the *k* limited spatial dependence assumption. When *k* is large enough, the impact area can maintain a considerable number of links with correlated travel times; and thus the

majority of link travel time correlations, along the path, can be captured within that impact area.

The proposed two-level hierarchical network was constructed using different *k* values (see Table 4.1). It can be seen from Table 4.1 that the size of a constructed hierarchical network also exponentially increases with *k* values, similar to the size of impact area. For instance, when *k* increases from 1 to 4, the number of links in the top hierarchical network grows by 29 times from 3,655 to 106,942. This growth rate is close to that of the impact area size (about 27 times). Therefore, the increase of the *k* value can improve the approximation accuracy of path travel time distributions, but at the cost of increasing the SD-RSPP problem size. It should be noted that when $k = 1$ (only correlations among adjacent links are considered), the primal network G can be used directly for solving SD-RSPP. The travel time correlations among adjacent links can be maintained as an attribute of each movement in the primal network.

Table 4.1 The sizes of impact area and hierarchical network under different *k* values

K value	Impact area (Links)	Ground hierarchy (Nodes)	Top hierarchy		
			Nodes	Links	Movements
1	**3.24**	-	1,367	**3,655**	11,849
2	13.59	5,022	3,655	11,849	34,555
3	38.16	16,871	11,849	34,555	106,942
4	***87.53***	51,426	34,555	***106,942***	315,172
5	174.63	157,001	106,942	315,172	958,541

The computational performance of the proposed *SDRSP-HA** algorithm, using different *k* values was tested on the RTIS network. The *SDRSP-HA** algorithm was coded in Visual C# programming language. The priority queue (SE) was implemented using the F-heap data structure (Fredman and Tarjan, 1987). The $h(j)$ used was the Euclidean distance function. All experiments were conducted in the computer with a four-core Intel Xeon 3.2GHz CPU (only one core was used) and a Windows Server 2003 operation system.

Reported in Table 4.2 is the computational performance of the proposed *SDRSP-HA** algorithm using different k values. Three risk-taking scenarios, including risk-averse ($\alpha = 0.9$), risk-seeking ($\alpha = 0.1$) and risk-neutral ($\alpha = 0.5$), were tested for each k value. All reported results were the average of 100 computer runs, using different O-D nodes in each run. The 100 O-D nodes were randomly selected and the same O-D node set was used for every test performed on all hierarchical networks.

Table 4.2 Computation performance of *SDRSP-HA** algorithm

K value	Risk-averse ($\alpha = 0.9$)		Risk-seeking ($\alpha = 0.1$)		Risk-neutral ($\alpha = 0.5$)	
	$\tilde{t}$	$\lvert \hat{p} \rvert$	$\tilde{t}$	$\lvert \hat{p} \rvert$	$\tilde{t}$	$\lvert \hat{p} \rvert$
1	0.017	2.327	0.020	2.458	0.007	1.325
2	0.391	13.286	0.409	12.928	0.044	3.810
3	2.403	35.760	4.324	40.211	0. 213	10.467
4	**79.085**	103.514	126.235	109.865	**1.365**	29.169

$\tilde{t}$: Average computational time (seconds)

$\lvert \hat{p} \rvert$: Average number of non-dominated paths in the top hierarchical network (10^3)

It is clearly seen from Table 4.2 that the computational time of the *SDRSP-HA** algorithm exponentially increases with respect to the k values. The *SDRSP-HA** algorithm runs much faster at the risk-neutral scenario at the other two risk-taking scenarios. For example, when k = 4, the computational time required by a risk-averse scenario is about 58 times (79.085/1.365) larger than that required by the risk-neutral scenario. This is because when travellers are risk-neutral, the *SDRSP-HA** algorithm becomes, essentially, a traditional A* algorithm. In this case, the least number of non-dominated paths is generated in the search process, since only a single path is kept at each top hierarchical link. For the risk-averse and risk-seeking scenarios, additional computational effort is required to generate a considerable amount of M-V non-dominated paths during the search process. It can also be found from Table 4.2 that the algorithm runs slightly faster (1.39 times faster when k = 4) in the risk-averse scenario than the risk-seeking scenario. This is positive information for the majority

of travellers whose major concern is late arrival.

4.7. Summary

In this chapter, the spatial-dependent reliable shortest path-finding problem (SD-RSPP) has been investigated taking into account travel time spatial correlations. The travel time of a link is assumed to be spatially correlated only with the neighboring links within a local impact area. The travel time correlations with neighboring links are considered and represented by variance-covariance matrices. Based on this assumption, the SD-RSPP was formulated and solved as a multi-criteria shortest path-finding problem. The two dominance conditions, first-order stochastic dominant and M-V dominance conditions presented in Chapter 3, were extended to determine dominated paths for the SD-RSPP.

Based on the established dominance conditions, a new multi-criteria A* algorithm was proposed to solve the SD-RSPP in an equivalent two-level hierarchical network, exactly. The proposed hierarchical network can well represent travel time correlations among neighboring links. The optimality of the proposed solution algorithm has been proved theoretically and the complexity of the proposed algorithm has also been analyzed.

In this chapter, a case study using data from the real-time travel information system (RTIS) in Hong Kong was carried out. Travel time data collected from RTIS demonstrated that link travel times are, indeed, strongly correlated within local impact areas. This finding indicated the common assumption that link travel times are independently distributed was erroneous. Numerical results also showed that the proposed limited spatial dependence assumption can well approximate path travel time standard deviation, if the size of the impact area is sufficiently large. Computational results indicated that the increase of the impact area size can significantly enhance the accuracy of reliable shortest path findings, but at the cost of additional computational time. Therefore, it is necessary to consider the trade-off between the accuracy and computational performance of the solution algorithm.

It should be noted that the M-B dominance presented in Chapter 3 can also be extended to determine dominated paths for the SD-RSPP. The M-B dominance, however, requires paths passing the same sub-path with $2k$ subsequent links. Consequently, this M-B dominance may not easily be used for solving SD-RSPP. The extended M-B dominance is given in Appendix B for reference.

It is assumed in this chapter that link travel times are stable for a relatively long time period (e.g. morning peak hours). In the following chapter, Chapter 5, the reliable shortest path problems are extended to stochastic time-dependent networks where link travel times vary with the time-of-day.

5. Time-dependent Reliable Shortest Path Problems

In last two chapters, the development of efficient solution algorithms, for solving reliable shortest path problems in stochastic stationary networks has been described. In this chapter, such reliable shortest path problems are extended to stochastic time-dependent (STD) networks where the statistical distributions of link travel times vary by time intervals throughout the day. For the convenience of the reader, the associated problems are hereafter referred to as the time-dependent reliable shortest path problems (TD-RSPP).

The problems of finding the reliable shortest path in STD networks become more complex when incorporating the time-dimension. Firstly, the first order stochastic dominant (FSD) condition presented in last two chapters may not be satisfied in TD-RSPP, when link travel times are not stochastic first in first out (S-FIFO). In such cases, the efficient multi-criteria A* algorithms, presented in last two chapters, are not applicable for solving TD-RSPP.

Secondly, temporal correlations on link travel times have to be considered in TD-RSPP. Fu and Rilett (1998) noted that the path travel time distributions in STD networks are a stochastic process, conditionally depending on arrival times at intermediate nodes and link travel times along the path. The generation of such complex path travel time distributions is mathematically intractable (Wu and Nie, 2009).

Thirdly, a reversibility property does not hold in TD-RSPP. TD-RSPP can be classified as two problems according to different routing scenarios. The first problem relates to the necessity to determine the earliest arrival time and associated reliable shortest path for a given departure time (referred as a forward TD-RSPP). The second problem relates to determining the latest departure time and associated reliable shortest path for a given preferred arrival time (referred as a backward TD-RSPP). As path travel time distributions in STD networks depend on arrival

times at intermediate nodes, they must be generated by forward searches from origin to destinations. Therefore, the TD-RSPP is not reversible. The backward TD-RSPP cannot be easily solved using algorithms designed for the forward problem by a search from destination to origin.

The above three issues, involved in TD-RSPP, are addressed in this chapter. A new stochastic travel speed model (S-TSM) is proposed to represent STD link travel times. It has been proved that the link travel time distributions, under S-TSM, satisfy S-FIFO property. In addition, an effective approximation method to generate path travel time distribution is proposed in this chapter. The multiple-criteria A* algorithm, presented in the last two chapters, is further extended to exactly solve the forward TD-RSPP. The backward TD-RSPP is formulated and solved using an equivalent two-stage approach.

The remainder of this chapter is structured as follows. The previous studies related to TD-RSPP are briefly reviewed in Section 5.1. The mathematical formulation of TD-RSPP is shown in Section 5.2. The S-FIFO property and the representation of link travel time distribution are discussed in Section 5.3. Solution algorithms for solving both forward and backward TD-RSPP are presented in Section 5.4. Discrete methods for generating path travel time distribution are discussed in Section 5.5. A case study using a real-world advanced traveller information system (ATIS) is reported in Section 5.6. Computation performance of proposed solution algorithms in several large-scale networks are examined in Section 5.7. Finally, a summary of this chapter is given in Section 5.8.

5.1. Background

The problem of finding shortest paths in networks with deterministic time-dependent travel times has been intensively studied. Chabini and Lan (2002) noted that this time-dependent shortest path problem (TD-SPP) can be efficiently solved using A* algorithm (Hart et al., 1968) on a time-expanded network, when link travel times satisfy first in first out (FIFO) situations. Daganzo (2002) also pointed out that

TD-SPP is reversible in a network with FIFO link travel times. An algorithm that solves the forward TD-SPP can also solve the backward TD-SPP when link travel times in the network are FIFO consistent. In road networks, FIFO implies that two vehicles travelling on the same link will arrive at the end of the link in the order in which they entered it. In other words, overtaking is not allowed in the FIFO network. Although overtaking behaviours is common in real networks, this particular FIFO property seems valid in the context of route guidance systems, since vehicles are assumed to be travelling at the same average speed.

The majority of previous studies addressed TD-SPP based on the link travel time model (LTM). In LTM, travellers' experienced link travel time is dependent on the time instance at which they entered the link. It is assumed to be fixed when vehicles travelling on that link (Orda and Rom, 1990; Sung et al., 2000). Link travel times in the LTM, however, may not always satisfy FIFO property (Chabini, 1998). Orda and Rom (1990) illustrated that the shortest path, in the non-FIFO network, can have cycles and hence violate Bellman's principle of optimality (Bellman, 1958). In this case, efficient dynamic programming algorithms (e.g. Dijkstra's and A* algorithms) cannot be used for solving TD-SPP. Sherali et al. (1998) and Sherali and Hill (2009) showed that TD-SPP can be NP-hard, even if only one link in the network violates FIFO property. As an alternative travel time model, Sung et al. (2000) proposed a travel speed model (TSM), in which the travel speed of a link is assumed to be fixed within each time interval and travellers' experienced travel speeds can vary when they are travelling on a network link. It is shown that link travel times in the TSM are FIFO consistent and that the results of shortest path findings based on TSM are stable to variance of time interval length.

As noted in Chapter 2, travel times in congested urban road networks are highly stochastic due to random demand fluctuations and supply degradations. In such cases, link travel times in congested urban road networks should be represented as STD variables. In the literature, the stochastic link travel time model (S-LTM) is a commonly used model for the representation of such STD link travel times (Chang et al., 2005; Nie and Wu, 2009a). As a stochastic extension of LTM, link travel time distribution in S-TLM is dependent on the time instance of travellers link enter; and

the link travel time distribution is assumed to be fixed for and when travellers travel on the link. This model is simple for STD network representation. However, travel times in such model may not be S-FIFO (stochastic first-in-first-out). S-FIFO implies that, under the same probability level of traffic congestion, vehicles arrive at the end of a link in the order in which they entered the link. Nie and Wu (2009b) pointed out that the reliable shortest path in non-S-FIFO networks can have cycles and efficient labeling algorithms cannot be used for finding reliable shortest paths in such networks.

The generation of path travel time distribution is another challenge involved in the STD network. It was found that the path travel time distribution in STD networks is a stochastic process, conditionally depending on arrival times at intermediate nodes and successor link travel times along the path (Fu and Rilett, 1998). Thus, the generation of exact path travel time distribution is mathematically intractable and only approximation methods can be used (Fu and Rilett, 1998; Miller-Hooks and Mahmassani, 1998).

Miller-Hooks and Mahmassani (1998) proposed a discrete method to generate such path travel time distribution. In their proposed method, every arrival time and successor link travel time along the path are discretized into L equal-probability intervals to address the conditional probability involved in the generation of path travel time distribution. This approximation method can obtain accurate path time distribution when the value of L is sufficiently large (Wu and Nie, 2009). This approximation method, however, requires considerable computational effort to calculate and sort L^2 discrete elements for each path extension operations. Fu and Rilett (1998) proposed a Taylor series approximation method to estimate the mean and variance of path travel time. Chang et al. (2005) assumed path travel times following normal distributions and proposed a discrete method to estimate the mean and variance of the path travel time. This approximation method has a significant computational advantage over the discrete scheme proposed by Miller-Hooks and Mahmassani (1998). However, no precise notion exists regarding how good this approximation is for the generation of path travel time distribution in real networks.

5.2. Problem Statement

The model formulation of TD-RSPP is described in this section. For notational consistency, throughout the chapter, capital letters represent random variables and lower-case letters represent deterministic variables. Let $f(\cdot)$ be the probability density function (PDF), $\Phi(\cdot)$ the cumulative probability function (CDF), and $\Phi^{-1}(\cdot)$ the inverse CDF of a random variable.

Let $G = (N, A, \Omega)$ be a directed STD network, where N is the set of nodes, A is the set of links, and Ω is the period of interest. The period of interest Ω is considered as a set of discrete times $\{..., \Delta_n = n\Delta, ...\}$, where n is an integer and Δ is the time interval. Each node i has a set of successor nodes $SCS(i) = \{j : a_{ij} \in A\}$ and a set of predecessor nodes $PDS(i) = \{w : a_{wi} \in A\}$. Each link $a_{ij} \in A$ has a tail node $i \in N$, a head node $j \in N$ and a set of random travel speeds varying over the period of interest. V_{ij}^n denotes a random travel speed at time interval (Δ_{n-1}, Δ_n).

Let $T_{ij}(y)$ be a random travel time for vehicles entering the link a_{ij} at the time instant $y \in (\Delta_{n-1}, \Delta_n)$. Suppose that the nodes $r \in N$ and $s \in N$ represent the O-D nodes. Let y_r and y_s be the travellers' departure time and preferred arrival time respectively. The travel time budget that travellers assign for their journey can be calculated as

$$b = y_s - y_r \tag{5.1}$$

Let $p_u^{rs} \in P^{rs}$ be a path from origin r to destination s. $T_u^{rs}(y_r)$ denotes travel time of journey starting at departure time y_r going through the path p_u^{rs}. It is assumed that waiting is not allowed in the network except for the destination. $T_u^{rs}(y_r)$ can then be calculated as

$$T_u^{rs}(y_r) = \sum_{a_{ij} \in A} T_{ij}(Y_i)\delta_{ij}^{rs,u} \tag{5.2}$$

where Y_i is the arrival time at node i and $\delta_{ij}^{rs,u}$ is the path-link incidence variable;

$\delta_{ij}^{rs,u}=1$ means that the link a_{ij} is on the path p_u^{rs}, and $\delta_{ij}^{rs,u}=0$ otherwise. As both Y_i and $T_{ij}(y)$ are random variables, the path travel time $T_u^{rs}(y_r)$ is a stochastic process conditionally depending on each arrival time Y_i and link travel time $T_{ij}(y)$ along the path. The arrival time at destination s, denoted by $Y_s(y_r)$, can be calculated as

$$Y_s(y_r)=y_r+T_u^{rs}(y_r) \tag{5.3}$$

Since $Y_s(y_r)$ is a random variable, the on-time arrival probability (or travel time reliability), denoted by α, can be expressed as

$$\alpha=\Pr(Y_s(y_r)\le y_s) \tag{5.4}$$

As indicated in Chapter 2, the on-time arrival probability represents travellers' attitudes toward the risks of being late. $\alpha>0.5$, $\alpha=0.5$ and $\alpha<0.5$ represents risk-averse, risk-neutral and risk-seeking attitude respectively.

According to Chen and Ji (2005), the alpha-shortest path that minimizes the travel time budget b required to ensure α probability of on-time arrival can be expressed as the following optimization problem:

$$\text{Min}\quad b=y_s-y_r \tag{5.5}$$

Subject to

$$\Pr(Y_s(y_r)\le y_s)\ge\alpha \tag{5.6}$$

$$Y_s(y_r)=y_r+T_u^{rs}(y_r) \tag{5.7}$$

$$T_u^{rs}(t_r)=\sum_{a_{ij}\in A}T_{ij}(Y_i)\delta_{ij}^{rs} \tag{5.8}$$

$$\sum_{j\in SCS(i)}\delta_{ij}^{rs,u}-\sum_{w\in PDS(i)}\delta_{wi}^{rs,u}=\begin{cases}1 & \forall\, i=r\\ 0, & \forall\, i\ne r;\, i\ne s\\ -1 & \forall\, i=s\end{cases} \tag{5.9}$$

$$\delta_{ij}^{rs,u}\in\{0,1\},\quad \forall a_{ij}\in A \tag{5.10}$$

Eq. (5.5) is the travel time budget that travellers want to minimize. Eq. (5.6) defines the probabilistic constraint that ensures that the on-time arrival probability is greater or equal to a pre-determined threshold α. The confidence level α can be determined based on travellers' trip purposes. Eqs. (5.7-5.8) define the path travel

time and arrival time as mentioned in Eqs. (5.2-5.3). Eq. (5.9) ensures that the links on the reliable shortest path are feasible. Eq. (5.10) is concerned with the link-path incidence variables which should be binary in nature.

Let P1 be the problem defined by Eqs. (5.5-5.10). Given departure time y_r, the problem P1 represents the forward TD-RSPP which aims to determine the earliest arrival time and associated reliable shortest path. Conversely, given a preferred arrival time y_s, the problem P1 represents the backward TD-RSPP to determine the latest arrival time and associated reliable shortest path. In the following sections, solution algorithms for solving both forward and backward TD-RSPP are introduced.

5.3. Stochastic Travel Speed Model

As indicated above, S-LTM (stochastic link travel time model) is a commonly used model for representing the link travel times in STD networks (Chang et al., 2005; Nie and Wu, 2009a). Link travel time distribution under S-LTM depends on the time instance vehicles entering the link; and the link travel time distribution is assumed to be fixed when vehicles travelling on the link. This S-LTM is simple, however it may has a significant bias on link travel times, since traffic conditions cannot be continuously updated when vehicles travelling on the link. In addition, link travel times under S-LTM may violate the S-FIFO property, resulting in a computational difficulty for solving TD-RSPP (Nie and Wu, 2009b).

In this section, a new stochastic travel speed model (S-TSM) is proposed to better represent link travel time distribution. In S-TSM, travel speeds of a link are assumed to be stable within each time interval, and thus travellers' experienced travel speeds can be varied when vehicles are travelling on the link. The model formulation of S-TSM is presented in Section 5.3.1. The solution algorithm for generating link travel time distribution $T_{ij}(y)$ from speed distributions V_{ij} is introduced in Section 5.3.2.

5.3.1 Model formulation

Suppose a vehicle enters link a_{ij} at time instant $y_i \in (\Delta_{n-1}, \Delta_n)$. In S-TSM, the travel speed distribution V_{ij}^m is assumed to be fixed in each time interval (Δ_{m-1}, Δ_m). As such, the distance $D_{y_i}^{y_a}$ that the vehicle travels, at time instance $y_a \in (\Delta_{m-1}, \Delta_m)$, can be calculated as

$$D_{y_i}^{y_a} = V_{ij}^n(\Delta_n - y_i) + V_{ij}^{n+1}\Delta + ... + V_{ij}^{m-1}\Delta + V_{ij}^m(y_a - \Delta_{m-1}) \tag{5.11}$$

Thus, travel distance $D_{y_i}^{y_a}$ is also a random variable. The probability that the vehicle arrived at node j before time instance y_a can be expressed as

$$\Pr(D_{y_i}^{y_a} \geq d_{ij}) = \lambda \tag{5.12}$$

This arrival probability λ can also be expressed in terms of arrival time distribution as

$$\Phi_{Y_j(y_i)}(y_a) = \lambda \tag{5.13}$$

where $Y_j(y_i)$ is arrival time distribution of the vehicle arriving at head node j. Based on Eq. (5.13), the inverse CDF of arrival time distribution at λ confidence level can be calculated as

$$\Phi_{Y_j(y_i)}(y_a) = \lambda \tag{5.14}$$

If $\Phi_{Y_j(y_i)}^{-1}(\lambda)$ for any confidence level λ is calculated using Eqs. (5.11-5.14), the whole arrival time distribution can then be generated and described in Section 5.3.2.

Link travel time $T_{ij}(y_i)$, required for the vehicle to travel on link a_{ij}, can be expressed as

$$T_{ij}(y_i) = Y_j(y_i) - y_i \tag{5.15}$$

In S-TSM, the link travel time $T_{ij}(y_i)$ satisfies S-FIFO property by the following theorem.

Theorem 5.1. (Stochastic first in first out). Assume that two vehicles respectively

enters the link a_{ij} at times y_1 and y_2, and they will arrive at link head node j at $Y_j(y_1)$ and $Y_j(y_2)$. If $y_1 < y_2$ holds, $\Phi^{-1}_{Y_j(y_1)}(\lambda) < \Phi^{-1}_{Y_j(y_2)}(\lambda) \quad \forall \lambda \in (0,1)$ is then satisfied.

Proof. According to Eqs. (1-4), $\Phi^{-1}_{Y_j(y_1)}(\lambda) < \Phi^{-1}_{Y_j(y_2)}(\lambda) \quad \forall \lambda \in (0,1)$ is equivalent to prove $\Phi^{-1}_{D^{y_a}_{y_1}}(\lambda) > \Phi^{-1}_{D^{y_a}_{y_2}}(\lambda), \forall \lambda \in (0,1), \quad \forall y_a \in \Omega$ as follows.

$$\Phi_{D^{y_a}_{y_1}}(d) - \Phi_{D^{y_a}_{y_2}}(d) = \int_0^d \int_0^{d-y} f_{D^{y_2}_{y_1}, D^{y_a}_{y_2}}(x,y)dxdy - \int_0^d f_{D^{y_a}_{y_2}}(x)dx$$

$$= \int_0^d \left(\int_0^{d-y} f_{D^{y_2}_{y_1}, D^{y_a}_{y_2}}(x,y)dt - \int_0^{+\infty} f_{D^{y_2}_{y_1}, D^{y_a}_{y_2}}(x,y)dy \right) dx < 0$$

Therefore, $\Phi_{D^{y_a}_{y_1}}(d) < \Phi_{D^{y_a}_{y_2}}(d), \quad \forall d \in R^+, \quad \forall y_a \in \Omega$ is satisfied. Thus, we have $\Phi^{-1}_{D^{y_a}_{y_1}}(\lambda) > \Phi^{-1}_{D^{y_a}_{y_2}}(\lambda), \forall \lambda \in (0,1)$. □

S-FIFO is a stochastic extension of the FIFO principle. S-FIFO guarantees that, at any confidence level, travellers arrive at the end of a link in the same order in which they enter that link. If the travel time distribution of a link satisfies the S-FIFO property, the link is said to be a S-FIFO link. Similarly, if every link in a network is a S-FIFO link, the network is then said to be a S-FIFO network.

Let $\Phi^{-1}_{Y_s(y_r)}(\lambda)$ and $\Phi^{-1}_{T^{rs}_u(y_r)}(\lambda)$ be the inverse CDF of arrival time $Y_s(y_r)$ and path travel time $T^{rs}_u(y_r)$ at λ confidence level respectively. It can be proved that the path travel time, under S-TSM, is S-FIFO as follows.

Proposition 5.1. In S-FIFO networks, the path travel time satisfies the S-FIFO property: $y_1 < y_2 \Rightarrow \Phi^{-1}_{Y_s(y_1)}(\lambda) = y_1 + \Phi^{-1}_{T^{rs}_u(y_1)}(\lambda) < \Phi^{-1}_{Y_s(y_2)}(\lambda) = y_2 + \Phi^{-1}_{T^{rs}_u(y_2)}(\lambda)$, $\forall \lambda \in (0,1)$.

Proof. See Appendix C. □

A simple example is given to illustrate the concepts of S-TSM with a comparison with the traditional S-LTM. As shown in Fig. 5.1, all travel speeds V^n_{ij} are assumed

to follow normal distributions in this simple example. Link length is 600 meters and the time interval used is one minute. The means of link travel speeds are given in Fig. 5.1a. The standard deviations of link travel speeds are 0.2 of their corresponding mean value (coefficients of variation (CV) are 0.2). The correlation coefficients between each two travel speed distributions are assumed to be 0.5.

These travel speed distributions are then converted into travel time distributions under S-LTM. As suggested by Kaparias et al. (2008), link travel time distributions $\tilde{T}_{ij}^{n}$ can be estimated from corresponding travel speed distributions and approximated as lognormal distributions. The means of link travel times are derived from the link distance dividing corresponding travel speeds (see Fig. 5.1b). The standard deviations of link travel speeds are 0.2 of their corresponding mean value (using the relationship that the CV of link travel times is equal to that of travel speeds). The temporal correlations between travel time distributions of different time interval can be ignored in the S-LTM, as the link travel time distribution only depends on the time instant travellers enter the link.

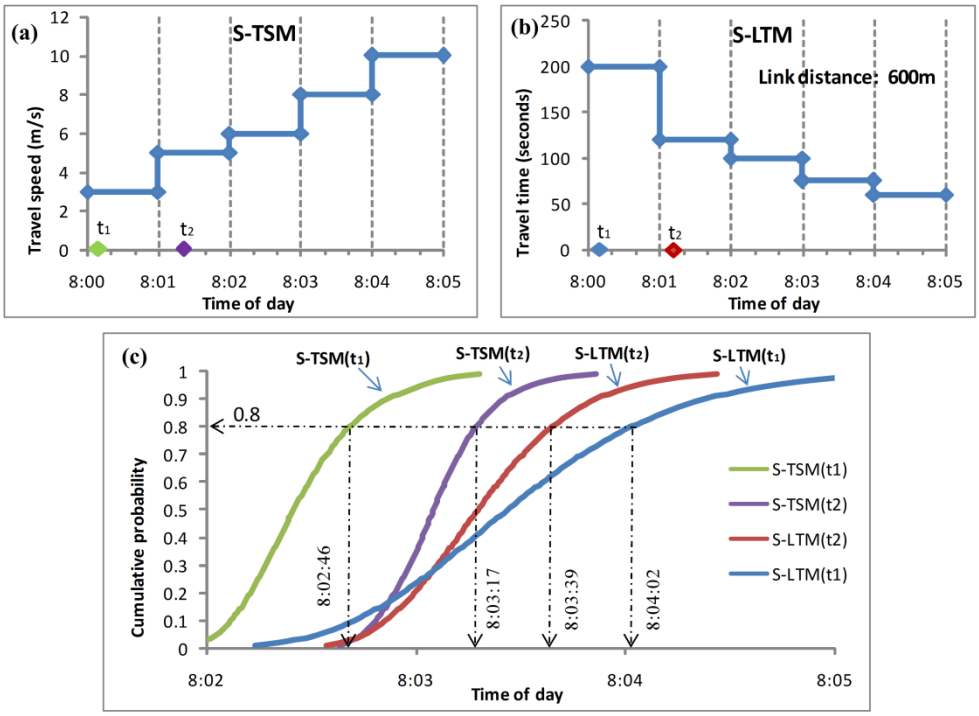


Figure 5.1 A simple illustrative example for the S-LTM and the S-TSM
(a) Mean travel speeds (b) Mean travel times (c) Arrival time distributions

Consider two vehicles entering the link at y_1 = 8:00:10 and y_2 = 8:01:20 respectively. Using S-LTM, the arrival time distribution $\tilde{Y}_j(y_1)$ of the vehicle arriving at a head node can be easily calculated by $\tilde{Y}_j(y_1) = y_1 + \tilde{T}_{ij}(y_1)$. Fig. 5.1c illustrates these two arrival time distributions $\tilde{Y}_j(y_1)$ and $\tilde{Y}_j(y_2)$ (see S-LTM(t_1) and S-LTM(t_2)). It can be clearly observed from Fig. 5.1c that link travel times in the S-LTM may not satisfy the S-FIFO property. For example, $\Phi^{-1}_{\tilde{Y}_j(y_1)}(0.8) = 8:04:02 < \Phi^{-1}_{\tilde{Y}_j(y_2)}(0.8) = 8:03:39$. This indicates that at the same probability level of traffic congestion ($\lambda = 0.8$), vehicles entered the link 70 seconds later will arrive at the end of the link 23 seconds earlier.

The arrival time distributions $Y_j(y_1)$ and $Y_j(y_2)$ can also be calculated using the S-TSM (see S-TSM(t_1) and S-TSM(t_2) in Fig. 5.1c). The detailed method for generating the arrival time distributions based on S-TSM is given in the following Section 5.3.2. It can be seen from Fig. 5.1c that under S-TSM, arrival time distributions are S-FIFO consistent (i.e. $\Phi^{-1}_{Y_j(y_1)}(\lambda) < \Phi^{-1}_{Y_j(y_2)}(\lambda)\ \forall \lambda \in (0,1)$). It can also be observed from Fig. 5.1c that arrival time distribution $\tilde{Y}_j(y_1)$ generated by S-LTM, has a significant bias when compared with $Y_j(y_1)$ generated by S-FSM. This bias is mainly due to S-LTM's assumption of fixed link travel time when vehicles travel on the link, although traffic conditions are significantly improved after vehicles enter the link.

5.3.2 Discretization and solution algorithm

In this section, a discrete method for generating link travel time distribution $T_{ij}(y_i)$, is proposed. The CDF of link travel time distribution is discretized into L equal-probability intervals, namely $\lambda = \varepsilon, 2\varepsilon, ..., 1.0$, where $L\varepsilon = 1.0$. Corresponding to L discrete points, the generation of link travel time distribution is equivalent to the generation of a sequence of discrete inverse CDF: $\Phi^{-1}_{T_{ij}(y_i)}(\lambda)$,

$\lambda = \varepsilon, 2\varepsilon, ..., 1.0$.

In this study, it is assumed that link travel speed V_{ij}^{n} follows a normal distribution as adopted in the work of Kaparias et al. (2008). The mean and standard deviation of travel speed distribution are respectively denoted by v_{ij}^{n} and $\sigma_{V_{ij}^{n}}$. Given a vehicle starting at head node i at time instance $y_i \in (\Delta_{n-1}, \Delta_n)$, the travel distance $D_{y_i}^{\Delta_m}$ at time instant Δ_m can be expressed as

$$D_{y_i}^{\Delta_m} = V_{ij}^{n}(\Delta_n - y_i) + V_{ij}^{n+1}\Delta + ... + V_{ij}^{m}\Delta \tag{5.16}$$

The mean and standard deviation of travel distance $D_{y_i}^{\Delta_m}$ are denoted by $d_{y_i}^{\Delta_m}$ and $\sigma_{D_{y_i}^{\Delta_m}}$ respectively. They can be calculated as

$$d_{y_i}^{\Delta_m} = v_{ij}^{n}(\Delta_n - y_i) + \sum_{k=n+1}^{k=m} v_{ij}^{k}\Delta \tag{5.17}$$

$$\sigma_{D_{y_i}^{\Delta_m}}^{2} = \sigma_{V_{ij}^{n}}^{2}(\Delta_n - y_i)^2 + \sum_{k=n+1}^{k=m}\sigma_{V_{ij}^{k}}^{2}\Delta^2 + \sum_{k=n+1}^{k=m} 2\,\mathrm{cov}(V_{ij}^{k}, V_{ij}^{n})\Delta(\Delta_n - y_i) + \sum_{k=n+1}^{k=m}\sum_{\substack{w=n+1 \\ w\neq k}}^{w=m} 2\,\mathrm{cov}(V_{ij}^{k}, V_{ij}^{w})\Delta^2 \tag{5.18}$$

where $\mathrm{cov}(V_{ij}^{k}, V_{ij}^{w})$ is the travel speed temporal covariance between two time intervals. Such temporal covariance can also be calculated by

$$\mathrm{cov}(V_{ij}^{k}, V_{ij}^{w}) = \rho_{V_{ij}^{k}}^{V_{ij}^{w}}\sigma_{V_{ij}^{k}}\sigma_{V_{ij}^{w}} \tag{5.19}$$

where $\rho_{V_{ij}^{k}}^{V_{ij}^{w}} \in [-1, +1]$ is the correlation coefficient between V_{ij}^{k} and V_{ij}^{w}; $\rho_{V_{ij}^{k}}^{V_{ij}^{w}} = +1$ and $\rho_{V_{ij}^{k}}^{V_{ij}^{w}} = -1$ are perfect positive and negative relationships between these two variables respectively.

Let $\Phi_{D_{y_i}^{\Delta_{m-1}}}(d_{ij}) = 1 - \beta_{m-1}$ and $\Phi_{D_{y_i}^{\Delta_m}}(d_{ij}) = 1 - \beta_m$ be the CDF of $D_{y_i}^{\Delta_{m-1}}$ and $D_{y_i}^{\Delta_m}$ for the link distance d_{ij} respectively. Thus, $\Phi_{T_{ij}(y_i)}^{-1}(\beta_{m-1})$ and $\Phi_{T_{ij}(y_i)}^{-1}(\beta_m)$ can be expressed as

$$\Phi_{T_{ij}(y_i)}^{-1}(\beta_{m-1}) = \Delta_{m-1} - y_i \tag{5.20}$$

$$\Phi_{T_{ij}(y_i)}^{-1}(\beta_m) = \Delta_m - y_i \tag{5.21}$$

For any $\lambda \in (\beta_{m-1}, \beta_m)$, $\Phi^{-1}_{T_{ij}(y_i)}(\lambda) \in (\Delta_{m-1}, \Delta_m)$ can be determined as following quadratic equation:

$$d_{ij} = d_{y_i}^{\Delta_{m-1}} + v_{ij}^m x + z_\lambda \sqrt{\sigma^2_{D_{y_i}^{\Delta_{m-1}}} + \sigma^2_{V_{ij}^m} x^2 + 2\eta x} \tag{5.22}$$

$$\eta = \mathrm{cov}(V_{ij}^n, V_{ij}^m)(\Delta_n - y_i) + \sum_{k=n+1}^{k=m-1} \mathrm{cov}(V_{ij}^k, V_{ij}^m)\Delta \tag{5.23}$$

where $x = \Phi^{-1}_{T_{ij}(y_i)}(\lambda) + y_i - \Delta_{m-1}$ and z_λ is the inverse CDF of standard normal distribution at λ confidence level. The quadratic equation, Eqs. (5.22-5.23), can be easily solved as

$$x = (-a_2 \pm \sqrt{a_2^2 - 4a_1 a_3}) / 2a_1 \tag{5.24}$$

$$a_1 = z_\lambda^2 \sigma^2_{V_{ij}^m} - (v_{ij}^m)^2 \tag{5.25}$$

$$a_2 = 2z_\lambda^2 \eta + 2d_{ij} v_{ij}^m - 2d_{y_i}^{\Delta_{m-1}} v_{ij}^m \tag{5.26}$$

$$a_3 = z_\lambda^2 \sigma^2_{D_{y_i}^{\Delta_{m-1}}} - (d_{ij} - d_{y_i}^{\Delta_{m-1}})^2 \tag{5.27}$$

Two solutions of x can be found by Eq. (5.24). When $\lambda \geq 0.5$, $x = (-a_2 - \sqrt{a_2^2 - 4a_1 a_3}) / 2a_1$ is adopted; while $x = (-a_2 + \sqrt{a_2^2 - 4a_1 a_3}) / 2a_1$ is chosen when $\lambda \leq 0.5$. Therefore, $\Phi^{-1}_{T_{ij}(y_i)}(\lambda)$ can be calculated by

$$\Phi^{-1}_{T_{ij}(t_i)}(\lambda) = x + \Delta_{m-1} - y_i \tag{5.28}$$

Based on the above method, link travel time distribution can be generated using the following procedure.

Procedure: GenerateLinkTime

Inputs: link a_{ij}, time instant $y_i \in (\Delta_{n-1}, \Delta_n)$ and discrete number L

Returns: discrete CDF of link travel time distribution, $\Phi^{-1}_{T_{ij}(y_i)}(\lambda)$, $\lambda = \varepsilon, 2\varepsilon, ..., 1.0$

Step 1. Initialization

Set time instance $\Delta_{m-1} := y_i$, mean distance $d_{y_i}^{\Delta_{m-1}} := 0$, and distance variance $\sigma^2_{D_{y_i}^{\Delta_{m-1}}} := 0$.

Set $\Delta_m := \Delta_n$, and calculate $d_{y_i}^{\Delta_m} := v_{ij}^n (\Delta_n - y_i)$ and $\sigma^2_{D_{y_i}^{\Delta_m}} := \sigma^2_{V_{ij}^n} (\Delta_n - y_i)^2$.

Set $\beta_{m-1} \coloneqq 0$ and $\eta \coloneqq 0$, and calculate $\beta_m \coloneqq 1 - \Phi_{D_{y_i}^{\Delta_m}}(d_{ij})$.

Step 2. Generate discrete link travel times.

For every $\lambda \in (\beta_{m-1}, \beta_m)$

Calculate parameters a_1, a_2 and a_3 using Eqs. (5.25-5.27).

If $\lambda \geq 0.5$, then $x = (-a_2 - \sqrt{a_2^2 - 4a_1 a_3})/2a_1$; otherwise $x = (-a_2 + \sqrt{a_2^2 - 4a_1 a_3})/2a_1$.

Calculate $\Phi^{-1}_{T_{ij}(y_i)}(\lambda) \coloneqq \Delta_{m-1} + x - y_i$ using Eq. (5.28).

End for

Step 3. Scan next time interval.

Set $\Delta_{m-1} \coloneqq \Delta_m$, $\Delta_m \coloneqq \Delta_m + \Delta$, $d_{y_i}^{\Delta_{m-1}} \coloneqq d_{y_i}^{\Delta_m}$ and $\sigma^2_{D_{y_i}^{\Delta_{m-1}}} \coloneqq \sigma^2_{D_{y_i}^{\Delta_m}}$.

Calculate $\eta \coloneqq \mathrm{cov}(V_{ij}^n, V_{ij}^m)(\Delta_n - y_i) + \sum_{k=n+1}^{k=m-1} \mathrm{cov}(V_{ij}^k, V_{ij}^m)\Delta$ using Eq. (5.23).

Calculate $d_{y_i}^{\Delta_m} = d_{y_i}^{\Delta_{m-1}} + v_{ij}^m \Delta$ and $\sigma^2_{D_{y_i}^{\Delta_m}} \coloneqq \sigma^2_{D_{y_i}^{\Delta_{m-1}}} + \sigma^2_{V_{ij}^m}\Delta^2 + 2\eta\Delta$.

Set $\beta_{m-1} \coloneqq \beta_m$ and calculate $\beta_m \coloneqq 1 - \Phi_{D_{y_i}^{\Delta_m}}(d_{ij})$.

If $\beta_{m-1} \geq 0.999$, then Stop; otherwise goto Step 2.

The detailed steps of link travel time distribution (or arrival time distribution) generation using S-TSM in Fig. 5.1 are given as follows. The number of discrete element is set as $L = 100$ ($\varepsilon = 0.01$). Consider a vehicle entering the link at y_1 = 8:00:10. In Step 1, travel distance distribution, $D_{y_1}^{\Delta_1}$, at Δ_1 = 8:01:00 can be calculated by using Eqs. (5.17-5.18). Mean and travel distance variance are $d_{y_1}^{\Delta_1} = 150$ and $\sigma^2_{D_{y_1}^{\Delta_1}} = 900$. Then, $\beta_0 = 0$ and $\beta_1 = 1 - \Phi_{D_{y_1}^{\Delta_1}}(600) = 0$. Step 2 at the first iteration can be skipped as $\forall \lambda \notin (\beta_0, \beta_1)$. In Step 3, next time instance is set as Δ_2 = 8:02:00, and travel distance $D_{y_1}^{\Delta_2}$ can be calculated as $\eta = 15$, $d_{y_1}^{\Delta_2} = 450$ and $\sigma^2_{D_{y_1}^{\Delta_2}} = 6300$. Subsequently, $\beta_2 = 1 - \Phi_{D_{y_1}^{\Delta_2}}(600) = 0.029 < 0.999$, and thus the algorithm goes to Step 2 for the second iteration.

In Step 2 of the second iteration, there are two discrete points $\lambda = 0.01, 0.02 \in (0, 0.029)$. For $\lambda = 0.01$, three parameters $a_1 = -19.6$, $a_2 = 4661.7$ and $a_3 = -197650.1$ can be calculated using Eqs. (5.25-5.27). Since $\lambda = 0.01 < 0.5$, $x = (-a_2 + \sqrt{a_2^2 - 4a_1a_3}) / 2a_1 = 55.2$ is adopted and thus $\Phi^{-1}_{T_{ij}(y_1)}(0.01) = 105.2$. Using the same method, the inverse CDF, $\Phi^{-1}_{T_{ij}(y_1)}(0.02) = 108.2$, can also be calculated for $\lambda = 0.02$. In Step 3, time instant is set as $\Delta_3 = 8{:}03{:}00$, and travel distance distribution can be calculated as $\eta = 54$, $d^{\Delta_3}_{y_1} = 810$ and $\sigma^2_{D^{\Delta_3}_{y_1}} = 17964$. Then, $\beta_3 = 0.942 < 0.999$ can be obtained and the algorithm goes to Step 2 for the third iteration.

As the algorithm progresses, $\Phi^{-1}_{T_{ij}(y_1)}(\lambda)$ for $\forall \lambda \in (0.029, 0.942)$ and $\forall \lambda \in (0.942, 0.999)$ can be respectively calculated in the third and fourth iterations. In such a way, the link travel time distribution $T_{ij}(y_1)$, as well as arrival time distribution $T_j(y_1)$, can be generated. The generated arrival time distribution is illustrated in Fig. 5.1c (see S-TSM(t_1)).

5.4. Non-reversibility of TD-RSPP

As indicated in Section 5.1, it is well known that the shortest path problems in time-dependent networks can be solved by using either the forward search (from origin to destination) or backward search (from destination to origin) (Daganzo, 2002). However, this reversibility property does not hold in TD-RSPP. An illustrative example is given in Fig. 5.2.

As shown in Fig. 5.2a, the network consists of three nodes and two links. All link travel times are assumed to follow normal distributions. The means of link travel times are given in Fig. 5.2b. The standard deviations of link travel times are 0.25 of their corresponding mean value. Consider a forward search from Node 1 to Node 3 starting at $y_r = 8{:}23AM$. The path travel time generated by this forward search is

shown in Fig. 5.2c (the line in blue color). As shown in the figure, the mean and standard deviation of the path travel time are 30 minutes and 5.5 minutes respectively. To ensure $\alpha = 90\%$ probability of on-time arrival, travellers should assign $b = 37$ minutes travel time budget, and thus their earliest arrival time is $y_s = 9AM$.

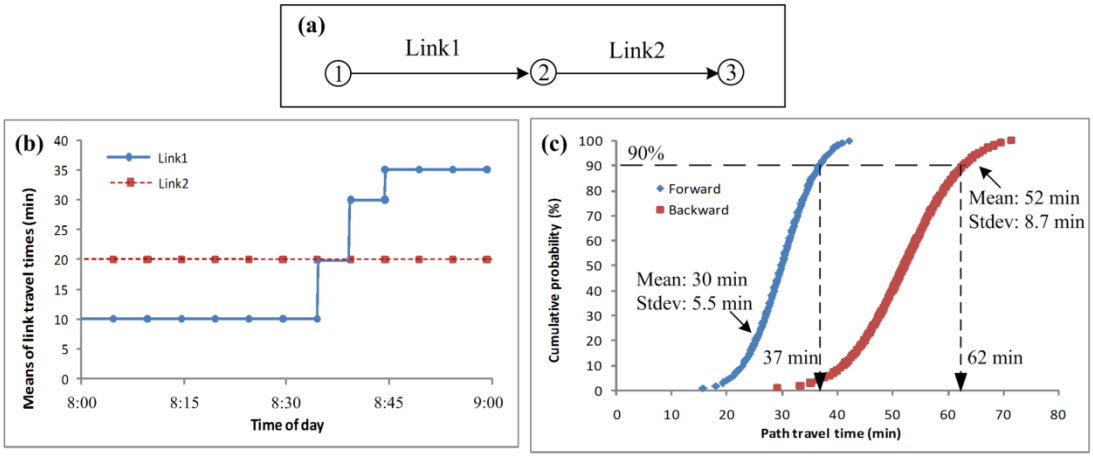


Figure 5.2 An illustrative example of the non-reversibility property
(a) a simple network (b) means of link travel times (c) path travel time distributions using forward search and backward search

The path travel time distribution is also generated by a backward search from Node 3 to Node 1 starting at $y_s = 9AM$. As shown in Fig. 5.2c (the line in red color), the path travel time generated by a backward search is significantly different from that generated by the forward search. It can be clearly observed from the figure that the backward search can result in a significant travel time budget bias ($b = 62$ minutes) and thus provide a sub-optimal departure time choice ($y_r = 7:58AM$).

This non-reversible property of TD-RSPP is mainly due to the temporal dependence of the path travel time distribution in STD networks. As formulated in Eq. (5.2), the path travel time distribution in STD networks is conditional on arrival times at intermediate nodes and link travel times along the path. As arrival time distributions must be generated by the forward search from origin to destination, the path travel time distributions cannot be generated by the reverse search from destination to origin. Therefore, TD-RSPP is not reversible and the backward TD-RSPP cannot be

solved by the reverse search using the algorithms designed for the forward problem. In the following sections, two different solution algorithms for solving the forward and backward TD-RSPP are presented.

5.4.1. Solution algorithm for solving the forward TD-RSPP

This section presents a solution algorithm for solving the forward TD-RSPP. In the problem P1, the probabilistic constraint in Eq. (5.6) is equivalent to

$$\Phi^{-1}_{T_u^{rs}(t_r)}(\alpha) \le b \tag{5.29}$$

By setting $b = \Phi^{-1}_{T_u^{rs}(t_r)}(\alpha)$, the probability constraint can be eliminated. Thus, the forward P1 problem is equivalent to following shortest path problem P2:

$$\text{Min} \quad b = \Phi^{-1}_{T_u^{rs}(t_r)}(\alpha) \tag{5.30}$$

Subject to

$$T_u^{rs}(y_r) = \sum_{a_{ij} \in A} T_{ij}(Y_i)\delta_{ij}^{rs,u} \tag{5.31}$$

$$\sum_{j \in SCS(i)} \delta_{ij}^{rs,u} - \sum_{w \in PDS(i)} \delta_{wi}^{rs,u} = \begin{cases} 1 & \forall\, i = r \\ 0, & \forall\, i \ne r; i \ne s \\ -1 & \forall\, i = s \end{cases} \tag{5.32}$$

$$\delta_{ij}^{rs,u} \in \{0,1\}, \quad \forall a_{ij} \in A \tag{5.33}$$

The problem P2 can be regarded as one of non-additive shortest path problems, as the path cost ($\Phi^{-1}_{T_u^{rs}(t_r)}(\alpha)$) cannot be calculated by the sum of the related link costs. The FSD (first order stochastic dominant) condition, established in Chapter 3, can be adopted to determine the dominated path in S-FIFO networks as follows

Proposition 5.2 (First order stochastic dominance condition). Given two paths $p_u^{ri} \ne p_v^{ri} \in P^{ri}$ in a S-FIFO network, p_u^{ri} dominates p_v^{ri} if they satisfy $\Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{T_v^{ri}(y_r)}(\lambda), \forall \lambda \in (0,1)$.

Proof. See Appendix C. □

The objective function value has the following monotonic property.

Proposition 5.3. Given a path p_u^{ri} and a adjacent link a_{ij}, $p_u^{rj} = p_u^{ri} \oplus a_{ij}$ satisfies $\Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{T_u^{rj}}(\lambda)$, $\forall \lambda \in (0,1)$.

Proof. See Appendix C. □

Using this monotonic property, the multi-criteria A* algorithm, presented in last two chapters, is extended to solve the forward TD-RSPP. The proposed algorithm makes use of a heuristic evaluation function $F(p_u^{ri}) = \Phi^{-1}_{T_u^{ri}(y_r)}(\alpha) + h(i)$ as a label for the path p_u^{ri}, where $h(i)$ is an estimated travel time budget from node i to destination s. The heuristic function is admissible if the following inequality is satisfied

$$F(p_u^{rj}) = \Phi^{-1}_{T_u^{rj}(y_r)}(\alpha) + h(j) \geq F(p_u^{ri}) = \Phi^{-1}_{T_u^{ri}(y_r)}(\alpha) + h(i) \tag{5.34}$$

Eq. (5.34) indicates that $F(p_u^{ri})$ should monotonically increase with path extensions.

The steps of the proposed A* algorithm (named *F-TDRSP-A**) are described below. Let $P^{ri} = \{p_1^{ri}, ..., p_n^{ri}\}$ a set of non-dominated paths from the origin r to node i. All non-dominated paths (from all nodes) are maintained in a scan eligible set, denoted by $SE = \{p_1^{ri}, ..., p_u^{ri}, ... p_1^{rj}, ...\}$ using a priority queue. All non-dominated paths in the queue are ordered based on $F(p_u^{ri})$. At each iteration, the label p_u^{ri} with minimum $F(p_u^{ri})$ is selected from SE for path extension. A temporary path, denoted by $p_u^{rj} := p_u^{ri} \oplus a_{ij}$, is then constructed by extending the selected path p_u^{ri} to its each successor node $\forall j \in SCS(i)$. The temporary path p_u^{rj} is inserted into P^{rj} (the set of non-dominated paths at node j), if p_u^{rj} is not dominated by any path in P^{rj} under FSD condition. The temporary path p_u^{rj} may also dominate paths in P^{rj}. These dominated paths are discarded without further consideration. The algorithm continues the path extension process until the destination is reached or

SE becomes empty.

---

*F-TDRSP-A**

Inputs: O-D nodes, confidence level α, and departure time y_r

Returns: the reliable shortest path

Step 1. Initialization:

Create a path p_u^{rr} from origin to itself.

Set path travel time distribution $T_u^{rr}(y_r) := 0$, and calculate $h(r)$ and $F(p_u^{rr})$.

Set $P^{rr} := \{p_u^{rr}\}$ and $SE := \{p_u^{rr}\}$.

Step 2. Path selection:

If $SE = \phi$, then stop; otherwise, continue.

Select p_u^{ri} at the top of SE and remove p_u^{ri} from SE.

If $i = s$, then stop; otherwise continue.

Step 3. Path extension:

For every successor node $j \in SCS(i)$

Construct a temporary path $p_u^{rj} := p_u^{ri} \oplus a_{ij}$.

Generate distribution $T_u^{rj}(y_r) := T_u^{ri}(y_r) + T_{ij}(Y_i)$, and calculate $h(j)$ and $F(p_u^{rj})$ using α-discrete approximation method.

If p_u^{rj} is a non-dominated path under FSD condition, then insert p_u^{rj} into P^{rj} and SE and remove all paths dominated by p_u^{rj} from P^{rj} and SE.

End for

Goto Step 2.

Proposition 5.4. If the heuristic function is admissible, the *F-TDRSP-A** algorithm can determine the reliable shortest path when a non-dominated path p_u^{rs} between O-D nodes is selected from SE.

Proof. Suppose that $\overline{P}^{rs}$ containing all non-dominated paths between O-D nodes except for p_u^{rs}. As the path p_u^{rs} is selected from SE, its heuristic function value $F(p_u^{rs})$ is less than that of any path in SE. If the heuristic function is admissible,

the heuristic function value is monotonically increasing with path extensions. Since all paths in $\overline{P}^{rs}$ are extended from current paths in SE, $F(p_u^{rs})$ is less than the heuristic function value of any path in $\overline{P}^{rs}$. As $h(s)=0$, $F(p_u^{rs})$ is the minimum travel time budget between O-D nodes and thus p_u^{rs} is the reliable shortest path. □

The approximation method used for generating path travel time distributions ($T_u^{rj}(y_r)=T_u^{ri}(y_r)+T_{ij}(Y_i)$) is another important factor affecting the performance of *F-TDRSP-A**. The methods for generating path travel time distributions are discussed in Section 5.5.

5.4.2. Solution algorithm for the backward TD-RSPP

Given a preferred arrival time y_s, the backward P1 problem is to determine the latest departure y_r satisfying a given on-time arrival probability. As path travel time distributions in TD-RSPP depend on the departure time, a two-stage approach is proposed to solve the backward TD-RSPP. The first stage is to determine the optimal departure time y_r for an initial path p_0^{rs} between O-D nodes. The second stage, after the initial departure time y_r has been determined, is to find the reliable shortest path p^{rs} by solving the forward problem P2. Based on this idea, the backward P1 problem is re-formulated as following two-stage problem P3:

$$\text{Min } | y_r + \Phi^{-1}_{T_u^{rs}(y_r)}(\alpha) - y_s | \tag{5.35}$$

where $\Phi^{-1}_{T_u^{rs}(y_r)}(\alpha)$ in the objective function can be calculated by solving the problem P2. It can be proved that the problem P3 is equivalent to the backward P1 problem as follows.

Proposition 5.5. The problem P3 is equivalent to the backward P1 problem in S-FIFO networks.

Proof. In the S-FIFO network, for any path p_u^{rs}, $y_r + \Phi^{-1}_{T_u^{rs}(y_r)}(\alpha)$ is monotonically increasing with y_r according to Proposition 1. Thus, the optimal departure time

$y_r^u = y_s - \Phi^{-1}_{T_u^{rs}(y_r)}(\alpha)$ can be determined for any path p_u^{rs}. Suppose y_r^* and p_*^{rs} is the optimal solution of the problem P3. Thus, $y_r^* + \Phi^{-1}_{T_*^{rs}(y_r^*)}(\alpha) = y_s$. As p_*^{rs} is the reliable shortest path at y_r^*, we have $\Phi^{-1}_{T_u^{rs}(y_r^*)}(\alpha) \geq \Phi^{-1}_{T_*^{rs}(y_r^*)}(\alpha)$ and $y_r^* + \Phi^{-1}_{T_u^{rs}(y_r^*)}(\alpha) \geq y_s$, $\forall p_u^{rs} \in \boldsymbol{P^{rs}}$. According to Proposition 1, we have $y_r^u + \Phi^{-1}_{T_u^{rs}(y_r^u)}(\alpha) > y_r^* + \Phi^{-1}_{T_u^{rs}(y_r^*)}(\alpha) \geq y_s$, $\forall y_r^u > y_r^*$, $\forall p_u^{rs} \in \boldsymbol{P^{rs}}$. Therefore, y_r^* and p_*^{rs} is also the optimal solution for the backward P1 problem. □

The solution algorithm (called *B-TDRSP*) for solving the above problem P3 is described as follows. In the first step, an initial path p_0^{rs}, with minimum network distance d_{rs}, is determined by Dijkstra's algorithm. The initial departure time y_r^0 is, then, determined by $y_s - d_{rs}/V_{\max}$. In the second step, the optimal departure time y_r^m for using a path p_m^{rs} is determined by solving a simple one parameter minimization sub-problem in Eq. (5.35). This minimization sub-problem can be solved by iteratively moving y_r^m to $y_s - \Phi^{-1}_{T_m^{rs}(y_r^m)}(\alpha)$, until the objective function $|y_r^m + \Phi^{-1}_{T_m^{rs}(y_r^m)}(\alpha) - y_s|$ is less than a given small error tolerance ε (e.g. $\varepsilon = 0.1$ second). In the third step, the *F-TDRSP-A** algorithm is employed to find the reliable shortest path p_{m+1}^{rs} at the departure time y_r^m determined in the second step. The algorithm is continuously running among Steps 2-3 until the termination condition $\Phi^{-1}_{T_{m+1}^{rs}(y_r^m)}(\alpha) = \Phi^{-1}_{T_m^{rs}(y_r^m)}(\alpha)$ is satisfied.

B-TDRSP

Inputs: Confidence level α, preferred arrival time y_s and error tolerance ε

Returns: the reliable shortest path and the optimal departure time y_r

Step 1. Initialization:

Find the path p_0^{rs} with shortest distance d_{rs} using Dijkstra's algorithm.

Set initial departure time $y_r^0 := y_s - d_{rs}/V_{\max}$.

Step 2. Departure time determination stage:

Generate path travel time distribution $T_m^{rs}(y_r)$ for the path p_m^{rs}.

If $|y_r + \Phi^{-1}_{T_m^{rs}(y_r)}(\alpha) - y_s| \leq \varepsilon$, then Goto Step 3; otherwise continue.

Set $y_r := y_s - \Phi^{-1}_{T_m^{rs}(y_r)}(\alpha)$ and Goto Step 2.

Step 3. Reliable shortest path finding stage:

Find reliable shortest path p_{m+1}^{rs} at departure time y_r^m using *F-TDRSP-A**.

If $\Phi^{-1}_{T_{m+1}^{rs}(y_r^m)}(\alpha) = \Phi^{-1}_{T_m^{rs}(y_r^m)}(\alpha)$, then stop; otherwise continue.

Set $y_r := y_s - \Phi^{-1}_{T_{m+1}^{rs}(y_r^m)}(\alpha)$ and Goto Step 2.

The simple example shown in Fig. 5.2 is adopted to illustrate the *B-TDRSP* algorithm. Suppose the length of both Links 1 and 2 is 5 kilometers (km); network design speed is 60 km/hour; Nodes 1 and 3 are origin and destination respectively; and the preferred arrival time is set as $y_3 = 9AM$, the same as that used in the reverse search of Fig. 5.2. In the first step, an initial path $p_0^{13} = a_{12} \oplus a_{23}$ and an initial departure time $y_1^0 = 7:58AM$ are determined.

In the second step, the optimal departure time for using p_0^{13} is calculated by using the iterative process. The steps of this iterative process are given in Table 5.1. In the first iteration, travel time distribution $T_0^{13}(y_1^0)$ is generated by using the α-discrete approximation method and then travel time budget $\Phi^{-1}_{T_0^{13}(y_1^0)}(0.9) = 37.2$ min can be obtained. As the objective function $|y_1^0 + \Phi^{-1}_{T_0^{13}(y_1^0)}(0.9) - y_3| = 34.8$ min $> \varepsilon = 0.1$ second, the algorithm continues the iterative process and a new departure time $y_1^1 = y_3 - \Phi^{-1}_{T_0^{13}(y_1^1)}(0.9)$ =8:23AM is then calculated for the second iteration. Finally, in the second iteration, $|y_1^1 + \Phi^{-1}_{T_0^{13}(y_1^2)}(0.9) - y_3| = 0 < \varepsilon = 0.1$ second can be achieved and thus optimal departure time y_1^m =8:23AM can be determined.

Table 5.1 The steps of the proposed *B-TDRSP* algorithm

Iteration	Departure Time	Travel time at 90% confidence level	Arrival time at 90% confidence level	Adjusted departure time
1	7:58AM	37.2min	8:35AM	8:23AM
2	**8:23AM**	37.2min	**9AM**	

Current time instant: 7:30AM, Preferred arrival time: 9AM

In the third step, the reliable shortest path p_{m+1}^{13} at $y_1^m = 8:23AM$ is calculated. As this simple example has only one path between O-D nodes, the termination condition $\Phi^{-1}_{T^{rs}_{m+1}(y_r^m)}(\alpha) = \Phi^{-1}_{T^{rs}_{0}(y_r^m)}(\alpha)$ can be satisfied; and thus the algorithm terminates in this iteration. It should be noted that for real road networks several iterations may be required for the algorithm running among Steps 2-3. Compared with the results of the forward search of Fig.5.2, it can be found that the *B-TDRSP* algorithm can simultaneously determine the latest departure time and reliable shortest path for this simple network. The optimality of *B-TDRSP* algorithm can be formally proved for general networks as follows:

Proposition 5.6. The *B-TDRSP* algorithm can simultaneously determine the latest departure time and reliable shortest path in finite iterations.

Proof. Suppose p_m^{rs} is the path generated in m iteration. In the S-FIFO network, an optimal departure time $y_r^m = y_s - \Phi^{-1}_{T^{rs}_{m}(y_r^m)}(\alpha)$ for p_m^{rs} can be determined according to Proposition 1. In the next $m+1$ iteration, given departure time t_r^m, the reliable shortest path p_{m+1}^{rs} can be determined in Step 3. Thus, $\Phi^{-1}_{T^{rs}_{m+1}(y_r^m)}(\alpha) \le \Phi^{-1}_{T^{rs}_{m}(y_r^m)}(\alpha)$. If $\Phi^{-1}_{T^{rs}_{m+1}(y_r^m)}(\alpha) < \Phi^{-1}_{T^{rs}_{m}(y_r^m)}(\alpha)$ (this indicates that $p_{m+1}^{rs} \neq p_m^{rs}$), we have $y_r^m < y_s - \Phi^{-1}_{T^{rs}_{m+1}(y_r^m)}(\alpha)$. According to Proposition 1, a new departure time $y_r^{m+1} = y_s - \Phi^{-1}_{T^{rs}_{m+1}(y_r^m)}(\alpha) > y_r^m$ can be determined in the Step 2. Therefore, departure time increases in the each iteration and reaches its maximum until $\Phi^{-1}_{T^{rs}_{m+1}(y_r^m)}(\alpha) = \Phi^{-1}_{T^{rs}_{m}(y_r^m)}(\alpha)$. Therefore, the *B-TDRSP* algorithm can simultaneously determine the latest departure time and reliable shortest path in finite iterations. □

5.5. Methods for Generating Path Travel Time Distribution

As indicated above, the path travel time distribution, in Eq. (5.2), is a stochastic process conditionally depending on arrival times at intermediate nodes and successor link travel times along the path. Thus, the generation of exact path travel time distribution is mathematically intractable and only approximation methods can be used (Fu and Rilett, 1998; Miller-Hooks and Mahmassani, 1998). In the following section, the effective α-discrete approximation method, proposed by Miller-Hooks and Mahmassani (1998), is briefly described. A mean-variance approximation method is then proposed to generate path travel time distribution when path travel times follow lognormal distributions.

5.5.1. α-discrete approximation method

The arrival time distribution ($Y_j = T_u^{rj}(y_r) + y_r$) is used in this approximation method, as the generated of $T_u^{rj}(y_r)$ is equivalent to that of Y_j. The CDF of arrival time distribution Y_i at node i is firstly discretized into L_1 equal-probability intervals, $Y_i = \{..., y_i^n, ...\}$. For each discrete arrival time y_i^n, the corresponding link travel time distribution $T_{ij}(y \mid y = y_i^n)$ is also discretized into L_2 equal-probability intervals, $T_{ij}(y \mid y = y_i^n) = \{..., t_{ij}^m(y \mid y = y_i^n), ...\}$. A discrete CDF of arrival time $Y_j = \{..., y_j^{nm}, ...\}$ can then be constructed by sorting $y_j^{nm} = y_i^n + t_{ij}^m(y \mid y = y_i^n)$, $\forall y_i^n$, $\forall t_{ij}^m(y \mid y = y_i^n)$. Finally, the discrete CDF of Y_j is aggregated into L_1 intervals to prevent the number of elements in Y_j from growing exponentially during the search process. It was found that this α-discrete method can obtain an accurate path travel time distribution when the number of discrete number is sufficiently large (Wu and Nie, 2009). This approximation method, however, requires considerable computational effort to calculate and sort $L_1 L_2$ discrete elements for each path extension operation.

5.5.2. Mean-variance approximation method

An approximation method, similar to that of Fu and Rilett (1998), is given in this section to generate the path travel time distribution by estimating its mean and variance. It is assumed that path travel time (or arrival time) follows a lognormal distribution. As indicated in Chapter 2, many empirical studies, based on field observations, have found that lognormal distributions can, indeed, well reflect path travel times in congested urban road networks. Under this assumption, generation of the path travel time distribution is equivalent to the estimation of its mean and variance, as the lognormal distribution is a two-parameter distribution.

Let $p_u^{rj} = p_u^{ri} \oplus a_{ij}$ be a path from origin r to node j passing through path p_u^{ri} and link a_{ij}. Suppose that the arrival time at node i, denoted by Y_i, follows a lognormal distribution, and its mean and standard deviation denoted y_i and σ_{y_i} are known. Let $u(Y_i)$ and $v(Y_i)$ be mean and variance of natural logarithm of Y_i:

$$v(Y_i) = \ln(1 + (\sigma_{y_i} / y_i)^2) \tag{5.36}$$

$$u(Y_i) = \ln(y_i) - 0.5 * v(Y_i) \tag{5.37}$$

The arrival time distribution at node j, can then be expressed as

$$Y_j = Y_i + T_{ij}(Y_i) \tag{5.38}$$

According to Fu and Rilett (1998), the mean and variance of Y_j can be calculated as

$$y_j = y_i + E(T_{ij}(Y_i)) \tag{5.39}$$

$$\sigma_{Y_j}^2 = \sigma_{Y_i}^2 + Var(T_{ij}(Y_i)) + 2Cov(Y_i, T_{ij}(Y_i)) \tag{5.40}$$

where $E(T_{ij}(Y_i))$ and $Var(T_{ij}(Y_i))$ are mean and variance of $T_{ij}(Y_i)$ respectively; and $Cov(Y_i, T_{ij}(Y_i))$ are travel time covariances between Y_i and T_{ij}.

Similar to Chang et al. (2005), a discrete method is proposed to estimate the conditional probability involved in Eqs. (5.39-5.40). In the proposed method, the arrival time distribution Y_i is discretized into L intervals of length Δ. Its probability mass function during time interval (Δ_{n-1}, Δ_n), denoted ω_i^n, can be calculated as

$$\omega_i^n = \Phi_{Y_i}(\Delta_n) - \Phi_{Y_i}(\Delta_{n-1}) \tag{5.41}$$

where $\Phi_{Y_i}(\Delta_n)$ and $\Phi_{Y_i}(\Delta_{n-1})$ respectively are the CDF of arrival time distribution Y_i for time instance Δ_n and Δ_{n-1}. For computational implementation, a numerical approach can be adopted to approximate $\Phi_{Y_i}(\Delta_n)$ as

$$\Phi_{Y_i}(\Delta_n) = \frac{1}{2} + \frac{1}{2} Erf\left(\frac{\ln(\Delta_n) - u(Y_i)}{\sqrt{2v(Y_i)}} \right) \tag{5.42}$$

$$Erf(x) \cong Sng(x)\sqrt{1 - \exp\left(-x^2(4/\pi + a_4 x^2)/(1 + a_4 x^2)\right)} \tag{5.43}$$

where $a_4 = 0.140012$ and $Sng(x)$ is a sign function; $Sng(x) = 1$ if $x \geq 0$, and $Sng(x) = -1$ otherwise. Using this discrete scheme, the mean and variance of $T_{ij}(Y_i)$ can be approximated as

$$E(T_{ij}(Y_i)) = E(E(T_{ij}(y \mid Y_i))) \cong \sum_{n=1}^{L} t_{ij}^n \omega_i^n \tag{5.44}$$

$$\begin{aligned} Var(T_{ij}(Y_i)) &= E(Var(T_{ij}(y \mid Y_i))) + Var(E(T_{ij}(y \mid Y_i))) \\ &\cong \sum_{n=1}^{L} (\sigma_{ij}^n)^2 \omega_i^n + \sum_{n=1}^{L} (t_{ij}^n)^2 \omega_i^n - \left(\sum_{n=1}^{L} t_{ij}^n \omega_i^n \right)^2 \end{aligned} \tag{5.45}$$

When link travel times are spatially independent, the travel time covariance in Eq.(5.40) can be approximated as

$$\begin{aligned} Cov(Y_i, T_{ij}(Y_i)) &= E(Y_i * T_{ij}(Y_i)) - E(Y_i) E(T_{ij}(Y_i)) \\ &= \mathrm{E}(Y_i * E(T_{ij}(y \mid Y_i))) - y_i E(T_{ij}(Y_i)) \\ &\cong \sum_{n=1}^{L} t_{ij}^n g_i^n - y_i \sum_{n=1}^{L} t_{ij}^n \omega_i^n \end{aligned} \tag{5.46}$$

where g_i^n is the partial expectation of arrival Y_i during time interval (Δ_{n-1}, Δ_n). The partial expectation g_i^n, in Eq.(5.46), can be calculated as

$$g_i^n = \int_{\Delta_{n-1}}^{\Delta_n} y f_{Y_i}(y)dy = \int_{\Delta_{n-1}}^{+\infty} y f_{Y_i}(y)dy - \int_{\Delta_n}^{+\infty} y f_{Y_i}(y)dy$$
$$= \mathrm{Exp}\left(u(Y_i) + \frac{v(Y_i)}{2}\right)\left\{\Phi_{Y_i}\left(\frac{u(Y_i)+v(Y_i)-\ln(\Delta_{n-1})}{\sqrt{v(Y_i)}}\right) - \Phi_{Y_i}\left(\frac{u(Y_i)+v(Y_i)-\ln(\Delta_n)}{\sqrt{v(Y_i)}}\right)\right\} \quad (5.47)$$

where $f_{Y_i}(y)$ is the PDF of arrival time distribution Y_i. Using Eqs. (5.36-5.47), arrival time distributions at intermediate nodes along the path can be recursively estimated, and consequently the arrival time distribution at destination, denoted by $Y_s(y_r)$, can be obtained. The path travel time distribution $T_u^{rs}(y_r)$ can then be obtained using Eq. (5.3).

5.6. Case Study in Hong Kong

A case study in Hong Kong is presented in this section. As indicated in Chapter 4, real-time traffic information on major urban roads is provided by a Real-time Travel Information System (RTIS) (http://tis.td.gov.hk/rtis/ttis/index/main_partial.jsp) (Tam and Lam, 2008). In RTIS, radio-frequency identification technology is adopted to collect the real-time traffic data. Offline variance-covariance matrices are also generated for travel time estimation. With the use of real-time and offline traffic data, RTIS can provide travel speed estimates, every five minutes, for the whole territory of Hong Kong.

As shown in Fig. 5.3, the RTIS network consists of 1,367 nodes and 3,655 links. In this case study, RTIS travel speed estimates for the whole year of 2009 were collected. Travel speed estimates of 242 weekdays (excluding public holidays) were adopted to generate travel speed distributions for each link and at each five-minute interval. The link travel speed distributions were fitted to normal distributions and the link travel speed covariance between different time intervals was calculated. Consequently, 1,052,640 (24*12*3655) travel speed distributions were generated.

For comparison, the travel time distributions under S-LTM were also constructed. All link travel speed estimates were firstly converted into corresponding link travel time estimates. The travel time distributions were then constructed from converted

link travel time estimates. The link travel time distributions were fitted to lognormal distributions. The S-FIFO property of link travel time distributions was investigated. It was found that almost all link travel time distributions under S-LTM satisfied the S-FIFO condition, with only 219 exceptions.

The non-S-FIFO links (denoted in green) are shown in Fig. 5.3. By examining these non-S-FIFO links, it was found that non-S-FIFO link travel times generally occur during time periods, when link travel times were dramatically reduced (e.g. 8PM or 11PM). In addition, the non-S-FIFO links tend to have a large link length. This result was expected. The violation of S-FIFO property for link travel times is mainly due to the assumption that S-LTM: 'link travel times are assumed to be fixed when travellers travel on the link'. Therefore, the longer link length, the larger the negative impact of such an assumption.

A typical cross boundary trip from Tsuen Wan (Hong Kong) to Shen Zhen (China) was selected for this case study (see Route1 in Fig. 5.3). As shown in Fig. 5.3, the selected Route1 contains two non-S-FIFO links. The path travel time between O-D nodes were generated using α-discrete approximation method ($L_1 = 100$ and $L_2 = 100$). As shown in Table 5.2, the path travel time distributions were generated by using both S-TLM and S-TSM. It can be seen from Table 5.2 that the path travel time distributions under S-TLM may not be S-FIFO. For example, a vehicle departing from origin at 20:01:00 will arrive at destination at 20:52:49 which is 1.08min later than another vehicle departing at 20:05:00. It can also be seen from Table 5.2 that the travel time distributions under S-TSM are S-FIFO.

Table 5.2 Path travel time distributions using both S-LTM and S-TSM

Departure time	S-LTM			S-TSM		
	Arrival time (α = 0.1)	Arrival time (α = 0.5)	Arrival time (α = 0.9)	Arrival time (α = 0.1)	Arrival time (α = 0.5)	Arrival time (α = 0.9)
20:01:00	20:41:02	20:46:09	***20:52:49***	20:38:36	20:41:12	*20:43:38*
20:03:00	20:39:51	20:44:23	*20:49:44*	20:39:49	20:42:26	*20:44:51*
20:05:00	20:41:40	20:46:01	***20:51:44***	20:41:38	20:44:11	*20:46:31*

The accuracy of α-discrete and mean-variance approximation methods was investigated. Fig. 5.4 gives the results of path travel time distribution approximations on Route1 using S-TSM. The departure time was set as 20:01:00. It can be clearly observed from Fig. 5.4 that, in this case study, these two approximation methods generate a closed path travel time distribution.

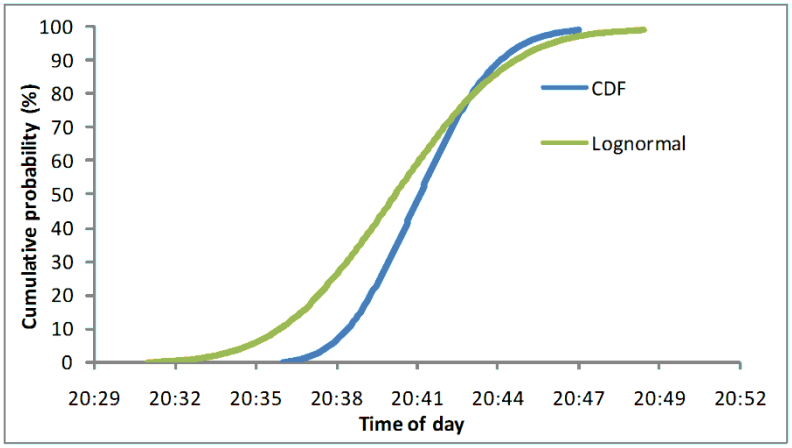


Figure 5.3 Accuracy of path travel time estimation method

5.7. Computational Performance

In this section, computational performance of proposed algorithms is examined, using several large scale networks. The proposed algorithms (*F-TDRSP-A** and *B-TDRSP*) were coded in the Visual C# programming language. The priority queue was implemented using the F-heap data structure (Fredman and Tarjan, 1987). The performance of the *F-TDRSP-A** algorithm depends on the quality of $h(i)$ used. In this study, the shortest distance heuristic function $h(i) = d_{is} / v_{\max}$ was adopted; where d_{is} is the network distance from node i to destination s, and $v_{\max}$ is maximum design speed of the network. The network distance d_{is} can be calculated by using Dijkstra's algorithm from the reverse direction (from destination to origin).

To comparatively evaluate *F-TDRSP-A**, a label-correcting algorithm (referred as *F-TDRSP-LC*) (Nie and Wu, 2009a; Wu and Nie, 2009) was also implemented, using the same programming language. All experiments were conducted in the computer

with a four-core Intel Xeon 3.2GHz CPU (only one core was used) and a Windows Server 2003 operation system.

The computational tests were conducted on four different sized networks. The characteristics of these testing networks are given in Table 5.3. For the RTIS network, link travel time distributions were constructed based on traffic data collected from RTIS, as described in Section 5.6. For the other three networks, travel time distributions of each link were generated by using the travel time pattern of the RTIS links. They were generated by randomly selecting travel speed distributions of one RTIS link; and then multiplying the means and standard deviations of selected RTIS link travel speed distributions.

Table 5.3 Basic characteristics of testing networks

Road networks			Grid networks		
Network	Nodes	Links	Network	Nodes	Links
Hong Kong RTIS	1,367	3,655	G1 (40*50)	2,000	7,820
Chicago Regional	12,982	39,018	G2 (50*100)	5,000	19,700

The computational performance of three different algorithms is given in Table 5.4. All reported results were the average of 100 runs, using different O-D nodes for each run. The 100 O-D nodes were randomly selected for each network and the same set of O-D nodes was used for every test performed on a given network. The link travel time distributions under the S-TSM were calculated based on the *GenerateLinkTime* procedure with $L = 20$. The α-discrete approximation method (Miller-Hooks and Mahmassani, 1998) was adopted to generate arrival time distributions. In the used α-discrete approximation method, arrival time distributions at each node were discretized into $L_1 = 100$ intervals, link travel time distributions were discretized into $L_2 = 20$ intervals and a 'merge sort method' was adopted to sort the discrete elements.

In these tests, the departure time was set as 9AM. The computational performance was evaluated in terms of computational time and the number of generated non-dominated paths in the network. Computational results from Table 5.4

demonstrate that the proposed *F-TDRSP-A** algorithm outperformed the *F-TDRSP-LC* algorithm especially for large networks. For example, the *F-TDRSP-A** algorithm was about 55 (151.96/2.74) times faster than the *F-TDRSP-LC* algorithm in the Chicago regional network. Apparently, this computational advantage was due to the different path searching approaches used in these two algorithms. Similar to conventional algorithms for solving multi-criteria shortest path problem, the *F-TDRSP-LC* algorithm generates all non-dominated paths from the same origin to all other nodes in the network and then determines the reliable shortest path from these generated paths. Therefore, the *F-TDRSP-LC* algorithm does not utilize the monotonic property of objective function in the forward TD-RSPP (Proposition 5.3), generating a large amount of unnecessary non-dominated paths. Based on this monotonic property, the proposed *F-TDRSP-A** algorithm utilizes the A* type path search approach and as such generates much less non-dominated paths during the path search process. It is expected that the computational advantage of *F-TDRSP-A** algorithm will be more obvious when the destination is close to the origin.

Table 5.4 Computational performance of algorithms for solving TD-RSPP

Network	*F-TDRSP-LC*		*F-TDRSP-A**		*B-TDRSP*	
	$\tilde{t}$	$\tilde{n}$	$\tilde{t}$	$\tilde{n}$	$\tilde{t}$	Number of iterations
RTIS	1.69	2.18	0.15	0.50	0.31	2.94
G1	4.26	4.77	0.29	0.58	0.43	2.98
G2	15.42	13.09	1.21	2.23	2.76	3.18
Chicago	**152.1**	34.06	***2.74***	6.02	7.82	3.72

$\tilde{t}$: computational time in seconds; $\tilde{n}$: number of non-dominated paths in 10^3

The *B-TDRSP* algorithm for solving the backward TD-RSPP was also studied. Table 5.4 gives the computational performance of the algorithm, in terms of computational time and number of iterations. The preferred arrival time was set as 9AM. It can be seen from the table that the *B-TDRSP* algorithm required only a few iterations to solve the backward problem in all networks. For example, it required 3.72 iterations of *F-TDRSP-A** and consumed 7.82 seconds. This computational time is slightly better than that required by 3.72 iterations of *F-TDRSP-A** (3.72*2.74=10.19 seconds), since only one reverse search of Dijkstra's algorithm is needed in the

B-TDRSP algorithm. Thus, improving the computational performance of *F-TDRSP-A** algorithm can also significantly improving the performance of *B-TDRSP*.

To improve the performance of the *F-TDRSP-A** algorithm, the mean-variance approximation method can be adopted. It should be noted that in the above tests, link travel time distributions under the S-TSM were calculated based on the *GenerateLinkTime* procedure on the fly. Another possible way, to improve the computational performance, is by pre-calculating these link travel time distributions before carrying out the search process. Table 5.5 gives the computational time of the *F-TDRSP-A** algorithm using these two improvements.

Table 5.5 Computational time of the *F-TDRSP-A** algorithm in seconds

Network	Calculating link travel times on the fly		Pre-calculating link travel times	
	α-discrete method	Mean-variance method	α-discrete method	Mean-variance method
RTIS	0.16	0.020	0.15	0.008
G1	0.29	0.026	0.28	0.011
G2	1.21	0.153	1.20	0.052
Chicago	***2.74***	**0.365**	2.67	***0.151***

It can be observed from Table 5.5 that the mean-variance approximation method can significantly improve the computational performance of the *F-TDRSP-A** algorithm. For example, in the regional network of Chicago, the algorithm based on the mean-variance approximation method was 7.5 times (2.74/0.365) faster than that using the α-discrete method. This computational performance can be further improved by pre-calculating link travel time distributions. For instance, it can further speed up the algorithm by 141% ((0.365-0.151)/0.151) in the Chicago regional network. It should be noted that, if α-discrete approximation method is adopted, the pre-calculation of link travel time distributions has only a slight improvement (2.6% in Chicago regional network) on the algorithm's computational performance. This is because the link travel time distributions, in the α-discrete approximation method, are needed to be discretized into a set of equal-probability elements. The link travel time distributions, generated by the *GenerateLinkTime* procedure, have been dicretized. Thus no additional discretization is needed when generating path travel

time distributions based on the α-discrete approximation.

Using the same O-D pairs, the computational results of two approximation methods were investigated. Fig. 5.4 illustrates the travel time budget differences calculated by these two approximation methods for the same networks with various network sizes. In the figure, the travel time budget ratio is defined by the ratio of the travel time budget calculated by mean-variance approximation method to that calculated by α-discrete approximation. It can be observed from the figure that these two approximation methods can generate closed travel time budgets (around 98%) for all networks. The mean-variance approximation method tends to slightly underestimate the travel time budget.

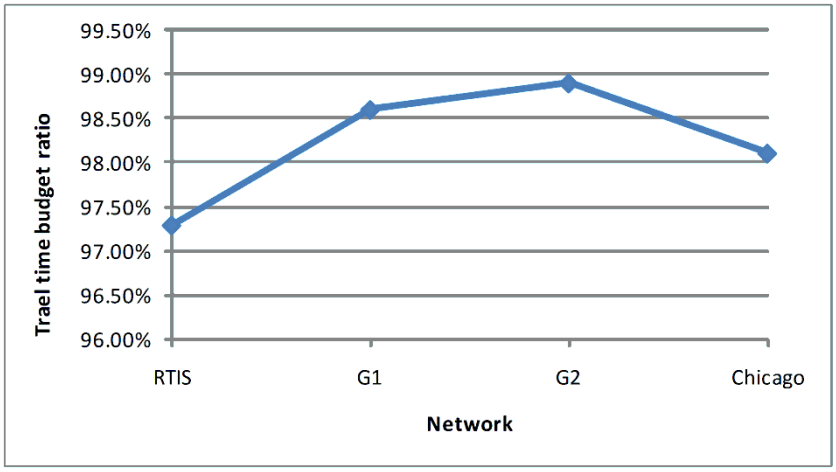


Figure 5.4 Comparison of computational results using α-discrete and mean-variance approximations

5.8. Summary

The investigation of two time-dependent reliable shortest path problems (TD-RSPP) has been described in this chapter. The forward problem is to determine the earliest arrival time and associated reliable shortest path for a given departure time; while the backward problem is to determine the latest departure time and associated reliable shortest path for a given preferred arrival time.

In this chapter, a new stochastic travel speed model (S-TSM) to represent STD link travel times, has been proposed. In S-TSM, the travel speed distributions of a link are assumed to be fixed within each time interval, and can be varied as vehicles travel on the link. Based on this model, a discrete method has been proposed to generate link travel time distributions from the corresponding STD travel speed distributions. It was proved that the link travel time distributions, under S-TSM, the stochastic first in first out (S-FIFO) property. A case study, with the use of traffic data from real-time traveller information system (RTIS) in Hong Kong, demonstrated that link travel time distributions using traditional stochastic link travel time model (S-LTM) may not be S-FIFO consistent; and may have a significant travel time bias compare to those using S-TSM.

Using S-TSM, the multi-criteria A* algorithm (named *F-TDRSP-A** algorithm) has been developed to solve the forward TD-RSPP. It was proved that, when all the link travel times are S-FIFO, the reliable shortest path in STD networks can be determined as one of the non-dominated paths under the first-order stochastic dominant (FSD) condition. It was shown in this chapter that the TD-RSPP is not reversible. The backward TD-RSPP cannot be easily solved using the algorithms designed for the forward problem by the search from destination to origin. The backward TD-RSPP in this chapter has been formulated and solved using an equivalent two-stage approach (named *B-TDRSP*). The first stage is to determine the optimal departure time for an initial path between O-D nodes. The second stage, after the initial departure time has been determined, is to find the reliable shortest path by solving the forward TD-RSPP. It has been shown that the optimality of this two stage method can be theoretically proved.

Computational performances showed that the proposed *F-TDRSP-A** and *B-TDRSP* algorithms can solve the forward and backward TD-RSPP in large-scale networks within satisfactory computational time. The *F-TDRSP-A** algorithm has a significant computational advantage over previous label-correcting method. The *B-TDRSP* algorithm required only a few iterations of forward search to solve the backward problem. The computational performance of these two algorithms can be improved by using an efficient method to generate path travel time distribution (e.g.

mean-variance approximation method) and pre-calculating link travel time distribution under S-TSM.

The solution algorithms proposed in this chapter only consider travel time temporal correlations, but spatial correlations have not been considered. The solution algorithms for solving TD-RSPP can be utilized to develop real-world ATIS-based routing system. The development of ATIS-based routing system with consideration of spatiotemporal correlations is discussed in Chapter 6.

6. Reliable Path Searching System: Case Study in Hong Kong

After introducing efficient reliable shortest path algorithms in Chapter 5, this chapter moves one step further to the implementation of a real-world route guidance system (RGS) in Hong Kong. The developed RGS, named reliable path searching system (RPSS), aims to aid road users making route choices under travel time uncertainties.

In the developed RPSS, the travel time spatiotemporal correlations are explicitly considered. The real-time and historical traffic information, from real-time travel information system (RTIS), is incorporated for real-time RGS operations. The developed RPSS is implemented using the Web Service technique. It can provide three service types to users, including reliable path searching service, location-based service and map presentation service. In this chapter, the architecture and the unique features of the developed RPSS are introduced in detail. The representation of complex road features for real-world RGS application is also discussed.

The remainder of this chapter is organized as follows. The background of RPSS is briefly described in the next section. The representation of real-world road features is briefly discussed in Section 6.2. The solution algorithm for reliable shortest path finding in real road networks is presented in Section 6.3. The architecture of the RPSS is introduced Section 6.4. The method to estimate travel time distributions is described in Section 6.5. The experimental and computational results of the RPSS are reported in Section 6.6. Finally, a summary of this chapter is given in Section 6.7.

6.1 Background

As indicated in previous chapters, online traffic information on major urban roads of Hong Kong is provided by a Real-time Travel Information System (RTIS) at five minute intervals (Tam and Lam, 2008). RTIS enables the real-time traffic

information to be collected by automatic vehicle identification (AVI) technology and is adopted for traffic time estimation. The online traffic information is then disseminated to travellers through various media, including variable message signs, mobile phones and the internet website portal (http://tis.td.gov.hk/rtis/ttis/index/main_partial.jsp). A driving route searching service (DRSS) is, also provided to road users of Hong Kong through the website portal (http://drss.td.gov.hk/drss). DRSS can determine and advise the users with the appropriate routes in real-time, according to users' routing criterion (minimum expected travel time, minimum travel distance, or minimum toll charge).

However, a limitation of DRSS is the unrealistic assumption that travel times in the road network are deterministic. As noted in Chapter 2, travel times are stochastic and time-dependent due to random demand fluctuations and capacity degradations. Under travel time uncertainties, travellers tend to choose reliable shortest paths for their travels, so that they can arrival at destinations with a given on-time arrival probability (referred as travel time reliability in the literature). Therefore, travel time reliability concerns are necessary inclusions in RGS applications. Responding to this need, RPSS (reliable path searching system) is developed to aid road users of Hong Kong making route choice decision in face of travel time uncertainty.

6.2 Network Representation

The representation of road networks is the foundation for developing RGS applications. As adopted in Chapters 5, the road network is represented as a time-expanded graph $G = (N, A, \Omega)$, consisting a set of nodes N, a set of links A and the period of interest Ω. The period of interest Ω is considered as a set of discrete times $\{..., \Delta_n = n\Delta, ...\}$, where n is an integer and Δ is the time interval. In this network model, nodes generally correspond to street intersections, while links correspond to street segments between two intersections. Each node i has a set of successor nodes $SCS(i) = \{j : a_{ij} \in A\}$ and a set of predecessor nodes $PDS(i) = \{w : a_{wi} \in A\}$. Each network link $a_{ij} \in A$ is one-way from its tail node $n_i \in N$ to

head node $n_j \in N$. A two-way street can be represented as two separate opposing directed links a_{ij} and a_{ji}. Impedances of link a_{ij} can be maintained as a set of link attributes $Z(a_{ij})$. It should be noted that this 'node-based' network model is non-planar without requirement of a node at each link crossing, especially when modeling underpasses and overpasses.

This node-based network model is simple and can be easily stored and manipulated in the memory using forward star or adjacent list data structures (Ahuja et al., 1993). Such node-based network models, however, may not be the best model for developing the RGS applications in real road networks. Firstly, a journey may not always starts from a network node and ends at another network node. It is common for the origin and destination (O-D) pair of the journey (e.g. schools, restaurants, shopping malls) to be located on network links.

Secondly, turn restrictions are not explicitly represented in this node-based network model. For instance, right turns in Hong Kong are very often prohibited, due to traffic controls implemented at street intersections. Ignoring these turn restrictions can generate infeasible paths for the RGS users. An illustrative example is given in the Fig. 6.1. The O-D pair of a journey is set as n_1 and n_5. A feasible path between this O-D pair is $p_1^{15} = a_{12} \oplus a_{23} \oplus a_{32} \oplus a_{25}$, going through four subsequent links. If turn restrictions are ignored, an illogical path $p_1^{15} = a_{12} \oplus a_{25}$ passing through two links a_{12} and a_{25} will be recommended. Therefore, modifications to the above node-based network model are required for developing RGS applications.

A link-based network model, initially proposed by Ziliaskopoulos and Mahmassani (1996), is modified to represent real road networks with stochastic time-dependent (STD) travel times in this study. A road network is represented as a graph $\mathrm{G} = (\mathrm{N}, \mathrm{A}, \Psi, \Omega)$, consisting of a set of nodes N, a set of links A, a set of allowed movements Ψ and the period of interest Ω. Similar to above node-based model, each link $a_{ij} \in A$ has a tail node $n_i \in N$, a head node $n_j \in N$ and a set of link

attributes $Z(a_{ij})$. Three attributes $Z(a_{ij}) = (\tau_{ij}, d_{ij}, T_{ij})$ are maintained for each link a_{ij}, where τ_{ij} is the toll charge, d_{ij} is the link length, and T_{ij} is the link travel time. Without loss of generality, τ_{ij} and d_{ij} are assumed to be deterministic; while T_{ij} is assumed to be a STD variable. The statistical distributions of link travel times can vary with the period of interest. $T_{ij}^{n}(y_i)$ denotes the travel time distribution within a time interval $y_i \in (\Delta_{n-1}, \Delta_n)$.

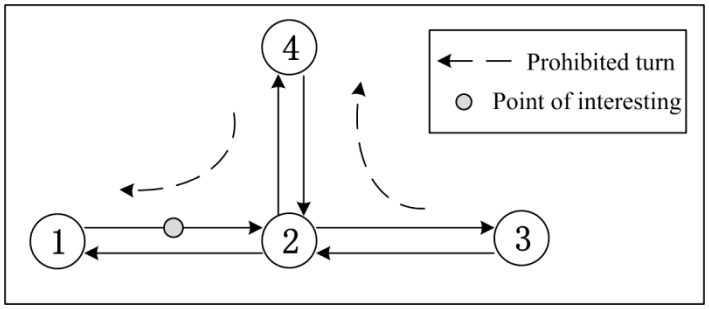


Figure 6.1 Turn restrictions at a simple street intersection

Elements in Ψ represent allowed movements at street intersections. Each movement $\psi_{wij} \in \Psi$ at node i has a star link a_{wi}, an end link a_{ij} and a set of attributes $Z(\psi_{wij})$. A movement $\psi_{wij} \notin \Psi$ means that the movement is restricted at node i (e. g. no left-turn or no U-turn).

In order to present essential ideas easily in this chapter, it is assumed that link travel time spatial correlations are restricted to only adjacent links (k = 1). The spatial correlations among k-neighouring links can be easily incorporated through the method presented in Chapter 4 by constructing two-level hierarchical network. Using this assumption, travel time spatial correlation between adjacent links a_{wi} and a_{ij}, denoted by $\mathrm{cov}(T_{wi}, T_{ij})$, can be represented as a time-dependent variable varying with the period of interest and maintained as a turn attribute. $\mathrm{cov}(T_{wi}^{n}, T_{ij}^{n})$ denotes travel time covariance for vehicles arriving at node i at a time instant $y_j \in (\Delta_{n-1}, \Delta_n)$. The link-based adjacent list data structure, proposed by Gutierrez and Medaglia (2008), can be modified to store and manipulate the network data in

the memory. Fig. 6.2 illustrates the data structure for the simple network sketched in Fig. 6.1.

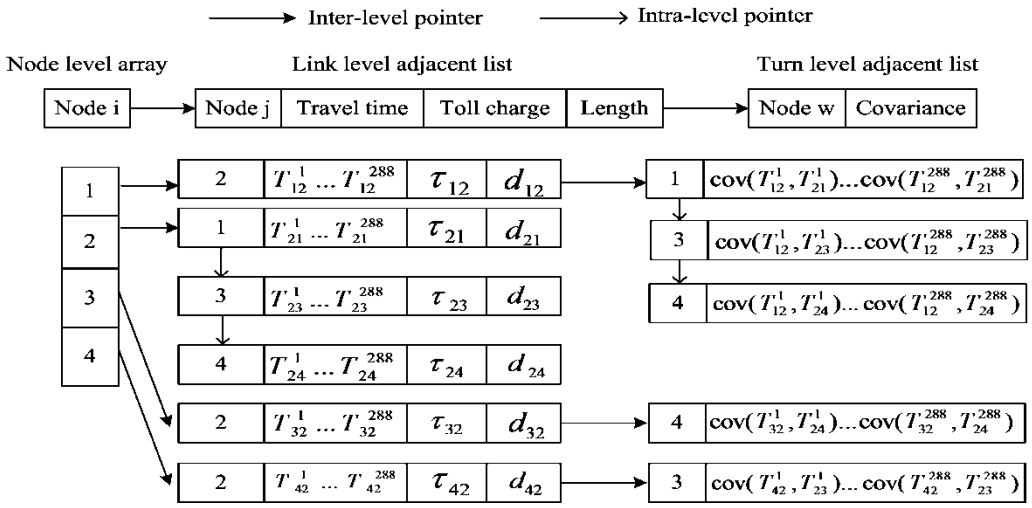


Figure 6.2 Data structure in the memory

In this study, the linear reference technique is adopted to represent the location of POI (Miller and Shaw, 2001). Given a POI_i in the network, its location can be defined as $POS(POI_i) = (a_{ij}, \theta)$; where a_{ij} is the link that POI_i located, and $\theta \in [0,1]$ indicates a relative position on the link a_{ij}. For example, POIs located at the beginning, middle and end of the link a_{12} in Fig. 6.1, can be respectively represented as $(a_{12}, 0)$, $(a_{12}, 0.5)$ and $(a_{12}, 1)$. The distance from POI_i to tail node i can be calculated by $\theta * d_{ij}$; while the distance from POI_i to head node j can be calculated by $(1-\theta) * d_{ij}$.

Let $\hat{a}_{\ell j}$ be a part of link a_{ij}. $\hat{d}_{\ell j}$, $\hat{\tau}_{\ell j}$ and $\hat{T}_{\ell j}$ denote link length, toll charge and travel time of the partial link $\hat{a}_{\ell j}$ respectively; and $\mathrm{cov}(\hat{T}_{\ell j}, T_{jw})$ denotes the travel time correlation between partial link $\hat{a}_{\ell j}$ and link a_{jw}. It is assumed that $\hat{\tau}_{\ell j}$ is equal to τ_{ij}, and $\hat{T}_{\ell j}$ and $\mathrm{cov}(\hat{T}_{\ell j}, T_{jw})$ are proportional to the link length (i.e. $\hat{T}_{\ell j} = T_{ij} * \hat{d}_{\ell j} / d_{ij}$ and $\mathrm{cov}(\hat{T}_{\ell j}, T_{jw}) = \mathrm{cov}(T_{ij}, T_{jw}) * \hat{d}_{\ell j} / d_{ij}$).

6.3 Generalized Reliable Shortest Path Problem

Let $p_u^{rs} = \{\hat{a}_{rj}, a_{jw}, ..., \hat{a}_{qs}\}$ be a path between the O-D pair. The O-D pair can either be at network node or on network links. If the origin is on link a_{ij}, $\hat{a}_{rj}$ corresponds to the partial link from the origin r to the head node j. Similarly, if the destination s is on link a_{qm}, $\hat{a}_{qs}$ correspond to the partial link from tail node q to destination s. Let τ_i^{rs}, d_i^{rs} and $T_u^{rs}(y_r)$ be the respective toll charge, the travel distance and the travel time of a path p_i^{rs}.

The toll charge and travel distance can be easily calculated as

$$\tau_u^{rs} = \sum_{a_{ij} \in A} \tau_{ij} \delta_{ij}^{rs,u} \tag{6.1}$$

$$d_u^{rs} = \sum_{a_{ij} \in A} d_{ij} \delta_{ij}^{rs,u} \tag{6.2}$$

where $\delta_{ij}^{rs,u}$ is the path-link incidence variable; $\delta_{ij}^{rs,u} = 1$ means that the link a_{ij} is on the path p_u^{rs}, and otherwise, $\delta_{ij}^{rs,u} = 0$. It is assumed that waiting is not allowed on the network except for the destination. As formulated in Chapter 5, the path travel time distribution can then be calculated as

$$T_u^{rs}(y_r) = \sum_{(i,j) \in A} T_{ij}(Y_i)\delta_{ij}^{rs,u} \tag{6.3}$$

where y_r is the departure time at origin r, and Y_i is the arrival time at an intermediate node i. Given a preferred arrival time y_s, the travel time budget (denoted by b) that travellers assigned trips can be expressed as

$$b = y_s - y_r \tag{6.4}$$

By incorporating toll charge and petrol cost, a generalized route dis-utility, denoted by $\varphi_{p_u^{rs}}$, can be written as

$$\varphi_{p_u^{rs}} = b + (\tau_u^{rs} + d_u^{rs} VOD)/VOT \tag{6.5}$$

where the parameter VOD is for converting travel distance d_u^{rs} into the petrol cost; and the weighting factor VOT is the value of time for converting the toll charge and the petrol cost into the time unit.

Utilizing the above generalized route dis-utility, time-dependent reliable shortest path problem (TD-RSPP), presented in Chapter 5, can be modified as the following optimization problem:

$$\text{Min} \quad \varphi_{p_u^{rs}} = b + (\tau_u^{rs} + d_u^{rs} VOD)/VOT \tag{6.6}$$

Subject to

$$\Pr(T_u^{rs}(y_r) \le b) \ge \alpha \tag{6.7}$$

$$\psi_{wij} \in \Psi, \quad \forall \psi_{wij} \in p_u^{rs} \tag{6.8}$$

$$\sum_{j \in SCS(i)} \delta_{ij}^{rs,u} - \sum_{w \in PDS(i)} \delta_{wi}^{rs,u} = \begin{cases} 1 & \forall i = r \\ 0, & \forall i \ne r; i \ne s \\ -1 & \forall i = s \end{cases} \tag{6.9}$$

$$\delta_{ij}^{rs,u} \in \{0,1\}, \quad \forall a_{ij} \in A \tag{6.11}$$

Eq. (6.6) is the dis-utility that travellers want to minimize. Eq. (6.7) defines the probabilistic constraint that ensures the on-time arrival probability is greater or equal to a pre-determined threshold α. Eq. (6.9-6.11) ensures that the links on the optimal path are feasible.

As indicated in Chapter 5, the path travel time $T_u^{rs}(y_r)$ is a stochastic process and can be generated by the mean-variance approximation method, referred to Eqs. (5.36-5.47), when path travel time follows a lognormal distribution. This mean-variance approximation method is extended to incorporate travel time spatial correlations among adjacent links. Let $p_u^{rj} = p_u^{rw} \oplus a_{wi} \oplus a_{ij}$ be a path from origin r to node j passing through path p_u^{rw} and links a_{wi} and a_{ij}. Suppose that the arrival times at nodes w and i (denoted by Y_w and Y_i) are known. The mean and variance of Y_j, denoted by y_j and $\sigma_{Y_j}^2$, can be calculated as

$$y_j = y_i + E(T_{ij}(Y_i)) \tag{6.12}$$

$$\sigma_{Y_j}^2 = \sigma_{Y_i}^2 + Var(T_{ij}(Y_i)) + 2Cov(T_{wi}, T_{ij}(Y_i)) + 2Cov(Y_w, T_{ij}(Y_i)) \quad (6.13)$$

where y_i and $\sigma_{Y_i}^2$ respectively are mean and variance of Y_i; $E(T_{ij}(Y_i))$ and $Var(T_{ij}(Y_i))$ respectively are mean and variance of conditional link travel time $T_{ij}(Y_i)$ and they can be calculated by Eqs. (5.41-5.45); $Cov(T_{wi}, T_{ij}(Y_i))$ is the travel time spatiotemporal correlations among adjacent links a_{wi} and a_{ij}; and $Cov(Y_w, T_{ij}(Y_i))$ is the temporal correlation among link a_{ij} and sub-path p_u^{rw}. The travel time spatiotemporal correlation component, $Cov(T_{wi}, T_{ij}(Y_i))$, can be approximated as

$$\begin{aligned} Cov(T_{wi}, T_{ij}(Y_i)) &= E(T_{wi} * T_{ij}(Y_i)) - E(T_{wi}) * E(T_{ij}(Y_i)) \\ &= \mathrm{E}(E(T_{wi}(y \mid Y_i) * T_{ij}(y \mid Y_i))) - E(E(T_{wi}(y \mid Y_i))) * E(E(T_{ij}(y \mid Y_i))) \\ &\cong \sum_{n=1}^{L} \mathrm{cov}(T_{wi}^n, T_{ij}^n)\omega_i^n \end{aligned} \quad (6.14)$$

where ω_i^n is the probability mass function of arrival time Y_i during time interval (Δ_{n-1}, Δ_n) and can be calculated by Eqs. (5.41-5.43). The temporal correlation component, $Cov(Y_w, T_{ij}(Y_i))$, can be approximated by

$$\begin{aligned} Cov(Y_w, T_{ij}(Y_i)) &= E(Y_w * T_{ij}(Y_i)) - E(Y_w)E(T_{ij}(Y_i)) \\ &= \mathrm{E}(Y_w * E(T_{ij}(y \mid Y_i))) - y_w E(E(T_{ij}(y \mid Y_i))) \\ &\cong \sum_{n=1}^{L} t_{ij}^n g_w^n - y_w \sum_{n=1}^{L} t_{ij}^n \omega_i^n \end{aligned} \quad (6.15)$$

where t_{ij}^n is mean travel time of link a_{ij} during time interval (Δ_{n-1}, Δ_n); and g_w^n is the partial expectation of arrival time Y_w during time interval (Δ_{n-1}, Δ_n). The partial expectation g_w^n can be calculated by Eq. (5.47) using Y_w instead of Y_i.

Solution algorithms (*F-TDRSP-A** and *B-TDRSP*), presented in Chapter 5, can be easily extended to solve the generalized TD-RSPP. To consider turn restrictions and spatiotemporal correlations, the first-order stochastic dominant (FSD) condition should be extended as below.

Let $P^{r,ij} = \{p_u^{r,ij}, ..., p_v^{r,ij}\}$ be a set of paths between the O-D pair passing the same link a_{ij}.

Definition 1. Given two paths $p_u^{r,ij} \neq p_v^{r,ij} \in P^{r,ij}$, $p_u^{r,ij}$ dominates $p_v^{r,ij}$, if $p_u^{r,ij}$ and $p_v^{r,ij}$ satisfy $g_u^{r,ij} < g_v^{r,ij}$ and $\Phi^{-1}_{p_u^{r,ij}(t_r)}(\lambda) < \Phi^{-1}_{p_v^{r,ij}(t_r)}(\lambda)$, $\forall \lambda \in (0,1)$.

Based on extended FSD conditions, *F-TDRSP-A** algorithm can be modified as a link-based shortest path approach. The optimal path can be determined through recursive path extensions through allowed movements. Labels of non-dominated paths are maintained at links instead of nodes. The heuristic value $F(p_u^{r,ij})$ for a non-dominated path $p_u^{r,ij}$ should also be modified to incorporate toll charges and patrol costs as

$$F(p_u^{r,ij}) = h(j) + \varphi_{p_u^{rs}} \tag{6.16}$$

where $h(j)$ is an estimated dis-utility from node j to destination s. The Euclidean distance function $h(j) = e_{js}(1/V_{\max} + VOD/VOT)$ can be adopted; where e_{js} is the Euclidean distance from node j to destination s and $V_{\max}$ is maximum travel speed of the network.

In addition, when the O-D pair of the journey is not at network nodes, two dummy partial links, $\hat{a}_{rj}$ and $\hat{a}_{js}$, can be constructed for solving the generalized forward TD-RSPP. The detailed step of the modified *F-TDRSP-A** is given below. It should be noted that, with the modified *F-TDRSP-A** algorithm, *B-TDRSP* algorithm can be directly used to solve the generalized backward TD-RSPP problem.

---

*Modified F-TDRSP-A**

Inputs: O-D pair, confidence level α, and departure time t_r

Returns: the reliable shortest path

Step 1. Initialization:

If origin r is a network node

For each successor node $j \in SCS(r)$

Create a path $p_u^{r,rj}$ at link a_{rj}.

Set $T_u^{r,rj}(y_r) \coloneqq T_{rj}(y_r)$, and calculate $h(j)$ and $F(p_u^{r,rj})$.

Set $P^{r,rj} \coloneqq \{p_u^{r,rj}\}$ and $SE \coloneqq SE \cup \{p_u^{r,rj}\}$.

End For

End if

If origin r is on a network link a_{ij}

Construct a dummy link $\hat{a}_{rj}$ and calculate link attributes $Z(\hat{a}_{rj})$.

Create a path p_u^{rj}, set $T_u^{rj}(y_r) \coloneqq T_{rj}(y_r)$, and calculate $h(j)$ and $F(p_u^{rj})$.

Set $SE \coloneqq \{p_u^{rj}\}$.

End if

If destination s is on link a_{ij}

Construct a dummy link $\hat{a}_{is}$ and calculate link attributes $Z(\hat{a}_{is})$.

End if

Step 2. Path selection:

If $SE = \phi$, then stop; otherwise, continue.

Select $p_u^{r,wi}$ at the top of SE and remove $p_u^{r,wi}$ from SE.

If $i = s$, then stop; otherwise continue.

Step 3. Path extension:

For every movement ψ_{wij} emanating from selected link a_{wi}

If destination s is on link a_{ij}

Construct a temporary path $p_u^{r,is} \coloneqq p_u^{r,wi} \oplus \hat{a}_{is}$.

Else if

Construct a temporary path $p_u^{r,ij} \coloneqq p_u^{r,wi} \oplus a_{ij}$.

End if

Generate path travel time distribution $T_u^{rj}(y_r)$ using Eq. (6.12-6.15)

Calculate $h(j)$ and $F(p_u^{r,ij})$.

If $p_u^{r,ij}$ is a non-dominated path under the extended FSD condition, then insert

$p_u^{r,ij}$ into $P^{r,ij}$ and SE and remove all paths dominated by $p_u^{r,ij}$ from

$P^{r,ij}$ and SE (using CheckDominance procedure presented in Chapter 5).

End for

Goto Step 2.

6.4 System Architecture

This section presents the architecture of RPSS. Fig. 6.3 illustrates its concept framework. As shown in the figure, the RPSS comprises four key components: Service and Application Provider (SAP), Content and Data Provider (CDP), Website Portal (WP) and Clients.

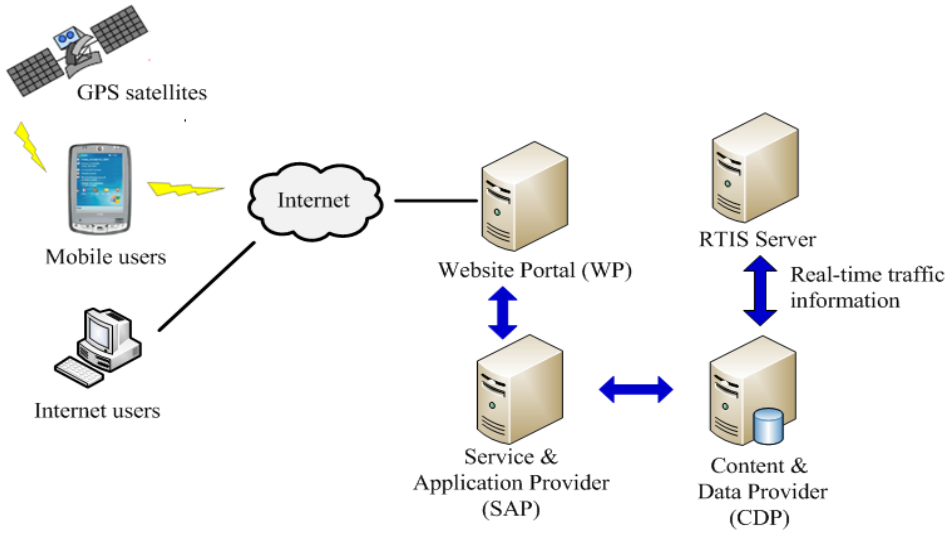


Figure 6.3 The system framework of RPSS

SAP acts as an application server for proceeding and answering core service requests. The core services implemented in the RPSS include location-based services (LBS), reliable path searching services, and map presentation services. LBS provides users with capabilities to find the nearest or a specific point of interest (POI) (e.g. “where is The Hong Kong Polytechnic University (PolyU)?”, “where is the nearest hotel to PolyU?”). The reliable path searching services can determine the optimal route between a specified O-D pair, satisfying travellers’ multiple routing criteria towards travel time reliability, travel distance and toll charge. The origin and destination

(O-D) pair can be determined through an LBS search or acquired by GPS (global positioning system) or user inputs. The map presentation services provide a way to render outputs of the LBS and the reliable path searching services on geographical maps.

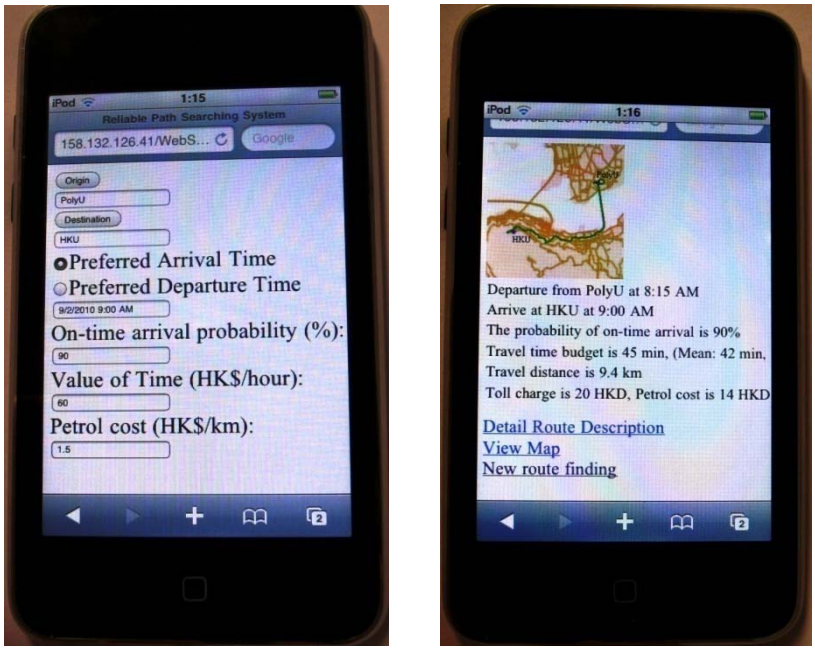
In this study, SAP is implemented using a Web Service technique which is an open industrial standard to allow interoperation across programming languages, platforms and operating systems. The Web Service offers protocols and mechanisms for rapidly publishing, discovering, and invocating services via internet (Friis-Christensen et al., 2009). With the availability of the capabilities of the Web Service, SAP can provide a flexible and loosely-integrated platform, not only for the implementation of RGS applications but also for the development of other ITS applications (e.g. logistic applications).

CDP maintains the network data needed for the implementation of SAP. The data stored in CDP server consists of GIS (geographical information system) maps, network topology data and traffic information The GIS maps include a base map, a road network map and a set of POI maps (e.g. hotels, restaurants and banks,.). The traffic data mainly consist of travel time estimates and traffic incident data. The travel time estimates are link travel time distributions generated by using the real-time and the historical traffic information collected from RTIS. The method for generating link travel time distributions is described in the following section.

WP provides interfaces for road users to access the services through the internet or mobile phones. When users request a service, WP will pass the request to SAP, and then transfer SAP processing results to the user. Fig. 6.4 illustrates a typical user interface for mobile users to access the reliable path services.

As shown in Fig. 6.4(a), the O-D pair can be set by entering their addresses or the names of POI. The routing parameters (on-time arrival probability, value of time and petrol cost etc.) can also be inputted in this interface. By entering the preferred arrival time, the reliable path searching services can determine the latest departure time and the associated route to users. The search results of this forward TD-RSPP

are returned to the user as shown in the Fig. 6.4(b). A detailed textual route description can also be generated to the user, if the user clicks the link “Detail Route Description” in this interface. As shown in Fig. 6.4 (a), the RPSS also provides a searching function for backward TD-RSPP. Given the preferred departure time, this type of reliable path searching is to determine latest arrival time required to satisfy the travel time reliability threshold.



(a) (b)

Figure 6.4 User interface for the reliable routing service (a) input routing criteria (b) reliable path searching results

6.5 Estimation of Travel Time Distributions

As indicated above, RTIS provides link travel time estimates, at five minute intervals, for the 1,367 node and 3,655 link Hong Kong road network and allows 11,849 movements. The historical and real-time travel times from the RTIS are collected to estimate travel time distributions. As link travel times are assumed to follow lognormal distributions, only mean and variance of travel times are estimated. The toll rates of tunnels / bridges are also collected from the Transport Department of Hong Kong Government.

In this study, the historical travel time estimates for the whole year of 2009 are collected to generate link travel time distributions. To take into account travel time variations due to time-of-day and day-of-week, the traffic data are disaggregated into a set of sub-groups. Days in a week are divided into six categories. Traffic data in each weekday forms an individual group, while the travel time in Saturday and Sunday as well as the public holidays are aggregated as one group because they have somewhat similar travel patterns. For each group, link travel speed distributions are firstly generated at five-minute interval. Using S-TSM presented in Chapter 5, these travel speed distributions are then converted to corresponding travel time distributions with S-FIFO (stochastic first in first out) property. The spatial correlations among adjacent links are generated for every five minute intervals.

The real-time traffic data from RTIS are employed to improve the estimation of link travel time distributions for current time periods. The mean travel time for the current time period can be adjusted as

$$t_{ij}^{c}(t) = (1-w)t_{ij}^{h}(t) + wt_{ij}^{r} \tag{6.17}$$

where $t_{ij}^{c}(t)$ is the mean travel time at current (five minutes) time period, $t_{ij}^{h}(t)$ is the historical mean travel time at the same time period**,** t_{ij}^{r} is the real-time travel time estimate at five minute intervals and w is a pre-defined weighting factor. The standard deviation of link travel time for the current time period denoted by $\sigma_{ij}^{c}(t)$, can be estimated by assuming that the coefficient of variation (CV) of link travel time keeps constant:

$$\sigma_{ij}^{c}(t) = t_{ij}^{c}(t) * CV_{ij}^{h}(t) = t_{ij}^{c}(t) * \sigma_{ij}^{h}(t) / t_{ij}^{h}(t) \tag{6.18}$$

Similarly, it is also assumed that travel time correlation coefficient between two adjacent links a_{ij} and a_{jw} is constant The spatial covariance between these two links, denoted by $\text{cov}(X_{ij}^{c}, X_{jw}^{c})$, can then be updated as

$$\text{cov}(X_{ij}^{c}, X_{jw}^{c}) = \rho_{ijw}^{h}(t) * \sigma_{ij}^{c}(t) * \sigma_{jw}^{c}(t) \tag{6.19}$$

where $\rho_{ijw}^{h}(t) = \text{cov}(X_{ij}^{h}, X_{jw}^{h}) / \sigma_{ij}^{h}(t) / \sigma_{jw}^{h}(t)$ is the travel time correlation coefficient calculated using the historical data.

6.6 Experimental Results

This section reported the reliable shortest path finding results on the Hong Kong road network. In Hong Kong, cross harbor traffic is one of the most significant travel patterns. As shown in Fig. 6.5, there are three cross-harbor tunnels in Hong Kong. They have significantly different toll rates, mean travel times and travel time variances. The Cross Harbor Tunnel (CHT) is the cheapest (HK$ 20) but with the largest mean travel time and travel time variance. The Western Harbor Crossing (WHC) is the most reliable and the mean travel time is also the least, but the toll charge is the most expensive (HK$ 50). The Eastern Harbor Crossing (EHC) toll charge is slightly higher than that of the CHT (HK$ 25), but its travel time variation is similar to that of the WHC.

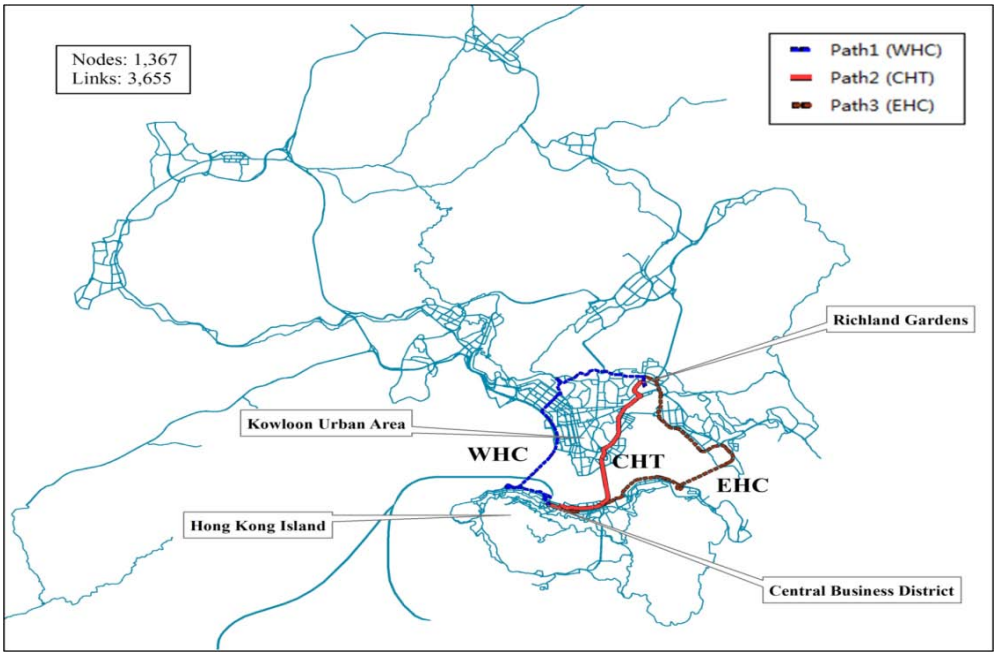


Figure 6.5 Results of reliable route searching from RG to CBD

A typical morning commute from the residential zone, Richland Gardens (RG) in the Kowloon urban area, to Central Business District (CBD) on Hong Kong Island is considered in this case study. The preferred arrival time was set at 9AM on 26 August 2010 (Thursday). Fig. 6.5 and Table 6.1 provided the results of the reliable

route finding from the RG to the CBD. As shown in Fig. 6.5, when the value of time was small (VOT = HK$ 60 / Hour), travellers preferred to use Route2 passing through the CHT which was the cheapest route (HK$ 36.3) but with the highest travel time variation.

Under this scenario, travellers chose appropriate departure times based on their on-time arrival attitude of. As shown in Table 6.1, risk-seeking travellers ($\alpha < 50\%$) tended to assign a small travel time budget for their trips. For example, when $\alpha = 10\%$, risk-seeking travellers assigned only 25.1 min travel time budget (departed at 8:36:22), which was 10.0 min less than expected travel time 35.1 min. With an increase in confidence level of on-time arrival, travellers departed from the origin earlier. As shown in Table 6.1, risk-neutral travellers ($\alpha = 50\%$) assigned 32.3 min travel time budget close to the expected travel time 33.2 min. When $\alpha = 50\%$, the travel time budget is essentially the median value of journey time. Since most link travel times followed a lognormal distribution, the median value was slightly smaller than the expected travel time. To ensure higher probability of on-time arrival, risk-averse travellers ($\alpha > 50\%$) assigned a larger travel time budget for their trips. For example, when $\alpha = 99\%$, risk-averse travellers started their trips at 8:13:53; and this travel time budget is about 14.9 min larger than the expected travel time 31.2 min. Interestingly, it is seen from Table 6.1 that when travellers started their trips earlier, the expected travel time and travel time variations decreased.

Another important observation from Table 6.1 is that with an increase of the value of time (VOT), travellers were more willing to use reliable but expensive routes to ensure the probability of on-time arrival. For instance, to obtain the same probability of on-time arrival $\alpha = 99\%$, risk-averse travellers used Route3 passing through EHC when VOT = HK$ 130 / Hour; used Route1 passing through WHC when VOT =HK$ 200 / Hour. In addition, it is seen that when VOD becomes large (VOD = HK$ 9 / Km is the cost by taxi in Hong Kong), travellers used Route2 again since it is the shortest route between OD nodes. However, when VOT = HK$ 300 / Hour, risk-averse travellers were willing to pay extra cost of HK$ 60.1 for using Route1 to save 12.2 min travel time.

Table 6.1 Departure time and route choices between RG and CBD

Probability on-time arrival α (%)	*VOT* (HK$ / Hour)	*VOD* (HK$ / Km)	Route	Departure time	Travel time budget (min)	Toll charge + operational cost (HK$)	Expected travel time (min)	Std. dev. of travel time (min)
10	60	1.5	Route2	***8:34:54***	**25.1**	36.3	**35.5**	8.9
30	60	1.5	Route2	***8:30:43***	29.3	36.3	34.1	8.2
50	60	1.5	Route2	***8:27:41***	**32.3**	36.3	**33.2**	7.8
70	60	1.5	Route2	***8:24:39***	35.4	36.3	32.4	6.8
90	60	1.5	Route2	***8:20:37***	39.4	36.3	31.6	5.9
99	60	1.5	Route2	***8:13:53***	**46.1**	36.3	**31.2**	5.5
99	***130***	1.5	***Route3***	*8:23:47*	36.2	52.4	29.4	2.7
99	***200***	1.5	***Route1***	*8:32:27*	27.6	63.1	23.6	1.6
99	200	***9.0***	***Route2***	*8:13:53*	39.8	117.6	31.2	5.5
99	***300***	9.0	***Route1***	*8:32:27*	27.6	178.7	23.6	1.6

The computational performance of reliable path searching services was also investigated. The generalized reliable path algorithms were coded by C# programming language. The scan eligible (*SE*) was implemented using the F-heap data structure (Fredman and Tarjan, 1987). All experiments were conducted in the computer with a four-core Intel Xeon 3.2GHz CPU (only one core was used) and a Windows Server 2003 operation system. In this case study, 100 O-D nodes were randomly selected and average computational time of 100 runs was calculated. It was found that the forward search only required 13.2 milliseconds to determine the optimal route and earlier arrival time. The backward search requires 51.4 milliseconds (about 4.2 iterations of forward search). From the result, it is evident that the developed reliable path searching services can be appropriate for online routing systems in the large-scale road network of Hong Kong, even for the situation when a large number of users simultaneously request the routing services.

6.7 Summary

This chapter describes a reliable path searching system (RPSS) to aid users make satisfactory route choices regarding Hong Kong urban road networks with travel time uncertainties. With the use of RPSS, road users can determine the latest possible

departure time for their travel, adopt the recommended route, which has the given on-time arrival probability. The proposed RPSS was implemented using the Web Service technique. It can provide three service types to users, including reliable path searching services, location-based services and map presentation services.

In the proposed RPSS, travel time distributions were generated using both real-time and historical traffic information from the real-time travel information system in Hong Kong. Based on the generated travel time distributions, the reliable path searching services were implemented on the basis of efficient solution algorithms presented in Chapter 5. A generalized route dis-utility function was adopted for reliable path searching. The generalized route dis-utility function takes into account travellers' multiple routing criteria towards travel time reliability, travel distance and toll charge (if any) in the road network. The modified reliable shortest path algorithms considered travel time spatiotemporal correlations as well as the turn restrictions in the road network of Hong Kong. The origin and destination of the vehicular trip can start, not only at intersections, but also at the road segments.

Experimental results indicated that the optimal routes did vary substantially with travellers' various preferences toward travel time reliability, travel distance and toll charge. Cost-sensitive travellers prefer to use cheaper but unreliable routes. In order to ensure higher reliability for their on-time arrivals, some travellers assign a larger travel time budget by departing from the origin earlier. Travellers who are willing to pay a premium, tend to use reliable but expensive routes, to ensure the probability of on-time arrival. The computational efficiency of the reliable route searching services has been investigated and described in this chapter. Computational results show that the reliable path searching services can determine the optimal route in the congested road network of Hong Kong within a very short computation time. It is evident that RPSS can be applicable for the online routing systems in large-scale road networks such as the one in Hong Kong, even for situations when a large number of users simultaneously request routing services.

7. An Efficient Solution Algorithm for Solving Multi-Class Reliability-Based Traffic Assignment Problem

Efficient solution algorithms have been proposed for solving reliable shortest path problems in Chapters 3, 4 and 5. The proposed solution algorithms can help individual traveller make his/her route choice decision from one origin to one destination. A reliable path searching system has been developed to illustrate the application of these 'one-to-one' reliable shortest algorithms in routing systems and is described in Chapter 6.

In this chapter, reliable shortest path algorithms are further incorporated into the reliability-based user equilibrium (RUE) problems for long-term transportation planning under demand and/or supply uncertainties. In the literature, few solution algorithms are available for solving the RUE problems in large-scale road networks. This is mainly due to the lack of efficient solution algorithms for solving reliable shortest path sub-problems. Most existing algorithms for solving RUE problems hedge this difficulty by enumerating paths or by defining a fixed path set (Lo et al., 2006; Shao et al., 2006a; Siu and Lo, 2008). The path enumeration, however, is time consuming for a large-scale problem and as such is appropriate only for small-size networks.

An effective 'one-to-all' reliable shortest path algorithm is porposed in this chapter to simultaneously determine reliable shortest paths from single origin to all network nodes for multiple user classes in one search process. The proposed reliable shortest path algorithm avoids the repeated path searching for each user class. Based on the extended reliable shortest path algorithms, a path-based solution algorithm is proposed to solve the multi-class RUE problem by using a column generation method. The proposed RUE solution algorithm does not require path enumeration and can achieve highly accurate RUE results within reasonable computational time.

This chapter is structured as follows. A review of related literature is presented in Section 7.1. General definitions and mathematical RUE problem statements are given in Section 7.2. The solution algorithms for solving reliable shortest path sub-problem and RUE problems are described in Section 7.3. Numerical examples conducted on several large-scale networks are reported in Section 7.4. Finally, a summary of this chapter is given in Section 7.5.

7.1. Background

Developing efficient solution algorithms for traffic assignment problems is one of the most significant subjects in transportation research. Several solution algorithms have been proposed for solving user equilibrium (UE) assignment problems. As shown in Table 7.1, UE solution algorithms can be classified into three categories: link-based, origin-based and path-based algorithms.

Table 7.1 Summary of solution algorithms for solving UE problems

Algorithm class	Decision variables	Convergence rate	Memory requirement	Sources
Link-based	Link flows	Slow	Low	Sheffi (1985)
Origin-based	Origin link flows	Fast	Moderate	Dial (2006); Bar-Gera (2002)
Path-based	Path flows	Fast	High	Jayakrishnan et al. (1994); Chen et al. (2002); Panicucci et al. (2007); Florian et al. (2009)

The Frank-Wolfe (F-W) algorithm (Sheffi, 1985) is the most popular link-based method in practice, because of its ease of implementation and small memory requirement. The F-W algorithm, however, converges slowly when close to the optimal solution, and even fails to achieve high accurate UE solutions at the expense of a very large computational time (Jayakrishnan et al., 1994; Bar-Gera, 2002).

The origin-based algorithms were recently developed by Dial (2006) and Bar-Gera (2002). This class of algorithms builds upon the property that equilibrium UE solutions do not contain cyclic paths. Based on this acyclic property, UE assignments can be performed more efficiently on a set of acyclic sub-networks originating at

each origin. The origin-based algorithms can achieve high UE solution accuracy. The memory requirement of these algorithms is moderate, since they operate in the space of origin link flows which are a decomposition of link flows from each origin.

The path-based algorithms are another important group of UE solution algorithms that operate in the space of path flows (Jayakrishnan et al., 1994; Chen et al., 2002; Panicucci et al., 2007; Florian et al., 2009). The algorithms could obtain highly accurate UE solutions within reasonable computational time. However, the path-based algorithms were not widely implemented in early software packages because of their large memory requirement. With recent advances in computing power, there is an increasing interest in the development of these algorithms for solving large-scale traffic assignment problems (Panicucci et al., 2007; Florian et al., 2009).

These three types of UE solution algorithms can be easily modified for solving the UE problems with multi-class users. It should be noted that in the literature there are two applications of multi-class users in road networks. In the first application, the flows in a road network can be divided into different types of vehicles (e.g. private cars, trucks and public transit) (Lam and Huang, 1992). Each class of vehicles has its own equivalent passenger car unit with different effects on travel times. In the second application, all road users are assumed to use the same type of vehicles but they are distinguished from each other in unobservable ways such as different values of time (Yang and Huang, 2004). To solve the multi-class UE problems for these two applications, the travel flows are generally divided and equilibrated by the user class. This results in a considerable increase of computational time and memory. The multi-class RUE problem, which is the concern of this chapter, is more difficult to solve than multi-class UE, hence efficient algorithms for multi-class RUE are needed, particularly, for large-scale networks.

An investigation into the RUE problem with multi-class users (the second application) in terms of different degrees of risk-aversion is described in this chapter. Due to the fact that the non-additive property of travel time budgets (referred to Chapter 3), above, UE solution algorithms cannot be directly extended to solve the multi-class

RUE problems. Existing algorithms for solving RUE problems hedge the non-additive difficulty by enumerating paths or by defining a fixed path set (Lo et al., 2006; Shao et al., 2006a; Siu and Lo, 2008). It should be noted that the path enumeration is time consuming for a large-scale problem and as such is appropriate only for small-size networks.

7.2. Definitions and Problem Statement

To facilitate the presentation of the essential ideas in this chapter, the RUE model proposed by Shao et al. (2006a) is briefly presented. It should be noted that the proposed algorithm can also be used to solve other RUE models, such as the ones developed by Lo et al. (2006), Watling (2006) and Siu and Lo (2008). For notational consistency, capital letters represent random variables and lower-case letters represent deterministic variables throughout the chapter.

Consider a road network represented by a strongly connected graph $\mathrm{G} = (\mathrm{N}, \mathrm{A})$, where N and A are, respectively, the sets of nodes and links. RS denotes the set of O-D nodes, where node $r \in \mathrm{N}$ is the origin and node $s \in \mathrm{N}$ is the destination. Traffic demand between each O-D pair, denoted by Q_{rs}, is a random variable, subject to day-to-day demand fluctuation. The mean and the standard deviation (SD) of Q^{rs} are denoted by q^{rs} and σ_q^{rs}, respectively. Let P^{rs} be the set of paths between the O-D pair rs. The flow along a path $p_k^{rs} \in P^{rs}$ is denoted by F_k^{rs}. It is assumed that (i) the path flow follows the same class of probability distribution as O-D demand; (ii) the Coefficient of varation (CV) of path flow is equal to that of O-D demand; (iii) the path flows are mutually independent. The flow conservation equations can then, be expressed as

$$Q_{rs} = \sum_k F_k^{rs} \qquad \forall rs \in RS \tag{7.1}$$

$$q^{rs} = \sum_k f_k^{rs} \qquad \forall rs \in RS \tag{7.2}$$

$$\sigma_{f,k}^{rs} = \sqrt{Var[F_k^{rs}]} = f_k^{rs} cv_{rs} \qquad \forall k \in P^{rs},\ \forall rs \in RS \tag{7.3}$$

where f_k^{rs} and $\sigma_{f,k}^{rs}$ are the mean and the SD of flow along path $p_k^{rs} \in P^{rs}$, and cv^{rs} is CV of O-D demand. Aggregating path flows passing through a link over all O-D pairs results in link flow V_a as

$$V_a = \sum_{rs}\sum_{k} F_k^{rs}\delta_{k,a}^{rs} \qquad \forall a \in \mathrm{A} \tag{7.4}$$

$$v_a = \sum_{rs}\sum_{k} f_k^{rs}\delta_{k,a}^{rs} \qquad \forall a \in \mathrm{A} \tag{7.5}$$

$$\sigma_v^a = \sqrt{Var[V_a]} = \sqrt{\sum_{rs}\sum_{k} Var[F_k^{rs}]\delta_{k,a}^{rs}}\ , \forall a \in \mathrm{A} \tag{7.6}$$

where $\delta_{k,a}^{rs}$ is the path-link indicator; $\delta_{k,a}^{rs} = 1$ if path p_k^{rs} uses link a, and 0 otherwise; v_a and σ_v^a are the mean and the SD of V_a respectively. The link travel time is assumed to follow widely used Bureau of Public Road (BPR) link performance function (Sheffi, 1985; Shao et al., 2006a):

$$T_a = T_a(V_a) = t_a^0\left(1+\beta\left(\frac{V_a}{c_a}\right)^n\right), \forall a \in \mathrm{A} \tag{7.7}$$

where t_a^0 is the free-flow travel time on link a; c_a is the link capacity; β and n are parameters. As the link flow V_a is a random variable, the link travel time T_a is also a random variable. Let the mean and the SD of T_a be t_a and σ_t^a, respectively. The path travel time T_k^{rs} can be expressed by summing the corresponding link travel time variables:

$$T_k^{rs} = \sum_a T_a\delta_{k,a}^{rs}\ ,\ \ \forall p_k^{rs} \in P^{rs},\ \ \forall rs \in RS \tag{7.8}$$

The path travel time is assumed to follow a normal distribution. The mean and SD of path travel time, denoted as t_k^{rs} and $\sigma_{t,k}^{rs}$, can then, be expressed as

$$t_k^{rs} = \sum_a t_a\delta_{k,a}^{rs}\ ,\ \ \forall p_k^{rs} \in P^{rs},\ \ \forall rs \in RS \tag{7.9}$$

$$\sigma_{t,k}^{rs} = \sqrt{\sum_a \left(\sigma_t^a\right)^2\delta_{k,a}^{rs}}\ ,\ \ \forall p_k^{rs} \in P^{rs},\ \ \forall rs \in RS \tag{7.10}$$

For modelling travellers' route choice behaviour under travel time uncertainty, the concept of travel time budget (presented in Chapter 2), defined as the sum of

expected travel time and buffer time, is adopted. Let $\Phi_{rs,k,m}^{-1}$ denotes the travel time budget for a given user class $m \in M$:

$$\Phi_{rs,k,m}^{-1} = t_k^{rs} + z_\alpha^m \sigma_{t,k}^{rs}, \quad \forall p_k^{rs} \in P^{rs,m}, \quad \forall rs \in RS, \quad \forall m \in M \tag{7.11}$$

where z_α^m is the inverse of standard normal cumulative distribution function at α confidence level for user class m. The confidence level α is defined as the probability of arriving at a destination within $\Phi_{rs,k,m}^{-1}$. The value of α depends on users' socio-economic characteristics and trip purposes. Travellers with a larger α value display a greater aversion to risk. As users generally preferred punctual arrival, the confidence level is assumed to be greater than or equal to 50% (i.e. $\alpha \geq 0.5$).

By assuming that all travellers in the road network with uncertainty would try to choose a reliable path in order to minimize their travel time budget, the RUE route choice pattern can be stated as: for each O-D pair and user class, the travel time budget $\Phi_{rs,k,m}^{-1}$ of all the used paths are equal and minimum; and all unused paths have an equal or higher travel time budget. Denoting $p^{rs,m} \in P^{rs,m}$ as the reliable shortest path with the minimum travel time budget $\Phi_{rs,m}^{-1}$ for the O-D pair rs and user class m, the RUE condition can be expressed as:

$$f_k^{rs,m}(\Phi_{rs,k,m}^{-1} - \Phi_{rs,m}^{-1}) = 0, \quad \forall p_k^{rs} \in P^{rs,m}, \quad \forall rs \in RS, \quad \forall m \in M \tag{7.12}$$

$$\Phi_{rs,k,m}^{-1} - \Phi_{rs,m}^{-1} \geq 0, \quad \forall p_k^{rs} \in P^{rs,m}, \quad \forall rs \in RS, \quad \forall m \in M \tag{7.13}$$

Let $u^{rs,m}$ be the proportion of user class m out of the total O-D demand. The above RUE problem can be further expressed as the following gap function formulation:

$$\min \quad GAP = \sum_m \sum_{rs} \sum_k f_k^{rs,m}(\Phi_{rs,k,m}^{-1} - \Phi_{rs,m}^{-1}) \tag{7.14}$$

$$\sum_k f_k^{rs,m} = u^{rs,m} q^{rs}, \quad \forall rs \in RS, \quad \forall m \in M \tag{7.15}$$

$$\sum_m u^{rs,m} = 1, \quad \forall rs \in RS, \quad \forall m \in M \tag{7.16}$$

$$\Phi_{rs,k,m}^{-1} - \Phi_{rs,m}^{-1} \geq 0, \quad \forall p_k^{rs} \in P^{rs,m}, \quad \forall rs \in RS, \quad \forall m \in M \tag{7.17}$$

$$f_k^{rs,m} \geq 0\,,\ \ \forall p_k^{rs} \in P^{rs,m}\,,\ \ \forall rs \in RS\,,\ \ \forall m \in M \tag{7.18}$$

In this formulation, decision variables are mean path flows $f_k^{rs,m}$. The gap function in Eq. (7.14) refers to the overall gap capturing the complementary slackness conditions of the RUE model, where $|M|$ represents the number of user classes. The gap function is non-negative (i.e. $GAP \geq 0$) (Lo and Chen, 2000).

7.3. Solution Algorithm

In this section, a path finding algorithm to determine the reliable shortest paths for multi-class users is first presented in Section 7.3.1. Based on this algorithm, a traffic assignment solution algorithm is then proposed for solving multi-class RUE model in Section 7.3.2.

7.3.1. Solution algorithms for solving reliable shortest path sub-problem

The mathematical definition of the one-to-all reliable shortest path problem (RSPP) can be formally expressed as the following optimization problem:

$$\min \quad \Phi_{rs,k,m}^{-1} = \sum_{a \in A} t_a \delta_{k,a}^{rs} + z_a^m * \sqrt{\sum_{a \in A} \left(\sigma_t^a\right)^2 \delta_{k,a}^{rs}} \tag{7.19}$$

Subject to

$$\sum_{\{j:a_{ij} \in A\}} \delta_{k,a_{ij}}^{rs} - \sum_{\{j:a_{ji} \in A\}} \delta_{k,a_{ji}}^{rs} = \begin{cases} |N| - 1, & \forall\, i = r \\ -1, & \forall\, i \neq r \end{cases} \tag{7.20}$$

$$\delta_{k,a}^{rs} \in \{0, 1\}, \quad \forall a_{ij} \in A \tag{7.21}$$

Objective function Eq. (7.19) is the summation of travel time budget for all paths originated at a given origin. Eq. (7.20) ensures that paths are feasible, where $|N|$ is the number of nodes in the network. Eq. (7.21) is concerned with the link-path incidence variables which are binary in nature.

As indicated in Chapter 3, due to the non-linear property of objective function in Eq. (7.19), the RSPP cannot be solved using the traditional shortest path algorithms (e.g.

Dijkstra's algorithm). The multi-criteria shortest path approach, established in Chapter 3, can be easily extended to solve the one-to-all RSPP. For example, the label-selection multi-criteria A* algorithm (*RSPP-LA**) can be modified into a label-selection label-correction algorithm (Guerriero and Musmanno, 2001). To determine reliable shortest paths for multiple user classes with different α values, a straightforward method is expressed by repeatedly using modified reliable shortest path algorithms for each α value. However, such a method may require considerable computational time when the number of user classes becomes large.

As indicated above, an efficient solution algorithm, to find reliable shortest paths for multiple user classes in one search process, is proposed, in this chapter using following property.

Proposition 7.1 If $p_j^{ru} \in P^{ru}$ is a non-dominated path under a confidence level α, then it is also a non-dominated path for any confidence level $\hat{\alpha} \geq \alpha$.

Proof. Suppose $p_i^{ru} \in P^{ru}$ M-B dominates p_j^{ru} for a given confidence level $\hat{\alpha} \geq \alpha$. According to the M-B dominance condition (Proposition 3.3), $\Phi_{rs,i,m}^{-1}(\hat{\alpha}) - \Phi_{rs,j,m}^{-1}(\hat{\alpha}) = t_i^{ru} + z_{\hat{\alpha}}^m \sigma_{t,i}^{ru} - t_j^{ru} - z_{\hat{\alpha}}^m \sigma_{t,j}^{ru} \leq 0$. Then, $\Phi_{rs,i,m}^{-1}(\alpha) - \Phi_{rs,j,m}^{-1}(\alpha)$ $= \frac{z_\alpha^m}{z_{\hat{\alpha}}^m}\{\Phi_{rs,i,m}^{-1}(\hat{\alpha}) - \Phi_{rs,j,m}^{-1}(\hat{\alpha})\} + (\frac{z_{\hat{\alpha}}^m - z_\alpha^m}{z_{\hat{\alpha}}^m})(t_i^{ru} - t_j^{ru})$. Since $z_{\hat{\alpha}}^m \geq z_\alpha^m \geq 0$ and $t_i^{ru} - t_j^{ru} \leq 0$, we have $\Phi_{rs,i,m}^{-1}(\alpha) - \Phi_{rs,j,m}^{-1}(\alpha) \leq 0$. Therefore, p_i^{ru} M-B dominates p_j^{ru} under confidence level α, contradicting the assumption that p_j^{ru} is a non-dominated path under a confidence level α. □

Based on Proposition 7.1, an efficient solution algorithm to find reliable shortest paths for multiple user classes can be developed in one search process. Let $\hat{\alpha}$ be the maximum confidence level amongst all user classes. The algorithm firstly generates all non-dominated paths under $\hat{\alpha}$ confidence level using M-B dominance condition (Proposition 3.3). According to Proposition 7.1, reliable shortest paths for each user class can then be determined from the generated non-dominated path sets by choosing the path with minimum travel time budget of each user class. The steps

of the proposed algorithm (*RSPP-LSLC*), based on label-selection label-correcting method, are presented as follows.

Algorithm *RSPP-LSLC*

Inputs: an origin node r, $|M|$ user classes

Returns: reliable shortest paths rooted at the origin for all user classes

Step 1. Initialization.

Create a path p_i^{rr} from r to itself, and set $t_i^{rr} := 0$, $(\sigma_{t,i}^{rr})^2 := 0$ and $\Phi_{rs,i,m}^{-1} := 0$. Add p_i^{rr} into label-vector P^{rr} and the list of candidate labels SE.

Step 2. Label selection.

If $SE = \phi$, then goto Step 4.

Take label $p_i^{ru} \in P^{ru}$ at node u from SE in FIFO order.

Step 3. Path extension.

For every outgoing link a of chosen node u (v denotes a successor node of node u)

Step 3.1 Generate a new label $p_i^{rv} \in P^{rv}$. Set $t_i^{rv} := t_i^{ru} + t_a$, $(\sigma_{t,i}^{rv})^2 := (\sigma_{t,i}^{ru})^2 + \left(\sigma_t^a\right)^2$ and $c_i^{rv} := t_i^{rv} + z_{\hat{\alpha}}\sigma_{t,i}^{rv}$.

Step 3.2 If $p_i^{rv} \in P^{rv}$ is acyclic, then goto Step 3.3; otherwise scan next link

Step 3.3 If p_i^{rv} is a non-dominated path under M-B dominance condition, then insert p_i^{rv} into P^{rv} and SE, and remove all paths M-B dominated by p_i^{rv} from P^{rv} and SE.

End for.

Goto Step 2.

Step 4. Determine the reliable shortest path for each O-D pair and user class. Stop.

7.3.2. Solution algorithm for solving multi-class RUE problems

In this section, using the proposed *RSPP-LSLC* algorithm, a path-based solution

algorithm is proposed for solving multi-class RUE assignment problems. Such an algorithm is called a *RUE-PB* algorithm, which uses a sequential decomposition by an O-D pair and user class. The path set for each O-D pair and each user class are generated by the column generation method using the proposed *RSPP-LSLC* algorithm on a needs basis. This avoids the burden of enumerating a predefined set of paths in advance. For each O-D pair and user class, a restricted sub-problem is solved by iteratively shifting flows from the costliest path, with a maximum travel time budget, to the cheapest path with a minimum budget. This method for solving the restricted sub-problem was initially proposed by Dafermos and Sparrow (1969) and found to be effective and computationally efficient by Dial (2006).

Similar to the work of Dial (2006), the max-min cost difference bound ε is adopted in the *RUE-PB* algorithm as the termination criterion. A set of used paths is said to be ε-equilibrated if the travel time budget difference between the costliest and the cheapest paths is less than or equal to the given tolerance ε. The RUE solution is called ε-optimal once all path sets containing reliable shortest paths are simultaneously ε-equilibrated. This termination condition is intuitive, as it directly reflects the RUE complementary slackness conditions in Eqs. (7.12-7.13). The steps of proposed *RUE-PB* algorithm are presented below.

Algorithm *RUE-PB*

Step 1. Initialization.

For each origin r

Step 1.1 Call procedure *RSPP-LSLC* under free flow condition to generate reliable shortest path $p^{rs,m} \in P^{rs,m}$ for each O-D pair rs and user class m

Step 1.2 Add $p^{rs,m}$ into the path set $P^{rs,m}$ and assign all trips $u^{rs,m}q^{rs}$ on $p^{rs,m}$.

End for

Step 1.3 Update link flows V_a and link travel times T_a.

Step 2. Column generation.

For each origin r

Step 2.1 Call procedure $RSPP\text{-}MC$ to generate reliable shortest path $p^{rs,m} \in P^{rs,m}$ for each O-D pair rs and user class m.

Step 2.2 Remove unused paths in $P^{rs,m}$.

Step 2.3 If $\Phi_{rs,m}^{-1}$ is less than any $\Phi_{rs,k,m}^{-1}$ in $P^{rs,m}$, then add $p^{rs,m}$ into $P^{rs,m}$.

End for

Step 3. Convergence test.

If all path sets are ε-equilibrated, then Stop; otherwise goto Step 4.

Step 4. Flow update.

For $n = 1$ to number of inner iteration

For each O-D pair rs and user class m

Step 4.1 Shift path flows between the costliest and cheapest paths in $P^{rs,m}$.

Step 4.2 Update link flows V_a and link travel times T_a.

End for

If all path sets are ε-equilibrated, then goto Step 2.

End for

Goto Step 2.

In Step 1, all O-D demands for each user class are assigned to the corresponding reliable shortest path under free flow (zero link flow) condition, referred as an all-or-nothing assignment. In Step 2, the reliable shortest path for each O-D pair and each user class is generated by using the proposed *RSPP-LSLC* algorithm. The column generation is then carried out by removing the unused paths and adding the new generated reliable shortest path. In Step 3, the max-min cost difference for each O-D pair and each user class is calculated to check whether all path sets are ε-equilibrated. If this termination condition is not satisfied, a flow update is carried out in Step 4. In Step 4, the path set for each O-D pair and each user class is equilibrated by shifting path flows from the costliest path with a maximum travel time budget to the cheapest path with a minimum travel time budget. The algorithm

is continuously running during Steps 2-4 until the termination condition is satisfied.

The mechanism used in Step 4.1 for shifting path flows from the costliest path $p_\kappa^{rs} \in P^{rs,m}$ to the cheapest path $p_\lambda^{rs} \in P^{rs,m}$ is described briefly, as follows. Shifting path flows ΔF from p_κ^{rs} to p_λ^{rs} only affects link flows along these two paths. As the BPR function in Eq. (7.7) is a strictly increasing function, with respect to the link flow, travel time budget $\Phi_{rs,\kappa,m}^{-1}(F_\kappa^{rs})$ is also an increasing function in respect to its path flow. Shifting ΔF flows from p_κ^{rs} to p_λ^{rs} results in a decrease on $\Phi_{rs,\kappa,m}^{-1}(F_\kappa^{rs} - \Delta F)$ but with an increase on $\Phi_{rs,\lambda,m}^{-1}(F_\lambda^{rs} + \Delta F)$. Consequently, the objective GAP function decreases. Two scenarios to determine an appropriate ΔF, are considered (1) If all trips on path p_κ^{rs} to path p_λ^{rs} are shifted and the relationship $\Phi_{rs,\lambda,m}^{-1}(F_\lambda^{rs} + \Delta F) \le \Phi_{rs,\kappa,m}^{-1}(F_\kappa^{rs} - \Delta F)$ still holds, ΔF is then simply set as $\Delta F = F_\kappa^{rs}$. (2) Otherwise, the appropriate ΔF to equilibrate these two paths is sought by solving following minimization problem:

$$\min \quad \left(\Phi_{rs,\kappa,m}^{-1}(F_\kappa^{rs} - \Delta F) - \Phi_{rs,\lambda,m}^{-1}(F_\lambda^{rs} + \Delta F)\right)^2 \tag{7.22}$$

Subject to

$$0 < E(\Delta F) < f_\kappa^{rs} \tag{7.23}$$

This is a simple non-linear constrained optimization problem with respect to one decision variable (i.e. $E(\Delta F)$). In this chapter, the exterior penalty method, due to its simplicity and efficiency, is adopted (Nocedal and Wright, 2006).

Proposition 7.2 When the *RUE-PB* algorithm terminates, the RUE solution can be obtained.

Proof. It can be proven that GAP reaches its approximate minimum value when the algorithm terminates. According to Step 3, when the *RUE-PB* solution algorithm terminates, all path sets are ε-equilibrated. Since ε is sufficiently small, the RUE condition holds approximately, i.e. $f_k^{rs,m}(\Phi_{rs,k,m}^{-1} - \Phi_{rs,m}^{-1}) \approx 0$. Summing up the RUE condition for all paths in each O-D pair and each user class yields that $GAP \approx 0$. As $GAP \ge 0$, the minimum value of the objective function is approximately obtained at

the termination.

It can also be proved that all the constraints are satisfied when the *RUE-PB* solution algorithm terminates. Obviously, the non-negative constraints (Eqs. (7.17-7.18)) are satisfied. The path flows, at each iteration, are updated by shifting flows between the same O-D pair and user class. Thus, the flow conservation conditions (Eqs. (7.15-7.16)) are satisfied. As a result, all the constraints are satisfied when the algorithm terminates.

It is seen from the above discussion that, when the *RUE-PB* solution algorithm terminates, the RUE solution can be obtained. □

7.4. Numerical Examples

The performance of the one-to-all reliable shortest path algorithms is examined with different implementations and described in Section 7.4.1. The sensitivity of the *RUE-PB* algorithm performance is then investigated to different parameters in Section 7.4.2.

Several real networks, obtained from (http://www.bgu.ac.il/~bargera/tntp/), were used for numerical experiments. The basic characteristic of these networks are presented in Table 7.2. Unless otherwise stated, three classes of travellers with different degrees of risk-aversion were tested, according to the survey data (Lo et al., 2006). These travellers are referred to as low reliability (LR), medium reliability (MR) and high reliability (HR) users. Confidence level α for LR, MR and HR users was set to be 54%, 81% and 96%, respectively. The respective proportions of these three classes of users account for 49.5%, 38% and 12.5% of the total O-D demands.

All solution algorithms proposed in this chapter were coded in Visual C# programming language, and tested on a desktop PC with an Intel dual-core 3.2 GHz CPU and 4 GB RAM running Windows Vista operating system.

Table 7.2 Basic characteristic of test networks

Network	Zone (Origin)	Node	Link	O-D Pairs
Sioux Falls	24	24	76	528
Anaheim	38	416	914	1,406
Chicago Sketch	386	933	2,950	93,513
Chicago Region	1,771	12,982	39,018	2,297,945

7.4.1. Computational performance of the reliable shortest path algorithms

The computational performance of *RSPP-LSLC* algorithm with three different implementations of labeling strategies is tested in this section. Based on the same M-B dominance (Proposition 3.3), two other labeling strategies, label-selection label-setting (*RSPP-LSLS*) (Martins, 1984) and node-selection label-correcting (*RSPP-NSLC*) (Brumbaugh-Smith and Shier, 1989), are also implemented for comparison. The detailed steps of *RSPP-LSLS* and *RSPP-NSLC* algorithms are given in Appendix D. Based on the M-V dominance (Proposition 3.2), the solution algorithm *BSPP-NSLC* (Skriver and Andersen, 2000) for solving general BSPP problems was also implemented to enable further comparisons. The performance of these algorithms was analyzed on several networks. For each network, the mean and variance of link travel times were obtained from the results of the proposed RUE model.

The computational performance of *BSPP-NSLC* and *RSPP-NSLC* algorithms using the same node-selection label-correcting strategy is reported in Table 7.3. Theoretically, the complexity of *NSLC* method is $O(|A||N||P|)$, where $|N|$ and $|A|$ represent the respective number of network nodes and links and $|P|$ is the maximum number of non-dominated paths associated at a node. It is seen from Table 7.3 that the proposed *RSPP-NSLC* algorithm is much faster than the *BSPP-NSLC* algorithm on all the tested networks. For instance, in the Chicago Sketch network with 933 nodes, the average computational time of *RSPP-NSLC* is 268.3 (837.11/3.12) times less than that of the *BSPP-NSLC* algorithm. This result is expected, as the M-B dominance condition adopted in *RSPP-NSLC* can significantly

reduce $|P|$ when compared to the M-V dominance condition used in *BSPP-NSLC*.

For example, in the Chicago Sketch network, the number of non-dominated paths generated by *RSPP-NSLC* is only 2,124, while the number of non-dominated paths generated by *BSPP-NSLC* increase significantly to 56,092.

Table 7.3 also presents the computational performance of the *RSPP-NSLC* and the *RSPP-LSLC* algorithms. Theoretically, the computational complexity of the *RSPP-NSLC* algorithm, $O(|A||N||P|)$, is better than that of the *RSPP-LSLC* algorithm $O(|A||N||P|^2)$. However, it is seen in Table 7.3 that the *RSPP-NSLC* algorithm performs less efficient than the *RSPP-LSLC* algorithm. For example, in the Chicago Region network, the average computational time required by *RSPP-NSLC* is about 129.7% of *RSPP-LSLC*. This result may be due to the fact that the *RSPP-LSLC* algorithm generates only a few non-dominated paths. In the *RSPP-NSLC*, all labels at node i are selected for path extension, at each iteration. In this way, more labels are generated for each node j adjacent to node i, at each iteration, than that generated by the *RSPP-LSLC* in which only a single label is selected at an individual node. As such, the *RSPP-NSLC* algorithm requires additional computational effort to generate more labels in the path search process.

Table 7.3 The computational time of reliable shortest path algorithms (millisecond)

Network	*BSPP-NSLC*	*RSPP-LSLC*	*RSPP-LSLS*	*RSPP-NSLC*
Sioux Falls	0.08	0.04	0.05	0.05
Anaheim	23.50	0.75	0.91	1.23
Chicago Sketch	**837.11**	**2.44**	2.68	3.12
Chicago Region	*****	***138***	160	***179***

*Computational time>30min

It is also seen from Table 7.3 that the *RSPP-LSLC* algorithm performs slightly better than the *RSPP-LSLS* algorithm. For example, in the Chicago Region network, the average computational time required by *RSPP-LSLC* is about 86.3% of *RSPP-LSLS*. This is mainly due to the use of different of label selection methods. The *RSPP-LSLC*

algorithm utilizes some computationally inexpensive ways for label selection (e.g. a first in first out sequence). However, there is no guarantee that the paths generated at the current stage are the actual non-dominated paths and consequently, such paths have to be re-evaluated at later stages of the path search process, hence increasing the computational time required. In contrast to the above label-correcting method, the *RSPP-LSLS* algorithm selects the label with a minimum travel time budget for path extension. In this way, paths generated at the current stage are the actual non-dominated paths and are without the requirement of further evaluation. Nevertheless, such label selection method is computationally expensive. The computational complexity of *RSPP-LSLS* is $O(|A||P|^2 + |N||P|Log(|N||P|))$ when using the F-heap data structure (Fredman and Tarjan, 1987) for the label selection.

7.4.2. Sensitivity analysis of the solution algorithm for solving RUE problems

The testing of the sensitivity of the *RUE-PB* performance against network size, congestion level, demand variation and number of user classes is described in this section. The *RUE-PB* algorithm was implemented using the *RSPP-LSLC* algorithm. The RUE solution accuracy level was measured by the relative gap *RGAP* as

$$RGAP = GAP / \sum_{m} \sum_{rs} \sum_{k} f_k^{rs,m} \Phi_{rs,\kappa,m}^{-1} \tag{7.24}$$

The *RGAP* value ranges from 0 to 1. Smaller values of *RGAP* mean better approximation of the RUE solution. For simplicity, the CV is set to be 0.2 for all the O-D demands, and the parameters are set for the BRP function, as $\beta = 0.15$ and $n = 4$.

7.4.2.1. Different implementations of shortest path algorithm

The *RUE-PB* algorithm using *BSPP-NSLC* is also implemented for comparison. Fig.7.1 shows the convergence of the two algorithms for the Chicago Sketch network. The x-axis represents the computational time in seconds; whereas the y-axis is the relative gap in the logarithmic scale. The figure clearly demonstrates the advantage

of using the proposed *RSPP-LSLC* algorithm for implementing the *RUE-PB* algorithm. For example, the *RUE-PB* using *BSPP-NSLC* algorithm takes 1869.34 seconds to achieve $RGAP = 10^{-7}$ (7.04E-8); while the one using *RSPP-LSLC* algorithm takes only 17.89 seconds for the same value of relative gap and takes 324.07 seconds for $RGAP = 10^{-10}$ (9.95E-11).

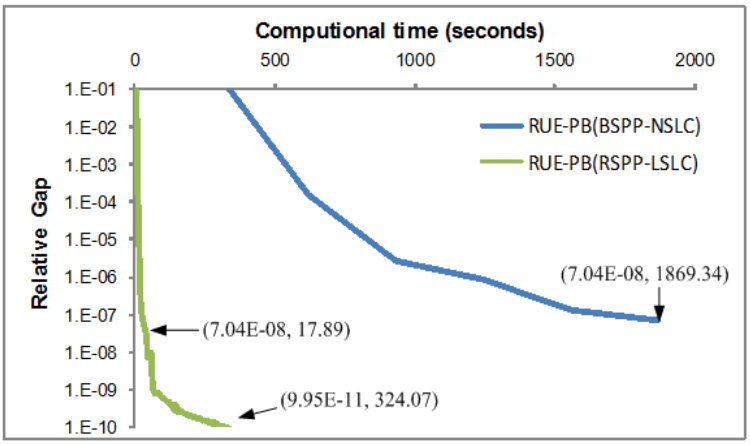


Figure 7.1 Relative gap vs. computational time for the Chicago Sketch network

7.4.2.2. Network size

Table 7.4 shows the required memory and computational time of the *RUE-PB* algorithm tested on different network sizes and accuracy levels.

Table 7.4 Performance of *RUE-PB* algorithm under different network sizes

Network	Memory requirement	Computational time			
		$RGAP=10^{-4}$	$RGAP=10^{-5}$	$RGAP=10^{-6}$	$RGAP=10^{-7}$
Sioux Falls	11.34M	0.065s	0.11s	0.17s	0.31s
Anaheim	17.04M	0.37s	0.63s	0.86s	1.08s
Chicago Sketch	**74.70M**	11.02s	12.99s	14.73s	**17.89s**
Chicago Region	***1.29G***	0.75h	0.94h	1.19h	***1.51h***

s - seconds, h - hours, M - megabytes, G – gigabytes

It can be seen from Table 7.4 that the *RUE-PB* algorithm is able to solve large-scale multi-class RUE problems with a high accuracy level. To achieve an efficient small

relative gap (say $RGAP \le 10^{-7}$), the *RUE-PB* algorithm only takes about 17.89 seconds CPU time and 74.70M RAM for Chicago Sketch network with 2,950 links, and about 1.51 hours CPU time and 1.29G RAM for Chicago Region network with 39,018 links.

7.4.2.3. Congestion level and demand variation

Fig. 7.2 gives the computational time of the RUE assignment on the Chicago Sketch network under various congestion levels and demand variations. The relative gap ($RGAP$) of 10^{-7} is adopted in this test. It is observed from Fig. 7.2 that the computational time, required by *RUE-PB* algorithm, increases as congestion levels and demand variation grow. For example, when the network is un-congested ($u = 0.5$), an increase of CV from 0.1 to 0.5 leads to a 9.5% increase in computational time. When the network is heavily congested ($u = 1.5$), the same CV increase leads to a 169% increase in computational time.

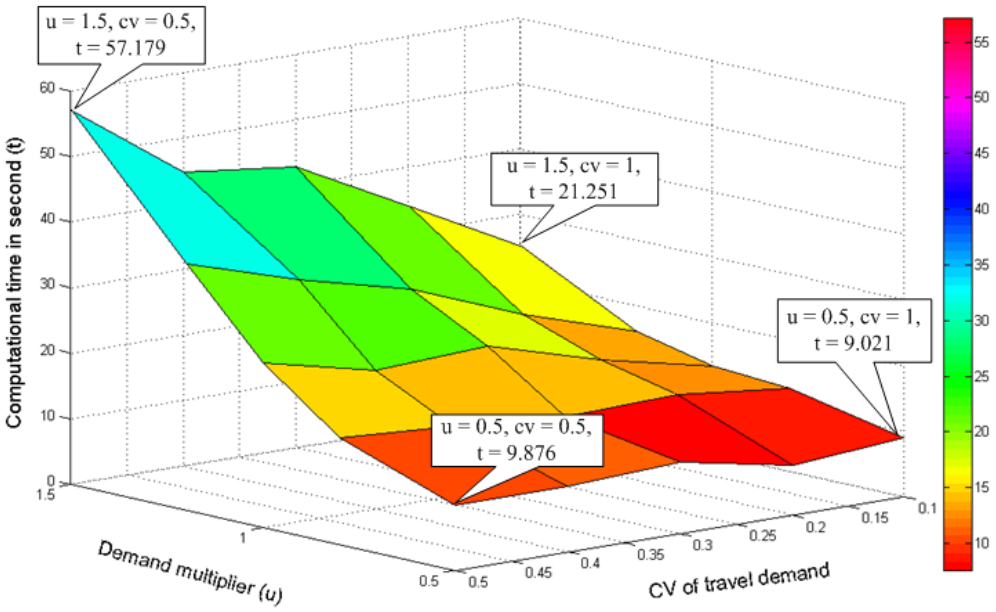


Figure 7.2 Computational time under different levels of congestions and demand variations

7.4.2.4. Number of user classes

The sensitivity of *RUE-PB* performance to different user class numbers is examined using three scenarios. The first scenario tests only the HR travellers. The second scenario is the aforementioned scenario with three classes of travellers. The last scenario tests five classes of travellers. In addition to LR, MR and HR, two other user classes are introduced. The confidence levels of these latter two user classes are 60% and 90%. In this scenario, the proportion of each user class accounts equally for 20% of the total O-D demand.

Table 7.5 shows the performance of *RUE-PB* algorithm on the large-scale Chicago Region network under the relative gap (*RGAP*) of 10^{-7}. Traditionally, memory required in the path-based algorithm is in proportion to the number of user classes. Specifically, if the number of user classes increases fivefold, the memory required also increases fivefold as the path flow of each user class needs to be stored. As shown in Table 7.5, the memory required by *RUE-PB* algorithm does not increase linearly with the number of user classes for the Chicago Region network. For instance, when the number of user classes increases fivefold (from one to five), it only has a 66% memory requirement increase instead of 500%. This property mainly contributes to the value of the *RSPP-LSLC* algorithm. As the *RSPP-LSLC* algorithm generates the reliable shortest paths for all user classes from each origin in one search process, the reliable shortest paths of all user classes from the same origin, are stored in one predecessor tree. This technique avoids the duplicated common portion storage of different paths for all user classes from each origin.

Table 7.5 Algorithm performance on Chicago network under different class numbers

Performance	Single user class	Three user classes	Five user classes
Number of iterations	11	13	17
Computational Time (Hours)	**0.75**	1.51	**2.29**
Memory requirement (Gigabytes)	***0.90***	1.29	***1.49***

From Table 7.5, it is also be seen that the computational time required by the *RUE-PB* algorithm on Chicago Region network also does not show a considerable increase, with the number of user classes, but rather grows relatively faster than the

memory requirement. An increase from one to five in the number of user classes, leads to a 205% increase in computational time. This is because the number of paths involved in the RUE problems increases linearly with the number of user classes and therefore the computational time required for solving restrict sub-problems grows significantly. In addition, the *RUE-PB* algorithm may require a greater number of iterations to equilibrate the path flows when the number of user classes grows. For example, the *RUE-PB* algorithm requires 11 iterations to solve the scenario with a single user class, and increases to 17 iterations when the five user classes are adopted. It should be noted that, at each iteration, the computational time required for finding the reliable shortest paths, does not change with the number of user classes.

7.5. Summary

In this chapter, an efficient solution algorithm for solving the reliability-based user equilibrium (RUE) traffic assignment problem with multiple user classes has been proposed. Firstly, an efficient one-to-all reliable shortest path algorithm (*RSPP-LSLC*) was proposed to determine reliable shortest paths from single origins to all network nodes for multiple user classes. This algorithm was developed upon the M-B dominance condition and its mathematical properties. It can find reliable shortest paths of multi-user classes in one search process, avoiding repeated search processes for each user class.

Secondly, the proposed *RSPP-LSLC* algorithm was further incorporated into a path-based solution algorithm (named *RUE-PB*) to solve the multi-class RUE problem. The *RUE-PB* algorithm used a column generation method for generating the path sets on a need basis. The restricted sub-problem was solved by iteratively shifting flows from the path with a maximum travel time budget, to the cheapest path with minimum travel time budget.

Finally, numerical results indicated that the proposed *RSPP-LSLC* algorithm could solve reliable shortest path problem in large-scale networks within a satisfactory computational time. The numerical results also showed that the proposed *RUE-PB*

algorithm can efficiently achieve accurate RUE results with reasonable computational time and memory requirement for large-scale networks, such as the Chicago Region network. Sensitivity tests in numerical examples demonstrate the robustness of *RUE-PB* algorithm in various network sizes, congestion levels, demand variations and number of user classes.

The RUE traffic assignment models have been regarded as valuable tools, not only for assessing impacts of different transportation planning projects and policies, but also for evaluating network performance and network vulnerability. In Chapter 8, the proposed reliable shortest path algorithms and RUE solution algorithms are applied to network vulnerability analysis to identify critical infrastructures, the failure of which would have the most serious impacts on the whole network.

8. Vulnerability Analysis for Large-scale and Congested Road Networks with Demand Uncertainty

This chapter describes a solution algorithm for identifying critical links in large-scale road networks. The critical links are defined as network links, the failure of which would have the most serious impact on the whole network. To measure of the consequences of a link closure, the RUE (reliability-based user equibirum) assignment algorithm presented in Chapter 7 is adopted. After identifying the critical infrastructures, the robustness of the transportation network to damage can be enhanced by hardening or reinforcing such critical infrastructures or constructing new alternative parallel paths.

A commonly used method, to identify critical links, is to use a full scan approach to assess all possible scenarios of link closure. This full scan approach is not viable for identifying critical links in large-scale networks, because of the large number of link closure scenarios and computational intensity of RUE assignment algorithms in these large-scale networks. In this chapter, an efficient “impact area” vulnerability analysis approach is proposed for identifying the most critical links in large-scale and congested road networks with demand variations. The proposed approach evaluates the consequences of a link closure within its local impact area instead of the whole network. This impact area vulnerability analysis is based on the empirical findings that the closure of a link would have serious impacts mainly on its adjacent links and nodes within the local impact area. As the local impact area is relatively small, computational times required for assessing the consequences of all possible link closures within the impact area, are moderate. With this approach, the computational performance required by the critical link identification can be dramatically reduced.

In addition, the effects of demand variations and travellers’ heterogeneous risk-taking behavior on the vulnerability analysis are investigated in this study. A new vulnerability index is introduced to evaluate the consequences of a link closure with

consideration of their effects. It is found in this study that both demand variations and travellers' risk-taking behavior have significant impacts on network vulnerability, and ignoring their impacts could underestimate the consequences of link closures and misidentify the most critical links.

This chapter is organized as follows. Section 8.1 briefly reviews the related studies in network vulnerability analysis literature. Section 8.2 presents the definition of the network efficiency and vulnerability index for congested road networks with demand uncertainties. The impact area vulnerability analysis approach is described in Section 8.3. Numerical examples of a medium-scale and a large-scale road network are presented in Section 8.4. Finally, a summary of this chapter is given in Section 8.5.

8.1. Background

A common technique for identifying critical links is the full network scan approach (Jenelius et al., 2006; Taylor et al., 2006). In this approach, each link is iteratively removed from the network and the consequences of its closure are measured in terms of reduced network performance. The critical links are then identified by evaluating all possible link closures. To reduce the computational burden, it is generally assumed that the link travel times are independent of traffic loads (Jenelius et al., 2006; Kurauchi et al., 2009). This assumption is quite reasonable for the network with a low travel demand, but may not be valid for the congested road networks (Berdica and Mattsson, 2007; Knoop et al., 2008). For this case, it is necessary to apply traffic assignment models to account for the congestion effects and traveller responses to the link closures (Chen et al., 2007; Taylor, 2008).

However, incorporating a full network scan approach with traffic assignments can be computationally intensive. For example, the well-known Chicago regional network for testing traffic assignment algorithms consists of 39,018 links. For each link closure, a traffic assignment conducted on the Chicago regional network requires about 1.4 hour (to achieve 10^{-3} relative gap) (Bar-Gera, 2002). Consequently, the full scan approach can take about 6.2 years to identify most critical links in such

networks. Thus, this approach may not be viable to large-scale real road networks.

In order to further reduce the computational burden associated with the full network scan approach, D'Este and Taylor (2001) and Taylor and D'Este (2004) pre-select potential vulnerable links based on certain strategies and only conduct analysis on these pre-selected links. The potential vulnerable links can either be part of a minimum cost path between O-D nodes, or the links with high choice probabilities, calculated by the stochastic traffic assignment. Based on similar ideas, Knoop et al. (2007) tested nine different link-based strategies (e.g. volume/capacity ratio) for selecting potential vulnerable links. They found that none of these strategies was good enough to properly identify the critical links on a congested road network.

8.2. Network Efficiency and Vulnerability Index

In the literature, various vulnerability indices have been proposed to evaluate the consequences of link closures. Kurauchi et al. (2009) evaluated the network vulnerability by considering the number of distinct paths connecting each origin-destination pair. Jenelius et al. (2006) used the increase of the generalized cost, weighted by the demand, as a vulnerability measure of a link closure. Taylor et al. (2006) adopted Hansen for assessing the vulnerability of the national road system of Australia, and the ARIA (Accessibility/Remoteness Index of Australia) index for the rural or remote area of Australia. Chen et al. (2007) introduced the utility-based accessibility index to take account of travellers' behavioural responses to the link closure. However, the demand variations associated with travellers' risk-taking behaviour have yet to be considered in these indices.

In this section, a new vulnerability index, based on the concept of "network efficiency", is proposed to takes into account the demand variations and associated travellers' risk-taking behavior. The concept of network efficiency was firstly introduced by Latora and Marchiori (2001) to measure the performance of communication networks as

$$E_{L-M}(G) = \frac{1}{|N|(|N|-1)} \sum_{r \neq s} \frac{1}{d_{rs}}, \qquad \forall rs \in N \tag{8.1}$$

where $|N|$ is the number of nodes in the network G, and d_{rs} is the shortest network distance (or the geodesic distance) between network nodes r and s. The larger the value of $E_{L-M}(G)$, the more efficient the information distribution in the network G.

Although this efficiency measure has been used to evaluate the performance of complex networks (e.g. communication networks, neural networks and social networks), it may not be the best index for assessing the performance of congested road networks. When it is applied to a congested road network, the definition of network efficiency given by Eq. (8.1) does not take into account the effects of travel demand, travel time and their variations. In this paper, the concept of network efficiency is therefore extended for the congested road network, as below:

$$E(G) = \frac{\sum_i \sum_{rs} \frac{u_i^{rs} q_{rs}}{\pi_i^{rs}}}{\sum_{rs} q_{rs}}, \quad \forall rs \in RS, \quad \forall i \in I \tag{8.2}$$

where recall that q_{rs} is mean travel demand between O-D nodes; u_i^{rs} is the proportion of type i travellers traveling from *r to s*; and π_i^{rs} is minimum travel time budget between the O-D pair for type i travellers. When $\pi_i^{rs} = d_{rs}$ and travel demand q_{rs} for all O-D pairs are equal, the proposed network efficiency $E(G)$ reduces to $E_{L-M}(G)$.

Let $E_0(G)$ be the network efficiency under the normal condition and $E_a(G)$ be the network efficiency after the closure of link a. The consequences of the link a closure on the network G can be evaluated in terms of the relative change in network efficiency, denoted by VUL_a^g, as below:

$$VUL_a^g = \frac{E_0(G) - E_a(G)}{E_0(G)} \tag{8.3}$$

Based on this vulnerability index VUL_a^g, the critical links of a congested road

network can be defined as those links, the closure of which causes a significant change of network efficiency.

8.3. Impact Area Vulnerability Analysis Approach

An intuitive way of identifying the whole network's most critical links is to use the traditional full network scan approach. In this approach, a RUE assignment (presented in Chapter 7) is firstly conducted to estimate equilibrium flow patterns and calculate the network efficiency under normal condition. Each link in turn is, then, removed from the network and a RUE traffic assignment is carried out again to account for congestion effects and travellers' responses to the link closure. The consequences of the link closure are measured in terms of reduced network efficiency using Eq. (8.3). Let $rank_a^g$ be the vulnerable ranking of link a closure. The $rank_a^g$ values can be calculated by sorting the VUL_a^g values of all possible link closures in a descendent order. This brute-force approach, however, can be computationally intensive for large-scale networks. In this section, an impact area vulnerability analysis approach is proposed to efficiently determine the critical links for large-scale road networks.

8.3.1 Impact area vulnerability index

The proposed impact area vulnerability analysis approach evaluates the consequences of a link closure on its local impact area instead of the whole network. It is postulated that the closure of a link would have significant impacts mainly on the adjacent links and nodes within its impact area. For example, under a certain disruptive incident, the capacity of link a may be significantly degraded, making it inaccessible to travellers. If the impact area of link a could provide alternatives for the travellers and the rerouting of these travellers did not cause serious congestion in the impact area, the effects of the link a closure will then impose restrictions only on its impact area and not disperse throughout the whole network. This postulation has been confirmed by many empirical findings after major incidents (Danczyk and Liu, 2009). Therefore, it seems reasonable to evaluate the consequence of a link

closure only in its local impact area. As the local impact area is much smaller than the whole network, the computation performance required by the traffic assignment for assessing the consequence of link closure can be significantly improved.

The impact area of the link a, denoted by $G_a = (N_a, A_a)$, is defined as a sub-network of the whole network, where N_a and A_a are the set of nodes and the set of links adjacent to the link a respectively. Let k be the size parameter of impact area measuring by a maximum un-weighted distance (or maximum number of nodes). In this way, in the impact area G_a, the un-weighted distance of any link (or node) to the link a should be less than or equal to k. If $k = 1$, only the direct neighboring the links and nodes are included in G_a.

The consequence of the closure of link a on the impact area is quantified by the "impact area vulnerability index", denoted by VUL_a^l. It can be calculated by

$$VUL_a^l = \frac{E_0(G_a) - E_a(G_a)}{E_0(G_a)} \tag{8.4}$$

where $E_0(G_a)$ and $E_a(G_a)$ respectively are the network efficiency of impact area G_a under the normal condition and after the closure of link a. Based on such impact area vulnerability index, the vulnerable ranking of link a, denoted by $rank_a^l$, can be determined in its local impact area instead of the whole network. For clarity, the vulnerability index defined on the whole network (in Eq. (8.1)) is hereafter referred to as the "global vulnerability index"; and the $rank_a^g$ value is referred as "global vulnerability ranking" (or true vulnerability ranking).

Based on the above postulation, it is expected that the impact area vulnerability ranking should have a very strong positive correlation with the global vulnerability ranking. To quantify such correlation, Spearman Rank Correlation (Spearman, 1904) is adopted. The Spearman's rank correlation coefficient ρ is calculated as

$$\rho = 1 - \frac{6\sum (rank_a^l - rank_a^g)^2}{|A|(|A|^2 - 1)} \tag{8.5}$$

where $|A|$ is the total number of links in the network. The value of ρ is between -1 and +1; and $\rho = 1$ if a perfect positive correlation exists between the ranks of the global and impact area vulnerability indexes (i.e. these two rankings are identical).

Let Q_a be the travel demand on the impact area G_a. The following steps are proposed to calculate the impact area vulnerability ranking for each link closure.

Procedure *RankByImpactArea*

Input: Impact area size parameter k

Output: Impact area vulnerability ranking for each link closure

Step 1: Conduct a RUE traffic assignment on the network G based on Q

Step 2: For each link $a \in G$

Generate impact area G_a by procedure *GenSubnetwork*

Generate travel demands Q_a for the impact area G_a

Conduct a RUE assignment on G_a to calculate $E_0(G_a)$

Remove the link a from G_a

Conduct a RUE assignment on G_a to calculate $E_a(G_a)$

Calculate impact area vulnerability index VUL_a^l using Eq. (22)

End for

Step 3: Determine the $rank_a^l$ values by sorting the VUL_a^l values in a descendent order

Procedure *GenSubnetwork*

Input: Impact area size parameter k, link a (connecting tail node j and head node h)

Output: the impact area of the link a

Step 1: Initialization:

Set the distance from the node l to the link a as $w_l := \infty, \forall l \in N$.

Set $w_j := 0$ and $w_h := 0$.

Set scan eligible node set as $SE := \{j, h\}$.

Set node set of impact area as $N_a := \{\phi\}$.

Step 2: Node selection:

Select the node l with minimum w_l from SE.

Set $SE := SE \setminus \{l\}$ and $N_a := N_a \cup \{l\}$.

Step 3: Path extension:

For each link b merging into node l (Denote m as the tail node of link b)

If $w_l + 1 \le k$ and $w_m < w_l + 1$

Set $w_m := w_l + 1$ and $SE := SE \cup \{m\}$.

End if

End for

If $\boldsymbol{SE} = \phi$, then goto Step 4; Otherwise, goto Step 2.

Step 4: Generate link set for the sub-network:

Set link set of impact area as $A_a := \{\phi\}$.

For each node $l \in N_a$

For each link b emanating from node l

If $\eta \in N_a$, then $A_a := A_a \cup \{b\}$. (Denote η as the head node of link b)

End for

End for

The proposed approach can be described briefly as follow. In the first step, a traffic assignment is conducted on the whole network G to estimate the equilibrium traffic pattern under the normal condition without any link closure. In the second step, the consequences of each link closure on the impact area are then evaluated once at a time. For each link a, the impact area G_a is generated by using the Procedure *GenSubnetwork*, which is modified from Dijkstra's algorithm. Based on the path flow pattern calculated in the first step under the normal condition, travel demand Q_a on the impact area G_a can be generated by extracting the path flows within G_a. The impact area vulnerability index VUL_a^l can then be calculated by

conducting the traffic assignments on G_a before and after the closure of link a. In the last step, the $rank_a^l$ values for all network links are determined by sorting the calculated VUL_a^l values in a descendent order.

8.3.2 Partial network scan approach

Due to limited resources, transportation planners or managers in practice want to identify only a small amount of the most critical links for road maintenance or improvement programs. The calculation of global vulnerability rankings for all network links thus is not necessary. As previously postulated, the impact area vulnerability ranking would have a very strong positive correlation with the global vulnerability ranking. In this case, the critical links with a high impact area vulnerability ranking are likely to be the actual critical links based on the global vulnerability rankings. Therefore, the critical links identified by impact area vulnerability rankings can be treated as potential vulnerable links (or candidate links) to identify the most critical links on the whole network. As such, the computational intensive tasks of calculating global vulnerability index are restricted to selected candidate links. The computational burden associated with the global vulnerability index calculation for other network links can be saved.

Let λ be the number of selected candidate links, and x be the number of most critical links that we want to identify. The detailed steps of the proposed impact area vulnerability approach are described as follows. In the first step, the impact area vulnerability ranking for each network link (i.e. $rank_a^l$ value) is calculated on its local impact area using the *RankByImpactArea* Procedure. The λ most critical links based on impact area vulnerability rankings (i.e. $rank_a^l = 1,\ldots, \lambda$) are selected as candidate links for further evaluation on the whole network. In the second step, each selected candidate link is iteratively removed from the whole network and a RUE assignment is conducted to calculate its global vulnerability index on the whole network. In the third step, the ranking of the link a amongst selected candidate links, denoted by $rank_a^p$, is obtained by sorting the calculated global vulnerability

index in a descendent order. The x most critical links are then identified as those links with the highest $rank_a^p$ values (i.e. $rank_a^p = 1,\ldots, x$).

Procedure *PartialNetworkScan*

Input: Impact area size parameter k, number of candidate links λ

Output: x most critical links

Step 1: Calculate $rank_a^l$ for each network link by using *RankByImpactArea* Procedure.

Select λ critical links with highest $rank_a^l$ values as candidate links.

Step 2: For each selected candidate link a

Remove the link a from the whole network G.

Conduct a RUE assignment on the whole network G to calculate $E_a(G)$.

Calculate the global vulnerability index VUL_a^g using Eq. (21).

Add the link a to G.

End for

Step 3: Determine $rank_a^p$ values by sorting calculated VUL_a^g values in a descendent order.

Select x most critical links with the highest $rank_a^p$ values.

The proposed impact area vulnerability analysis approach can obtain the same x most critical links (i.e. $rank_a^g = 1,\ldots, x$) as traditional full network scan approach, if these actual x most critical links have been selected in the λ candidate links. Let y be the number of actual x most critical links that are not included in the selected candidate links. In this case, y critical links are missed, while the relative rankings of identified critical links are not changed. The percentage of actual critical links identified by the proposed approach can thus be calculated as $(x-y)/x$.

8.4. Case Studies on Real-world Road Networks

This section presents two case studies involving two real networks to illustrate the application of the proposed impact area vulnerability analysis approach. The first case study considers the vulnerability of the well-known medium-size Sioux Falls network. This case study examines different aspects of vulnerability analysis including congestion effects, demand uncertainty, and travellers' risk-taking behaviour. The second case study is concerned with the application to the Hong Kong road network, of the proposed vulnerability analysis. The computational performance and accuracy of the proposed impact area vulnerability analysis approach are presented and discussed.

8.4.1 Case study of the network of Sioux Falls

The Sioux Falls network, as shown in Fig. 8.1, consists of 24 nodes, 76 links and 528 O-D pairs. The O-D demands and link parameters are downloaded from website http://www.bgu.ac.il/~bargera/tntp/. The parameters for the BRP function are $\beta = 0.15$ and $n = 4$. For simplicity, the CV vlaue is set at 0.2 for all the O-D demands in this network and three types of travellers with different degrees of risk aversion, are assumed. According to the survey results (Lo et al., 2006), the degrees of risk aversion α for Type 1, 2 and 3 travellers are found to be 54%, 81% and 96% respectively. The proportions of these three classes of travellers are set to be 49.5%, 38% and 12.5% of the total O-D demand, respectively.

8.4.1.1 Full network scan of the network of Sioux Falls

The full network scan approach is firstly used to determine the critical links on the whole network. Fig. 8.1 shows the results of the ten most critical links of the Sioux Falls network. It is seen from Fig. 8.1 that the closure of these critical links decrease the network efficiency by 4% or more. The worst case is the closure of link 43 connecting node 15 to node 10, which leads to a 5.6% decrease in network efficiency.

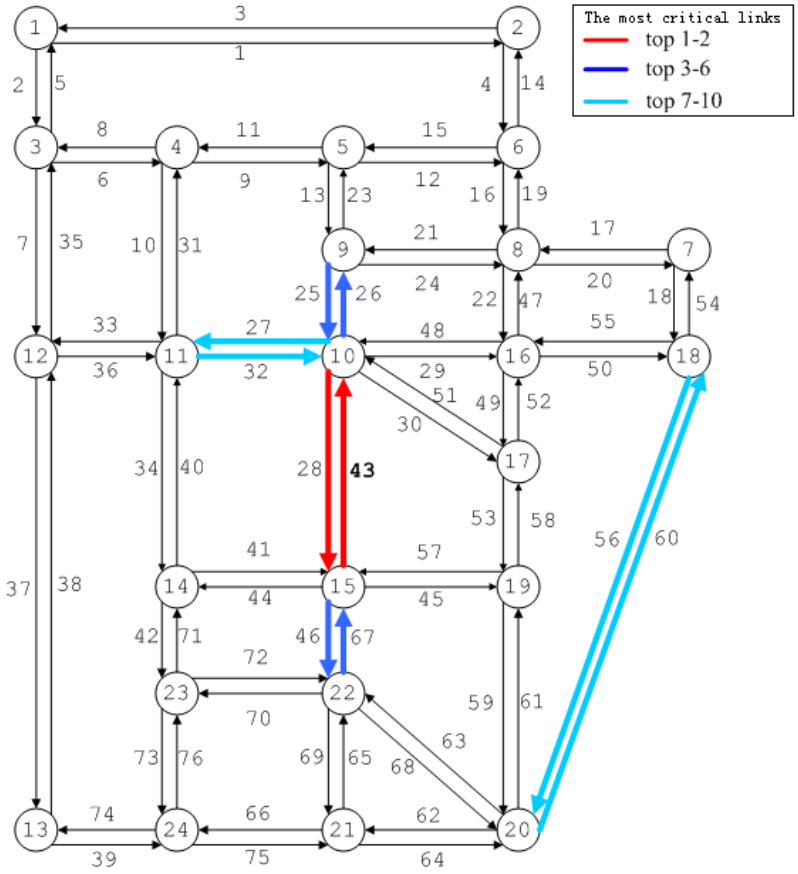


Figure 8.1 The Sioux Falls network

The consequences of the closure at link 43 are depicted in Fig. 8.2 under various demand multipliers (u) and CV values to illustrate the effects of congestion and demand uncertainty on network vulnerability. From this figure, it can be observed that the network tends to be more vulnerable with the increase of congestion level. For instance, the closure at link 43 leads to a 1.0% reduction in efficiency for un-congested conditions (u=0.1, $cv_{rs} = 0.5$), a 3.8% reduction in efficiency for congested conditions (u=0.6, $cv_{rs} = 0.5$), and a 8.5% reduction in efficiency for heavy congested conditions (u=1, $cv_{rs} = 0.5$). This result is expected because if the network is more congested, owing to closure, less spare capacity is available to absorb the rerouted traffic.

In Fig. 8.2, it is also seen that the consequences of link closure strongly depend on the level of demand uncertainty. When the CV of travel demand is large, the network becomes unreliable from the travellers' perceptive. In this case, travellers may be

more sensitive to the link closure and tend to budget a large portion of travel time as a safety margin to ensure the probability of on-time arrival.

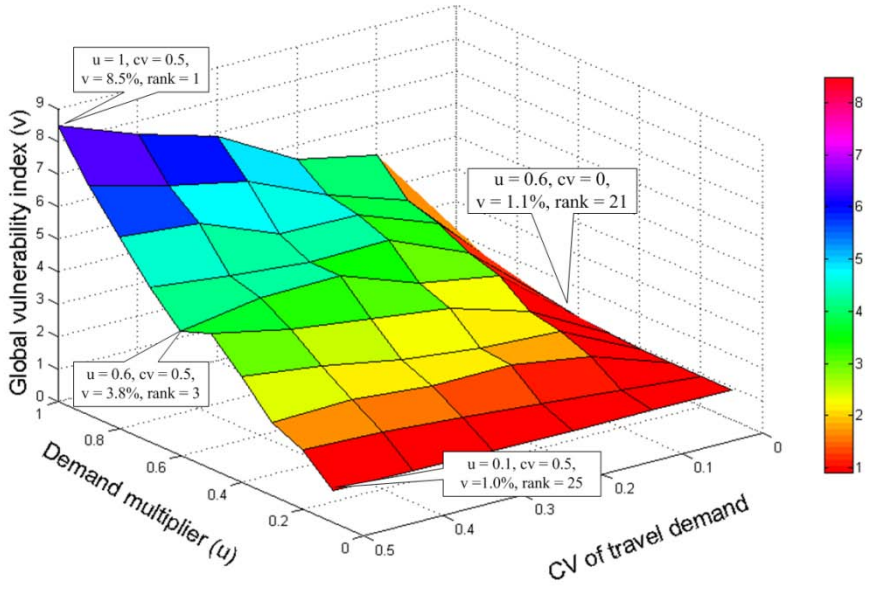


Figure 8.2 Results of vulnerability analysis under various demand multipliers and CV values

Fig. 8.2 also shows that the global vulnerability rankings were varied owing to the congestion level and demand uncertainty. For example, link 43 was identified as the most critical link when $u = 1$ and $cv_{rs} = 0.5$. However this link was ranked at $rank_a^g = 3^{\text{rd}}$ when $u = 0.6$ and $cv_{rs} = 0.5$, and $rank_a^g = 21^{\text{th}}$ when $u = 0.6$ and $cv_{rs} = 0$. For the scenario with deterministic demand scenario ($cv_{rs} = 0$), the RUE model results are those of the traditional UE model. The results of the vulnerability analysis under this scenario can be characterized by travellers' risk-taking behaviour being ignored. The above analysis also indicates that ignoring travellers' risk-taking behaviour could underestimate the consequences of link closures and misidentify the most critical links. Therefore, it is necessary to capture the stochastic aspects of the O-D demands and consider the travellers' risk-taking behaviour for the vulnerability analysis on congested road networks.

8.4.1.2 Impact area vulnerability analysis of the Sioux Falls network

The proposed impact area vulnerability analysis approach is applied to the Sioux Falls network to study the effects of different parameters on the impact area vulnerability analysis results. In the first test, O-D demands are deterministic and the impact area parameter k is set as 3. In other words, the impact area vulnerability analysis is investigated by using the traditional UE assignment.

Fig. 8.3 shows the correlation between the global and impact area vulnerability indices. The *x-axis* represents the ranking of the impact area vulnerability index; whereas the *y-axis* is the corresponding ranking of the global vulnerability index. This figure clearly shows that a strong positive correlation between the rankings of the global and impact area vulnerability indices, exists (Spearman's rank correlation coefficient $\rho = 0.97$). Thus, for this network, the impact area vulnerability analysis can be used for determining potential vulnerable links so as to reduce the search space for efficiently determining the most critical links. In this example, to determine the same ten most critical links as full network scan approach, 12 candidate links ($\lambda = 12$) are required by impact area vulnerability analysis approach.

The effects of demand uncertainty on the impact area vulnerability analysis were examined. Table 8.1 gives the ρ and λ values under different of demand variation levels. It was found that there is a strong positive correlation between the global and impact area vulnerability indices for all levels of demand uncertainty. All Spearman's rank correlation coefficient (i.e. ρ values) are 0.9 or above. It should also be noted that the stochastic demand seems to have a negative effect on the impact area vulnerability analysis. When the cv_{rs} value increases from 0 to 0.5, the ρ value decreases from 0.97 to 0.90, a decrease of about 9%. A similar negative effect can be observed at the λ values. To identify the same ten most critical links, the number of candidate links (i.e. λ value) increases from 12 to 18 when the cv_{rs} value increases from 0 to 0.5.

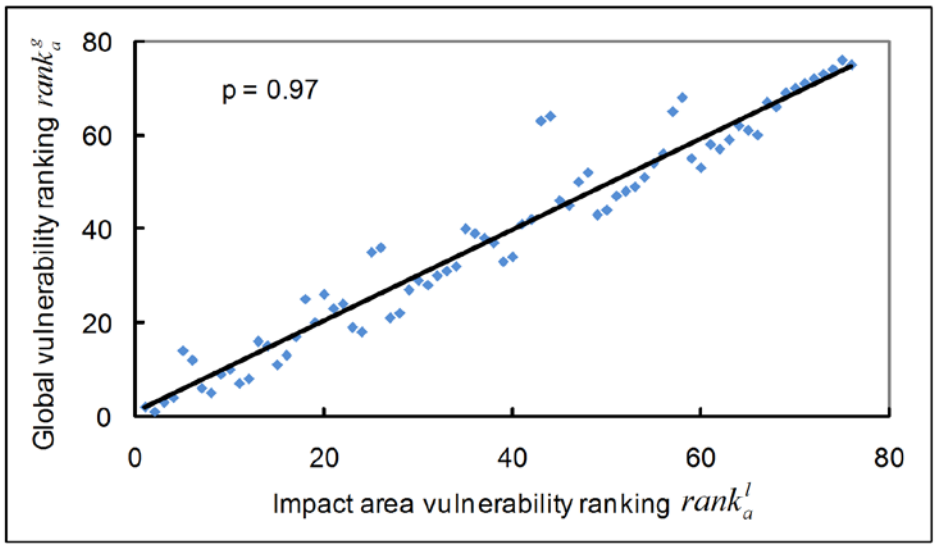


Figure 8.3 Correlation between global and impact area vulnerability

Table 8.1 Impact area vulnerability analysis for various demand uncertainty

CV of travel demand	Correlation coefficient ρ	λ values
0	0.97	12
0.1	0.93	11
0.2	0.92	14
0.3	0.92	14
0.4	0.93	18
0.5	0.90	18

8.4.2 Case study in road network of Hong Kong

Hong Kong is one of the most densely populated cities in the world. According to the Transport Department of Hong Kong Government, at end June 2009, there were 575,000 licensed cars in Hong Kong, 67% of which were privately owned. Hong Kong has a highly developed road network system, with about 1,900 kilometers of paved highways and 12 vehicular tunnels. Three of the 12 tunnels, are cross-harbor tunnels connecting the Kowloon urban area to the Hong Kong Island (Fig. 8.4). Central located on Hong Kong Island, is the Central Business District (CBD). As indicated in Chapter 6, the high travel demand attracted by the CBD leads to heavy cross-harbor traffic flows in Hong Kong particularly during the peak periods.

The road network and travel demand data used in the Third Comprehensive

Transport Study (CTS-3) was adopted for this case study. The CTS-3 was commissioned by the Transport Department of Hong Kong in late 1997 with the objective to facilitate the mobility of Hong Kong people and goods by road, rail and ferry in a sustainable manner, up t the year, 2016. As shown in Fig. 8.4, the CTS-3 road network consists of 294 zones, 1,367 nodes and 3,655 links (including 790 connectors). The morning peak period (8:00-9:00) was selected for this case study. During this period, there were 39,586 O-D pairs with the total of 237,211 trips in passenger car unit. As the O-D demands defined in CTS-3 are deterministic, in this case study the CV of O-D demands is assumed to be 0.1 for all O-D pairs in order to facilitate the presentation of essential ideas. Similar to the previous section, three types of travellers with different degrees of risk aversion according to the survey data in Hong Kong were tested (Lo et al., 2006).

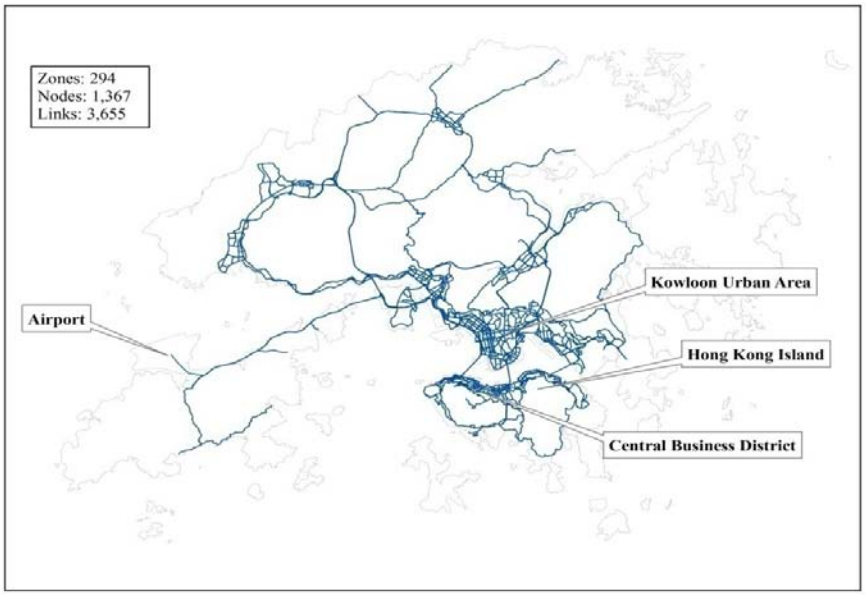


Figure 8.4 The road network of Hong Kong

8.4.2.1 Full network scan of the road network of Hong Kong

The full network scan approach is used, firstly for identifying the most critical links. To reduce the computational burden associated with the network scan approach, one possible way is to adopt a warm start technique which makes use of the assignment results, obtained under normal condition, as an initial flow pattern for subsequent RUE assignments. Some authors suggested that the warm start technique could

significantly speedup the assignment performance (Dial, 2006). In addition, it is not necessary to test the closure of centroid connectors, since they may not be the existing links in the actual road network. Under this setting, the computational time required for the full network scan is about 16.5 hours on the computer with a duo-core 1.6GHz CPU (single processor is used) for evaluating the vulnerability of the CTS-3 road network.

Fig. 8.5 shows the critical links of the Hong Kong network. The most critical links are the major roads around the Hong Kong Island, as well as the three cross harbour tunnels. The high travel demands at the CBD of Hong Kong Island may be one of the most important factors contributing to the vulnerability of these links. The other reason may be the lack of alternative roads to these links. Most roads in the Kowloon urban area, however, are found to be robust to link closure.

Despite the high traffic volume on these roads, the high network density (measured by the number of links per unit area) in the Kowloon urban area can mitigate the consequence of link closure. Thus, for the congested urban area, not those roads in high density areas but the roads with few alternatives, tend to be vulnerable to disruptive incidents. In contrast to urban area, the factor of demand level becomes much more important in the remote area where network density is low. For instance, as shown in Fig. 8.5 there is only a single path connecting the “Airport” and “Mui Po” to the Kowloon urban area. However, due to different levels of travel demand to and from these two areas, the expressway to the Airport is much more vulnerable than the road to Mui Po.

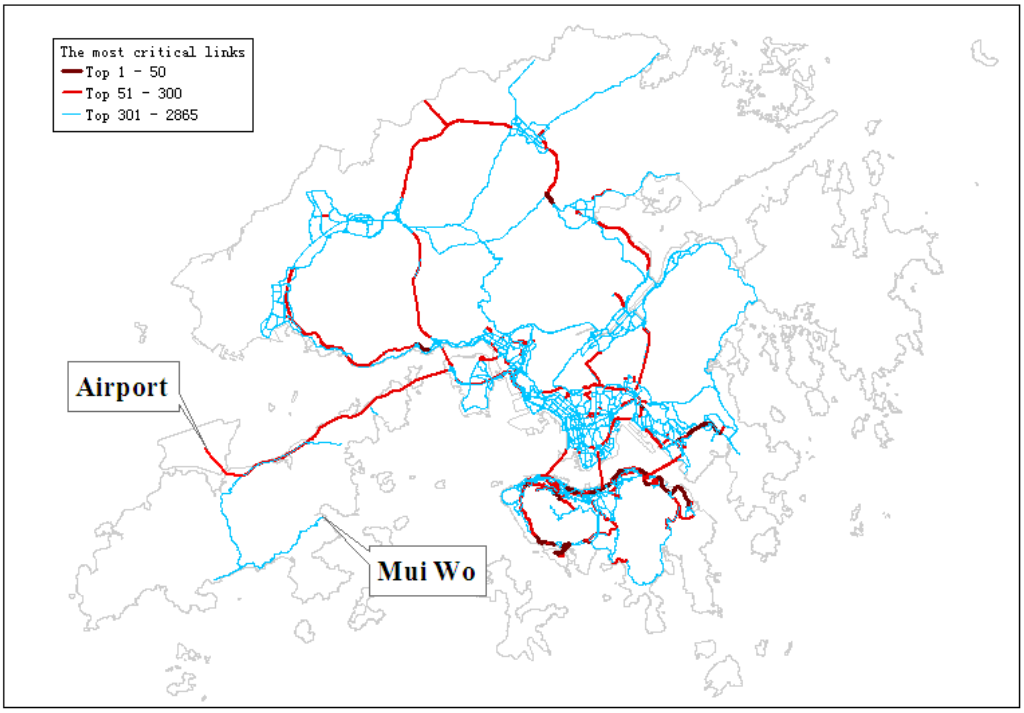


Figure 8.5 The most critical links of the Hong Kong road network when CV=0.1

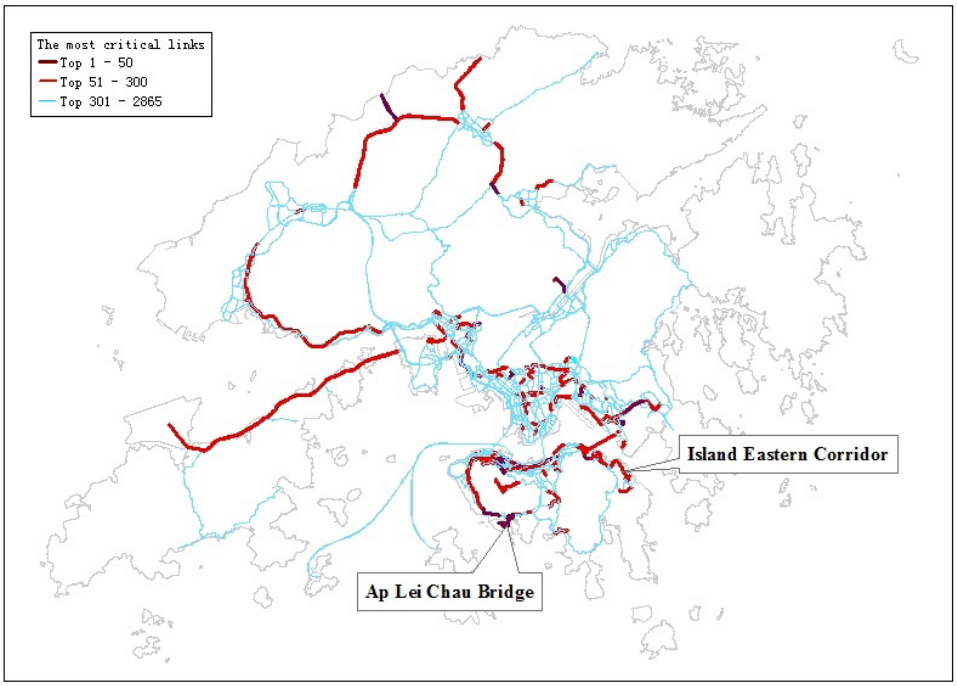


Figure 8.6 The most critical links of the Hong Kong road network when CV=0.5

The effects of demand variations on critical link identification were also investigated. Fig. 8.6 illustrates the critical links of the Hong Kong network when $cv_{rs} = 0.5$. It

can be seen from Fig. 8.6 that the major roads around Hong Kong Island are still identified as the most critical links but two of three cross harbor tunnels are ranked after 300th. In addition, the global vulnerability rankings of identified critical links are significantly changed. For example, with the increase of cv_{rs} value from 0.1 to 0.5, the global vulnerability ranking of "Ap Lei Chau Bridge" raises from $rank_a^g =$ 30th to $rank_a^g = 1^{st}$; and the ranking of the "Island Eastern Corridor" decreases from $rank_a^g = 48^{th}$ to $rank_a^g = 252^{th}$. Therefore, as previously discussed, travel demand variations and associated travellers' risk-taking behavior have a significant impact on network vulnerability analysis and ignoring their effects may misidentify the most critical links.

8.4.2.2 Impact area vulnerability analysis of the road network of Hong Kong

In this section, the impact area vulnerability analysis approach was also applied to the Hong Kong network. Table 8.2 shows the corresponding impact areas of the impact area network under different values of k. As expected, the impact area size dramatically increases with the value of k. If only the direct neighbors of each link are considered ($k = 1$), the average number of links within the impact area is 13. The average number of links in the impact area networks grows to 161 when $k = 5$, and reaches its maximum value of 3,655 when $k = 49$.

Table 8.2 Impact area vulnerability analysis for various impact area parameters

Impact area Parameter k	Average number of nodes	Average number of links	Link coverage percentage	Correlation coefficient ρ
1	7	13	0.3%	0.75
2	15	33	0.9%	0.82
3	28	65	1.7%	0.84
5	65	161	4.4%	0.86
7	116	297	8.1%	0.86
9	183	473	12.9%	0.87
49	1367	3655	100%	1

Also summarized in Table 8.2 is the correlation between the global and impact area vulnerability indices. In Table 8.2, the Spearman's rank correlation coefficient ρ

increases with the value of k. This implies that the wider the impact area of the impact area network, the more accurate the result of the impact area vulnerability analysis.

Fig. 8.7 illustrates the correlation between the global and impact area vulnerability indices when $k = 5$. In Fig. 8.7, the *x-axis* represents the rank of the impact area vulnerability index; whereas the *y-axis* is the corresponding rank of the global vulnerability index. In Fig. 8.7, a strong positive correlation is shown between the rankings of the global and impact area vulnerability index ($\rho = 0.86$). This result indicates that the critical links identified by the impact area vulnerability index can be confidently considered as the likely potential vulnerable links candidates of the whole network.

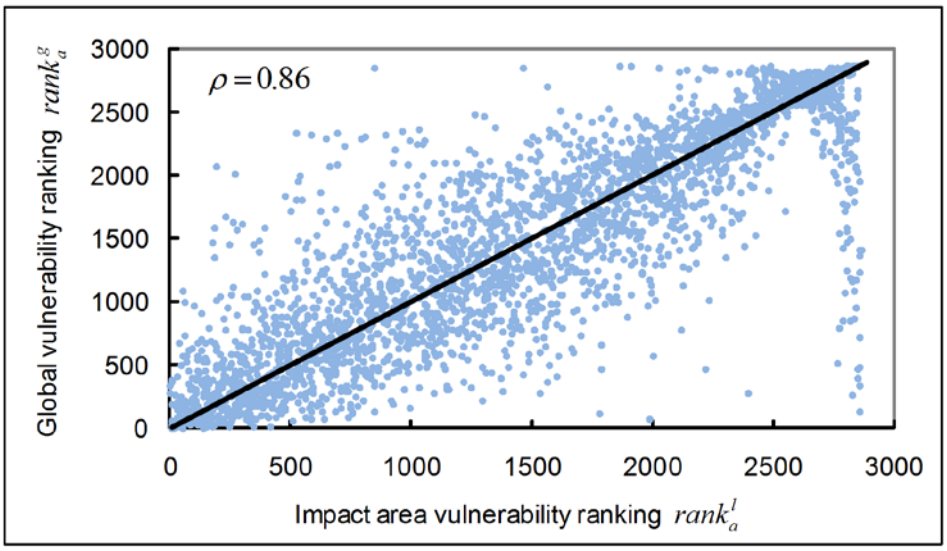


Figure 8.7 Correlation between global and impact area vulnerability rankings when CV=0.1

The 50 most critical links in the Hong Kong road network was determined by the impact area vulnerability analysis approach. In this approach, λ candidate links (i.e. $rank_a^l = 1^{st}, \ldots, \lambda^{th}$) were selected for the partial network scan. The 50 most critical links (i.e. $rank_a^p = 1^{st}, \ldots, 50^{th}$) were identified and then compared with the actual 50 most critical links based on traditional full network scan approach (i.e. $rank_a^g =$

1^{st}, …, 50^{th}).

The accuracy of the impact area vulnerability analysis approach under different k and λ values were investigated and shown in Table 8.3. It can be seen from Table 8.3 that the percentage of the identified actual 50 most critical links is monotonic increasing with the λ value. For instance, only 60% of the 50 most critical links (i.e. 30 links) were identified if $\lambda = 100$ and $\eta = 5$. Using the same k value but $\lambda = 300$, this percentage increases to 94% and only 3 actual most critical links were missed (i.e. $rank_a^g$ = 13^{th}, 18^{th}, and 35^{th}). When $\lambda = 500$, the impact area vulnerability analysis approach can identify the same 50 most critical links as the traditional full network scan approach.

It can be seen from Table 8.3 that the accuracy of the proposed impact area vulnerability analysis approach increases with the impact area size parameter k. This is reasonable, since the ρ value increases with k (see Table 8.2) and thus the selected candidate links are more likely to be the actual critical links on the whole network. It should be noted that if the k and λ values are too small (e.g. $k = 1$ or $\lambda = 100$), the proposed approach may misidentify a considerable set of most critical links.

Table 8.3 The accuracy of partial network scan approach under different parameters

Number of candidate links λ	Impact area parameter k	Percentage of identified the 50 most critical links
100	5	60%
200	5	84%
300	5	94%
400	5	98%
500	5	100%
500	1	60%
500	2	82%
500	3	94%
500	4	100%
500	5	100%

Also investigated were the effects of travel demand variations on the impact area

vulnerability analysis. Fig. 8.8 shows the correlation between global and impact area vulnerability rankings when $cv_{rs} = 0.5$. It can be observed from Fig. 8.7 and 8.8 that with the increase of cv_{rs} value from 0.1 to 0.5, the value of such correlation decreased from 0.86 to 0.81. This confirmed the previous finding that the travel demand variations tend to have a negative impact on the impact area vulnerability analysis. In this scenario ($cv_{rs} = 0.5$), the accuracy of impact area vulnerability analysis approach was examined. It was found that when $\lambda = 500$ and $\eta = 5$, the impact area vulnerability analysis approach can still identify the same 50 most critical links as the traditional full network scan approach.

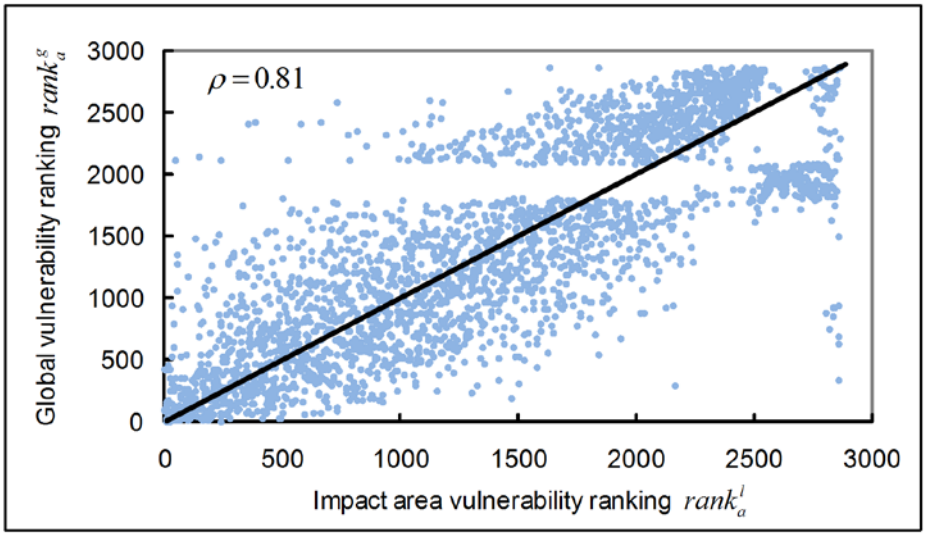


Figure 8.8 Correlation between global and impact area vulnerability rankings when CV=0.5

The computational performance of the partial network scan is affected by two aspects: the computational time required for calculating the impact area vulnerability index (denoted as t_1) and the computational time for further evaluation of the potential vulnerable links identified by the impact area analysis (denoted as t_2). Note that t_1 only depends on the value of k; and t_2 is mainly affected by λ. Table 8.4 summarizes the computational performance of the partial network scan approach. To improve the computational performance, the warm start technique is also adopted. It can be observed from Table 8.4 that t_1 and t_2 increase with k and λ respectively. In Table 8.4, t_1 does not increase linearly with k. This result

is intuitive because the computational effort of the traffic assignment (involved in impact area vulnerability analysis) increases exponentially with the size of impact area (Chen et al., 2002). Thus, it is understandable however, that t_2 would increase linearly with λ, since all the traffic assignments involved in this step are conducted in the same whole network.

As shown in Table 8.4, if $k = 5$ and $\lambda = 500$ is chosen for the identification of the same 50 most critical links, the total computational time required by the partial network scan approach is 140 (11.0+129) minutes. This is about 14% of computational time required by the traditional full network scan. In other words, the partial network scan approach is about 7.1 times faster in identifying the same 50 most critical links, than the full network scan. To achieve 94% accuracy of identifying the 50 most critical links, the partial network scan approach (requires 88 minutes, using $k = 5$ and $\lambda = 300$) is about 11.3 times faster than the full network scan.

Table 8.4 Computational performance of the partial network scan approach

Impact area parameter k	1	2	3	**5**	7	9	10
Computational time t_1 (min)	7.6	9.3	10.2	**11.0**	13.7	26.5	34.0
Parameter λ	100	200	300	400	**500**	600	700
Computational time t_2 (min)	26	51	77	103	**129**	155	181

In view of the above results, it can be observed that the impact area size parameters k and number of candidate links λ have significant impacts on both the accuracy and computational performance of the impact area vulnerability analysis approach. The larger the values of k and λ, the higher the accuracy and the longer the computational time. Thus, it is necessary to consider the trade-off between the accuracy and computational performance in choosing appropriate values of k and λ. Some general guidelines are suggested as follow:

(1) The value of k should not be less than 3. It is based on the results of the sensitivity tests shown in Table 8.3.

(2) The value of k should not be larger than 10. When the value of k is larger than 10, the impact area network becomes quite large, and thus requires a considerable computational effort to conduct the impact area vulnerability analysis on the impact area networks. In addition, if the value of k is quite large, only a small improvement in accuracy could be achieved by increasing the value of k.

(3) The λ value should be at least 4 times larger than the given preferred number of the most critical links (see Table 8.3). A larger λ value may be required when the demand variations are quite large (e.g. $cv_{rs} \geq 0.5$), as the demand variations seems to have a negative effects on impact area vulnerability index (see Fig. 8.7 and 8.8).

(4) The λ value should not be too large (e.g. $\lambda = |A| * 50\%$). In the proposed impact area vulnerability analysis approach, the global vulnerability indexes are calculated for selected λ candidate links. If the λ value is too large, the computational advantage of the proposed approach will be limited. When $\lambda = |A|$, the proposed approach reduces to the traditional full network scan approach.

Further studies, however, should be carried out to justify the above guidelines and investigate the strategies for choosing the appropriate values of k and λ.

8.5. Summary

In this section, the problem of vulnerability analysis for large-scale and congested road networks with demand uncertainty has been investigated. The effects of stochastic demand and heterogeneous travellers' risk-taking behaviour on the network vulnerability were explicitly considered by employing the reliability-based user equilibrium (RUE) model presented in Chapter 7. According to the same behavioural rationale as that of the RUE model, a new vulnerability index was introduced to evaluate the consequences of link closure. Numerical examples of the Sioux Falls network made apparent that stochastic demand and travellers' risk-taking behaviour have a significant impact on network vulnerability analysis, especially under high network congestion and large demand variations. The numerical results indicated that the most critical links of congested road networks may be misidentified, if the impacts of stochastic demand and travellers' risk-taking

behaviour arising from link closure, are totally ignored.

The impact area vulnerability analysis approach for identifying the most critical links was proposed in this chapter. A new vulnerability index was given to evaluate the consequences of link closure within its impact area instead of the whole network. The case study based on the Hong Kong road network showed that the proposed impact area vulnerability index has a very strong positive correlation with the vulnerability index of the whole network. This result indicated that the critical links, identified by the impact area vulnerability index, can be considered as a potential set of vulnerable links and used to reduce the search space for determining the most critical links of the whole network. The results of the case study in Hong Kong indicated that the proposed impact area vulnerability analysis approach is more efficient in identifying the most critical links, with suitable models parameters, than the traditional full network scan approach.

9. Conclusions and Further Studies

9.1. Summary of Research Findings

Reliable shortest path finding in stochastic road networks is one of the fundamental issues in transportation network modelling. However, few reliable shortest path algorithms have been recorded in the literature. This research intends to develop efficient solution algorithms for solving four variants of reliable shortest path problems with different link travel time characteristics. These four variants of reliable shortest path problems are summarized in Table 9.1. The applicability of these developed reliable shortest path algorithms is then illustrated in this research with respect to route guidance system (Chapter 6), reliability-based traffic assignment (Chapter 7), and network vulnerability analysis (Chapter 8).

Table 9.1 Summary of reliable shortest path problems

Problems	Link travel time distributions		Presented in the thesis
RSPP	Stochastic static	Independent	Chapter 3
SD-RSPP		Spatial correlated	Chapter 4
TD-RSPP	Stochastic time-dependent	Temporal correlated	Chapter 5
ST-RSPP		Spatiotemporal correlated	Chapter 6

RSPP: Reliable shortest path problem

SD-RSPP: Spatially-dependent reliable shortest path problem

TD-RSPP: Time-dependent reliable shortest path problem

ST-RSPP: Spatiotemporal-dependent reliable shortest path problem

The research presented in this thesis contributed to the literature of reliable shortest path finding in both scientific and practical aspects. The major scientific findings of this research can be summarized as following five aspects:

- The non-additive difficulty involved in all variants of reliable shortest path problems can be tackled by multi-criteria shortest path approach. Three types of dominance conditions have been established. The first order stochastic dominance (FSD) is used to determine dominated paths for any type of travel time distributions. The Mean-Variance (M-V) and Mean-travel time Budget (M-B) dominance conditions can help reduce the number of generated non-dominated paths when travel times follow normal distributions. The established dominance conditions enable the use of generalized dynamic programming approach to solve the reliable shortest problems, and thus have important implications on the algorithm design.

- Four efficient multi-criteria A* algorithms, based on established dominance conditions, were proposed to exactly solve the four variants of reliable shortest problems. The proposed multi-criteria A* algorithms can determine the reliable shortest path without the requirement of generating all non-dominated paths in the entire network. Computational experiments showed that the proposed multi-criteria A* algorithms have a significant computational advantage over the existing reliable shortest path algorithms.

- It was shown that in congested urban road networks, travel time of a link is strongly correlated only with its neighboring links within a local impact area. This limited spatial dependence property can be utilized to well approximate path travel time standard deviation in stochastic road networks with spatial correlated link travel times. Numerical results demonstrated that the increase of the impact area size can significantly enhance the approximate accuracy of path travel time standard deviation, but at the cost of additional computational burden on reliable shortest path findings. Therefore, it is necessary to consider the trade-off between the accuracy and computational performance of the reliable shortest path finding in stochastic networks with spatial correlated link travel

times.

- The stochastic first in first out (S-FIFO) property of TD-RSPP was investigated. Stochastic link travel time model (S-LTM) is commonly used model in the literature for representing the link travel times in stochastic time-dependent (STD) road networks. In this model, link travel time distribution depends on the time instance vehicles entering the link; and the link travel time distribution is assumed to be fixed when vehicles travelling on the link. It was shown that link travel times under S-LTM may not be S-FIFO consistent because traffic conditions cannot be continuously updated when vehicles travelling on the link. The violation of such S-FIFO property disallows the applicability of the efficient multi-criteria A* algorithm to solve TD-RSPP. In this research, a new stochastic travel speed model (S-TSM) was proposed. In S-TSM, travel speeds of a link are assumed to be stable within each time interval, and thus travellers' experienced travel speeds can be varied when vehicles are travelling on the link. It has been proved that the proposed S-TSM can ensure the S-FIFO property of TD-RSPP.

- It was found in this research that TD-RSPP is not reversible. TD-RSPP can be classified into two variants based on different routing scenarios. The 'forward' variant is to determine the earliest arrival time and associated reliable shortest path for a given departure time. The 'backward' variant is to simultaneously determine the optimal departure time and reliable shortest path for a given preferred arrival time. In the literature, TD-RSPP is assumed to be reversible. The backward TD-RSPP is generally solved by algorithms designed for the forward problem using a backward search from destination to origin. It was demonstrated in this research that TD-RSPP is not reversible; and the backward search can result in a significant travel time budget bias and thus provide a sub-optimal results of reliable shortest path finding. In this research, the backward TD-RSPP was formulated and solved using an equivalent two-stage

approach. The optimality of this two stage approach was rigorously proved.

The major practical findings of this research can be summarized as following three aspects:

- The proposed reliable shortest path algorithms were applied to a route guidance system, namely reliable path searching system (RPSS). In the developed RPSS, real-time and historical travel information from a real-world advanced traveller information system (ATIS) was incorporated to consider the travel time spatiotemporal correlations among links. Case study in Hong Kong road network shown that the developed RPSS can take account travellers' various routing preferences in congested road networks with travel time uncertainties. Computational experiments demonstrated that the developed reliable shortest path algorithms can determine the reliable shortest path in real-world road network within a satisfied computational time, even for situations when a large number of users simultaneously request routing services.

- The proposed reliable shortest path algorithms were incorporated into reliability-based user equilibrium (RUE) models. In this research, an efficient algorithm was proposed to determine reliable shortest paths for multiple user classes with heterogeneous risk-taking behaviours. The proposed shortest path algorithm was capable of finding optimal paths for all user classes in one search process so as to avoid the repeated search process for each user class. The proposed shortest path algorithm was further incorporated into a path-based traffic assignment algorithm for solving RUE problems without requirement of path enumeration. Experimental results demonstrated that the proposed solution algorithms can efficiently achieve high accuracies of the RUE results within a reasonable computational time. The development of such traffic assignment algorithm can enable the use of advanced RUE models for long-term

transportation planning in real-world large-scale road networks with demand and/or supply uncertainties.

- The proposed reliable shortest path algorithms and RUE algorithms were applied to the network vulnerability analysis. A new impact area network vulnerability analysis approach was proposed to identify critical links in large-scale road networks with demand uncertainty. The proposed approach evaluates the consequences of a link closure within a given impact area instead of the whole network so as to reduce the search space for determining the critical links. The results of the case studies in real-world networks showed that the proposed impact area vulnerability approach is capable of identifying critical links in large-scale networks. The numerical results indicated that the most critical links of congested road networks may be misidentified, if the impacts of stochastic demand and travellers' risk-taking behaviour arising from link closure are totally ignored.

9.2. Recommendations for Further Studies

Although the research presented in this thesis covers a wide area of reliable shortest path finding, there remain many interesting questions and important issues for which answers have not been provided. A few of these issues are outlined below:

1. The computational performance of reliable shortest path algorithms proposed in this research need to be further investigated so as to improve their efficiency for practical applications. In this research, the F-heap data structure (Fredman and Tarjan, 1987) was adopted for implementing the priority queue of the multi-criteria A* algorithm. Employing another priority queue implementation technique, such as the approximation bucket (Zeng and Church, 2009), may yield better computer running time for the proposed solution algorithm. In the

literature, many efficient solution algorithms (Jing et al., 1998; Chan and Lim, 2007), based on a multi-hierarchical network approach have been proposed to solve the traditional shortest path problems with huge network size. The integration of this multi-hierarchical shortest path approach with the proposed reliable shortest algorithm is a topic for further work.

2. The extension of the proposed reliable shortest path algorithm into the in-vehicle navigation systems is an important area for further study. The reliable shortest path algorithm presented in this research (Chapters 3-6) are designed mainly for implementing the reliable path searching services in the pre-trip planning applications. These reliable shortest path algorithms may not provide the best solution algorithm for in-vehicle navigation systems, in which traffic conditions are updated continuously during journeys. In the context of in-vehicle navigation systems, the repeated use of proposed reliable shortest path algorithms can be computationally burdensome for on-board devices with limited computing power. A re-optimization technique that makes best use of the previous shortest path searching results could be a promising solution approach in this context, to significantly save computation time (Koenig et al., 2004; Miller-Hooks and Yang, 2005; Huang et al., 2007).

3. Another direction for future research is to predict link travel time distributions based on real-time and offline travel information. In the developed reliable path search system (Chapter 6), estimated mean link travel times are obtained from a real-world advanced traveller information system. The current link travel time variations are simply estimated by assuming that the coefficient of variation (CV) of link travel time distribution keep constant. This simple assumption may not well represent the stochasticity of link travel times in congested road networks. The development of advanced mathematical models (Khosravi et al., 2011; Du et al., 2012; Mazloumi et al., 2011) for accurately predicting link travel time

distributions is required for further research.

4. The reliable shortest path algorithm formulated during this research and presented in this thesis only provides reliable path searching services to individual travellers. Under an excessive market penetration scenario, some 'reliable shortest paths' may become more congested if a large number of informed travellers choose the same reliable paths (Li et al., 2010). The use of coordinated routing and other advanced functions such as coordinated ramp metering, coordinated traffic signal control, and transit dispatching may be a possible way to optimize traffic flows over the entire network (Peeta, 1994; Peeta and Ziliaskopoulos, 2001).

5. In this research, only the private car mode is considered in the reliable shortest path problems. In reality, many travellers in densely populated cities may conduct trips using private cars, public transits, subways or combined modes. Further study is required as an extension of the proposed reliable shortest path algorithms for multi-modal transportation networks. Several key issues concerning reliable shortest path findings in multi-modal transportation networks should be explicitly considered, including non-linear fare structures, probable transfers, seat availability and representation of waiting and boarding times at transit stations (Lo et al., 2003; Schmöcker et al., 2011).

6. In order to facilitate the presentation of essential ideas in the RUE assignment problems (Chapter 7), several simplistic assumptions were adopted. In this study, the correlations of path/link flows have not been considered. However, in urban transportation networks, such correlations may exist due to variations of OD demands (Lam et al., 2008) particularly under congestion conditions. Further research should integrate such correlations in the proposed RUE solution algorithm. It was also assumed in this study that the mean travel demands of

road users are fixed. However, in practice, the demands may be affected by the travel time budget of different road users. These situations can be captured by the elastic-demand function. The extension of the proposed RUE solution algorithm, to take account elastic-demand warrants further study (Babonneau and Vial, 2008).

7. The proposed vulnerability analysis given in Chapter 8 only considers the scenarios of single link closure. The consideration of multiple link closures in impact areas (due to certain natural disaster events) is an interesting extension of this study (Lleras-Echeverri and Sanchez-Silva, 2001; Sumalee and Watling, 2008). Another interesting extension is to take into account travellers' perception errors and day-to-day adjustment processes for modelling travellers' behavioural responses to link closure (Shao et al., 2006; Bie and Lo, 2010) particularly in terms of demand variations. Further studies are also required to investigate strategies for determining appropriate impact area and to validate the results of identified most critical links.

Appendix A

Proofs in Chapter 3 are given in this appendix.

Proposition 3.1 Given two paths $p_u^{ri} \neq p_v^{ri} \in P^{ri}$, $p_u^{ri} \succ p_v^{ri}$ if p_u^{ri} and p_v^{ri} satisfy $\Phi_{T_u^{ri}}^{-1}(\lambda) < \Phi_{T_v^{ri}}^{-1}(\lambda)$ for any confidence level $0 < \lambda < 1$.

Proof. $\Phi_{T_u^{ri}}^{-1}(\lambda) < \Phi_{T_v^{ri}}^{-1}(\lambda), \forall \lambda \in (0,1)$

$\Rightarrow \Phi_{T_u^{ri}}(b) > \Phi_{T_v^{ri}}(b), \forall b \in R^+$

$\Rightarrow \int_0^b (\Phi_{T_u^{ri}}(b-t) - \Phi_{T_v^{ri}}(b-t)) f_{T^{iw}} dt > 0, \ \forall b \in R^+, \ \forall p^{iw} \in P^{iw}, \ \forall w \in N$

$\Rightarrow \int_0^b \Phi_{T_u^{ri}}(b-t) f_{T^{iw}} dt - \int_0^b \Phi_{T_v^{ri}}(b-t) f_{T^{jw}} dt > 0, \forall b \in R^+, \forall p^{iw} \in P^{iw}, \ \forall w \in N$

$\Rightarrow \Phi_{T_u^{rw}}(b) - \Phi_{T_v^{rw}}(b) > 0, \ \forall b \in R^+, \ \forall p^{iw} \in P^{iw}, \ \forall w \in N.$

$\Rightarrow \Phi_{T_u^{rw}}^{-1}(\lambda) < \Phi_{T_v^{rw}}^{-1}(\lambda), \ \forall \lambda \in (0,1), \ \forall p^{iw} \in P^{iw}, \ \forall w \in N. \square$

Let $p_u^{rw} = p_u^{ri} \oplus p^{iw}$ and $p_v^{rw} = p_v^{ri} \oplus p^{iw}$ be two paths from origin r to node w going through the same sub-path p^{iw}. T_u^{rw} and T_v^{rw} denote travel time for these two paths respectively. We introduce following function to facilitate the proofs

$$g_{uv}^{ri}(p^{iw}) = \Phi_{T_u^{rw}}^{-1}(\alpha) - \Phi_{T_v^{rw}}^{-1}(\alpha) = (t_u^{ri} - t_v^{ri}) + Z_\alpha(\sqrt{(\sigma_u^{ri})^2 + (\sigma^{iw})^2} - \sqrt{(\sigma_v^{ri})^2 + (\sigma^{iw})^2})$$

The $g_{uv}^{ri}(p^{iw})$ has following properties

Lemma 3.1

(i) If $Z_\alpha \sigma_u^{ri} < Z_\alpha \sigma_v^{ri}$, and thus $g_{uv}^{ri}(p^{iw})$ is a monotonic increasing function, $g_{uv}^{ri}(p^{iw}) \in [\Phi_{T_u^{ri}}^{-1}(\alpha) - \Phi_{T_v^{ri}}^{-1}(\alpha), t_u^{ri} - t_v^{ri})$;

(ii) If $Z_\alpha \sigma_u^{ri} > Z_\alpha \sigma_v^{ri}$, and thus $g_{uv}^{ri}(p^{iw})$ is a monotonic decreasing function, $g_{uv}^{ri}(p^{iw}) \in (t_u^{ri} - t_u^{ri}, \Phi_{T_u^{ri}}^{-1}(\alpha) - \Phi_{T_v^{ri}}^{-1}(\alpha)]$;

(iii) If $Z_\alpha \sigma_u^{ri} = Z_\alpha \sigma_v^{ri}$, and $g_{uv}^{ri}(p^{iw}) = t_u^{ri} - t_v^{ri}$.

Proof. The gradient of $g_{uv}^{ri}(p^{iw})$ can be formulated as

$g_{uv}^{ri}{}'(p^{iw}) = \frac{\partial g_{uv}^{ri}(p^{iw})}{\partial (\sigma^{iw})^2} = \frac{Z_\alpha}{2} \frac{\sqrt{(\sigma_v^{ri})^2 + (\sigma^{iw})^2} - \sqrt{(\sigma_u^{ri})^2 + (\sigma^{iw})^2}}{\sqrt{(\sigma_u^{ri})^2 + (\sigma^{iw})^2} * \sqrt{(\sigma_v^{ri})^2 + (\sigma^{iw})^2}}$. Therefore, when $Z_\alpha \sigma_u^{ri} < Z_\alpha \sigma_v^{ri}$, we have $g_{uv}^{ri}{}'(p^{iw}) > 0$ and $\Phi_{T_u^{ri}}^{-1}(\alpha) - \Phi_{T_v^{ri}}^{-1}(\alpha) \le g_{uv}^{ri}(p^{iw}) < t_u^{ri} - t_v^{ri}$; when $Z_\alpha \sigma_u^{ri} > Z_\alpha \sigma_v^{ri}$, we have $g_{uv}^{ri}{}'(p^{iw}) < 0$ and $t_u^{ri} - t_v^{ri} < g_{uv}^{ri}(p^{iw}) \le \Phi_{T_u^{ri}}^{-1}(\alpha) - \Phi_{T_v^{ri}}^{-1}(\alpha)$; when $Z_\alpha \sigma_u^{ri} = Z_\alpha \sigma_v^{ri}$, we have $g_{uv}^{ri}{}'(p^{iw}) = 0$ and $g_{uv}^{ri}(p^{iw}) = t_u^{ri} - t_v^{ri}$.□

According to Definition 3.3, $p_u^{ri} \succ p_v^{ri}$ is equivalent to $g_{uv}^{ri}(p^{iw}) < 0$ for any path $p^{iw} \in P^{iw}$. We will prove $g_{uv}^{ri}(p^{iw}) < 0$ for following dominance conditions.

Proposition 3.2 (M-V dominance). Given an on-time arrival probability α and two paths $p_u^{ri} \ne p_v^{ri} \in P^{ri}$, $p_u^{ri} \succ p_v^{ri}$ if p_u^{ri} and p_v^{ri} satisfy either

(i) $t_u^{ri} \le t_v^{ri}$ and $Z_\alpha \sigma_u^{ri} < Z_\alpha \sigma_v^{ri}$ or

(ii) $t_u^{ri} < t_v^{ri}$ and $Z_\alpha \sigma_u^{ri} \le Z_\alpha \sigma_v^{ri}$

Proof. It can be easily followed by Lemma 3.1.□

Proposition 3.3 (M-B dominance). Given an on-time arrival probability α and two paths $p_u^{ri} \ne p_v^{ri} \in P^{ri}$, $p_u^{ri} \succ p_v^{ri}$ if p_u^{ri} and p_v^{ri} satisfy $t_u^{ri} \le t_v^{ri}$ and $\Phi_{T_u^{ri}}^{-1}(\alpha) < \Phi_{T_v^{ri}}^{-1}(\alpha)$.

Proof. When $Z_\alpha \sigma_u^{ri} < Z_\alpha \sigma_v^{ri}$ and $t_u^{ri} \le t_v^{ri}$ according to Lemma 3.1(i), we have $g_{uv}^{ri}(p^{iw}) < (t_u^{ri} - t_v^{ri}) \le 0$. When $Z_\alpha \sigma_u^{ri} > Z_\alpha \sigma_v^{ri}$ and $t_u^{ri} \le t_v^{ri}$ according to Lemma 3.1(ii), we have $g_{uv}^{ri}(p^{iw}) < (\Phi_{T_u^{ri}}^{-1}(\alpha) - \Phi_{T_v^{ri}}^{-1}(\alpha)) \le 0$. When $Z_\alpha \sigma_u^{ri} = Z_\alpha \sigma_v^{ri}$ and $\Phi_{T_u^{ri}}^{-1}(\alpha) < \Phi_{T_v^{ri}}^{-1}(\alpha)$, we have $t_u^{ri} < t_v^{ri}$. According to Lemma 3.1(iii), we have $g_{uv}^{ri}(p^{iw}) = (t_u^{ri} - t_v^{ri}) < 0$. Therefore, when $t_u^{ri} \le t_v^{ri}$ and $\Phi_{T_u^{ri}}^{-1}(\alpha) < \Phi_{T_v^{ri}}^{-1}(\alpha)$, we have $g_{uv}^{ri}(p^{iw}) < 0$, $\forall p^{iw} \in P^{iw}$, $\forall w \in N$. □

Appendix B

The Proofs in Chapter 4 are given in this appendix.

Let $p^{ij,k} = \{a^1, ..., a^k\}$ be a path from node i to node j consisting of k consecutive links, and $p_u^{rj,k+\lambda} = p_u^{ri,\lambda} \oplus p^{ij,k}$ be a path from origin r to node j going through sub-path $p^{ij,k}$. The first-order stochastic dominant (FSD) condition for SD-RSPP can be formally defined as below.

Proposition 4.1 (FSD condition). Given two paths $p_u^{rj,\lambda+k} \neq p_v^{rj,\eta+k} \in P^{rj}$, $p_u^{rj,\lambda+k} \succ p_v^{rj,\eta+k}$ if $p_u^{rj,k+\lambda}$ and $p_u^{rj,k+\eta}$ satisfy $\Phi_{T_u^{rj}}^{-1}(y) < \Phi_{T_v^{rj}}^{-1}(y), \forall y \in (0,1)$.

Proof. Given a path $\forall p^{j\ell} \in P^{j\ell}, \forall \ell \in N$ and an on-time arrival probability $\forall y \in (0,1)$, $\Phi_{T_u^{r\ell}}^{-1}(y) = t_u^{rj} + t^{j\ell} + Z_y \sqrt{(\sigma_u^{rj})^2 + 2\mathrm{cov}(p^{ij,k}, p^{j\ell}) + (\sigma^{j\ell})^2}$ holds under the k limited spatial dependence assumption. When $(\tilde{\sigma}^{j\ell})^2 = 2\mathrm{cov}(p^{ij,k}, p^{j\ell}) + (\sigma^{j\ell})^2$ is used, this limited spatial dependence formula is equivalent to the following spatial independence formula $\Phi_{p_u^{rj} \oplus \tilde{p}^{j\ell}}^{-1}(y) = t_u^{rj} + t^{j\ell} + Z_y \sqrt{(\sigma_u^{rj})^2 + (\tilde{\sigma}^{j\ell})^2}$. Therefore, according to Proposition 3.1, $p_u^{rj,\lambda+k}$ dominates $p_v^{rj,\eta+k}$. □

In addition to the FSD condition, following mean-variance (M-V) dominance exists:

Proposition 4.2. (M-V dominance) Given an on-time arrival probability α and two paths $p_u^{rj,\lambda+k} \neq p_v^{rj,\eta+k} \in P^{rj}$, $p_u^{rj,\lambda+k} \succ p_v^{rj,\eta+k}$ if $p_u^{rj,k+\lambda}$ and $p_u^{rj,k+\eta}$ satisfy either

(i) $t_u^{rj} \leq t_v^{rj}$ and $Z_\alpha \sigma_u^{rj} < Z_\alpha \sigma_v^{rj}$ or

(ii) $t_u^{rj} < t_v^{rj}$ and $Z_\alpha \sigma_u^{rj} \leq Z_\alpha \sigma_v^{rj}$

Proof. (i) When $\alpha \geq 0.5$ and $Z_\alpha \sigma_u^{rj} \leq Z_\alpha \sigma_v^{rj}$, we have $Z_\alpha \geq 0$, $\sigma_u^{rj} \leq \sigma_v^{rj}$ and

$\sqrt{(\sigma_u^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, p^{j\ell}) + (\sigma^{j\ell})^2} \le \sqrt{(\sigma_v^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{j\ell}) + (\sigma^{j\ell})^2}$

$\forall p^{j\ell} \in P^{j\ell}, \forall \ell \in N$. When $\alpha < 0.5$ and $Z_\alpha \sigma_u^{rj} \le Z_\alpha \sigma_v^{rj}$, we have $Z_\alpha < 0$, $\sqrt{(\sigma_u^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{j\ell}) + (\sigma^{j\ell})^2} \ge \sqrt{(\sigma_v^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, p^{j\ell}) + (\sigma^{j\ell})^2}$ and $\sigma_u^{rj} \ge \sigma_v^{rj}$. Therefore, when $Z_\alpha \sigma_u^{rj} \le Z_\alpha \sigma_v^{rj}$, for any path $p^{j\ell} \in P^{j\ell}$, we have $Z_\alpha \sqrt{(\sigma_u^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{j\ell}) + (\sigma^{j\ell})^2} \le Z_\alpha \sqrt{(\sigma_v^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{j\ell}) + (\sigma^{j\ell})^2}$. Thus, if $t_u^{rj} < t_v^{rj}$ holds, $\Phi^{-1}_{T_u^{r\ell}}(\alpha) < \Phi^{-1}_{T_v^{r\ell}}(\alpha)$ $\forall p^{j\ell} \in P^{j\ell}, \forall \ell \in N$ is satisfied. According to Definition 3.3, we have $p_u^{rj,\lambda+k} \succ p_v^{rj,\eta+k}$.

(ii) The proof is similar to (i). □

Let $p_u^{rw,\lambda+2k} = p_u^{ri,\lambda} \oplus p^{ij,k} \oplus p^{jw,k}$ be a path from origin r to node w going through the sub-paths $p^{ij,k}$ and $p^{jw,k}$. The M-B dominance can be expressed as follows.

Proposition 4.7. (M-B dominance**)** Given an on-time arrival probability α and two paths $p_u^{rw,\lambda+2k} \ne p_v^{rw,\eta+2k} \in P^{rw}$, $p_u^{rw,\lambda+2k} \succ p_v^{rw,\eta+2k}$ if $p_u^{rw,\lambda+2k}$ and $p_v^{rw,\eta+2k}$ satisfy $t_u^{rj} \le t_v^{rj}$, $\Phi^{-1}_{T_u^{rj}}(\alpha) < \Phi^{-1}_{T_v^{rj}}(\alpha)$ and $\mathrm{cov}(T^{ij,k}, T_{jw}) \ge 0$.

Proof. According to Proposition 4.1, we have $p_u^{rj,\lambda+k} \succ p_v^{rj,\eta+k}$ if $Z_\alpha \sigma_u^{rj} < Z_\alpha \sigma_v^{rj}$, $t_u^{rj} \le t_v^{rj}$, and $\Phi^{-1}_{T_u^{rj}}(\alpha) < \Phi^{-1}_{T_v^{rj}}(\alpha)$ are satisfied. Therefore, in this scenario, we also have $p_u^{rw,\lambda+2k} \succ p_v^{rw,\eta+2k}$ according to Definition 3.3.

It can be proved that $p_u^{rw,\lambda+2k} \succ p_v^{rw,\eta+2k}$ also holds if $Z_\alpha \sigma_u^{rj} > Z_\alpha \sigma_v^{rj}$, $t_u^{rj} \le t_v^{rj}$, $\Phi^{-1}_{T_u^{rj}}(\alpha) < \Phi^{-1}_{T_v^{rj}}(\alpha)$ and $\mathrm{cov}(T^{ij,k}, T^{jw,k}) \ge 0$ are satisfied. According to Definition 3.3, $p_u^{rw,\lambda+2k} \succ p_v^{rw,\eta+2k}$ is equivalent to $g_{uv}^{rw}(p^{w\ell}) = \Phi^{-1}_{T_u^{r\ell}}(\alpha) - \Phi^{-1}_{T_v^{r\ell}}(\alpha) < 0$, $\forall p^{w\ell} \in P^{w\ell}$. The gradient of $g_{uv}^{rw}(p^{w\ell})$ can be formulated as

$$g_{uv}^{rw}{}'(p^{w\ell}) = \frac{Z_\alpha}{2} \frac{\sqrt{(\sigma_v^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{jw,k}) + (\sigma^{j\ell})^2} - \sqrt{(\sigma_u^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{jw,k}) + (\sigma^{j\ell})^2}}{\sqrt{(\sigma_u^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{jw,k}) + (\sigma^{j\ell})^2} * \sqrt{(\sigma_v^{rj})^2 + 2\mathrm{cov}(T^{ij,k}, T^{jw,k}) + (\sigma^{j\ell})^2}}$$

As $Z_\alpha \sigma_u^{rj} > Z_\alpha \sigma_v^{rj}$, we have $g_{uv}^{rw}{}'(p^{w\ell}) < 0$ and thus $g_{uv}^{rw}(p^{w\ell})$ is a monatomic

decreasing function with respect to $(\sigma^{j\ell})^2 + 2\,\mathrm{cov}(T^{ij,k}, T^{jw,k})$. Since $\Phi^{-1}_{T_u^{rj}}(\alpha) < \Phi^{-1}_{T_v^{rj}}(\alpha)$, we have $g^{rw}_{uv}(\phi) < 0$. As $\mathrm{cov}(T^{ij,k}, T^{jw,k}) \geq 0$, we have $(\sigma^{j\ell})^2 + 2\,\mathrm{cov}(T^{ij,k}, T^{jw,k}) \geq 0$ and thus $g^{rw}_{uv}(p^{w\ell}) < 0, \forall p^{w\ell} \in P^{w\ell}$. Therefore, $p_u^{rw,2k} \succ p_v^{rw,2k}$ holds when $t_u^{rj} \leq t_v^{rj}$, $Z_\alpha \sigma_u^{rj} > Z_\alpha \sigma_v^{rj}$, $\Phi^{-1}_{T_u^{rj}}(\alpha) < \Phi^{-1}_{T_v^{rj}}(\alpha)$ and $\mathrm{cov}(T^{ij,k}, T^{jw,k}) \geq 0$ are satisfied.□

Proposition 4.5. Given a top hierarchical path $\forall \hat{p}_u^{rs,\lambda} \in H^t$, its travel time distribution is equivalent to that of the corresponding primal path $p_u^{rs,\lambda+k-1} \in G$.

Proof. Since $t^{\hat{p}}_{rs,u} = t_u^{ij,k-1} + \sum_{m=1}^{\lambda} t^{\hat{a},m}_{\ell q,u} = \sum_{m=1}^{k-1} t^m + \sum_{m=1}^{\lambda} t^{m+k-1} = \sum_{m=1}^{\lambda+k-1} t^m = t_u^{rs,\lambda+k-1}$ holds, $\hat{p}_u^{rs,\lambda}$ and $p_u^{rs,\lambda+k-1}$ have a same mean travel time. It can also be proved that these two paths also have a same travel time variance as follows.

$$(\sigma^{\hat{p}}_{rs,u})^2 = (\sigma_u^{ri,k-1})^2 + \sum_{m=1}^{\lambda} (\sigma^{\hat{a},m}_{\ell q,u})^2 + \sum_{m=2}^{\lambda} 2\,\mathrm{cov}(\hat{T}^{m-1}_{jw,u}, \hat{T}^{m}_{\ell q,u})$$

$$= (\sigma_u^{ri,k-1})^2 + (\sigma^k)^2 + \sum_{n=1}^{k-1} 2\,\mathrm{cov}(T^n, T^k) + \sum_{m=2}^{\lambda} (\sigma^{\hat{a},m}_{\ell q,u})^2 + \sum_{m=2}^{\lambda} 2\,\mathrm{cov}(\hat{T}^{m-1}_{jw,u}, \hat{T}^{m}_{\ell q,u})$$

$$= (\sigma_u^{rj,k})^2 + \sum_{m=2}^{\lambda} (\sigma^{\hat{a},m}_{\ell q,u})^2 + \sum_{m=2}^{\lambda} 2\,\mathrm{cov}(\hat{T}^{m-1}_{jw,u}, \hat{T}^{m}_{\ell q,u})$$

$$= (\sigma_u^{rj,k})^2 + (\sigma^{k+1})^2 + \sum_{n=1}^{k-1} 2\,\mathrm{cov}(T^{n+1}, T^{k+1}) + 2\,\mathrm{cov}(T^1, T^{k+1}) + \sum_{m=3}^{\lambda} (\sigma^{\hat{a},m}_{\ell q,u})^2 + \sum_{m=3}^{\lambda} 2\,\mathrm{cov}(\hat{T}^{m-1}_{jw,u}, \hat{T}^{m}_{\ell q,u})$$

$$= (\sigma_u^{rw,k+1})^2 + \sum_{m=3}^{\lambda} (\sigma^{\hat{a},m}_{\ell q,u})^2 + \sum_{m=3}^{\lambda} 2\,\mathrm{cov}(\hat{T}^{m-1}_{jw,u}, \hat{T}^{m}_{\ell q,u})$$

$$= \ldots$$

$$= (\sigma_u^{r\ell,\lambda+k-2})^2 + \sum_{m=\lambda}^{\lambda} (\sigma^{\hat{a},m}_{\ell q,u})^2 + \sum_{m=\lambda}^{\lambda} 2\,\mathrm{cov}(\hat{T}^{m-1}_{jw,u}, \hat{T}^{m}_{\ell q,u})$$

$$= (\sigma_u^{r\ell,\lambda+k-2})^2 + (\sigma^{\lambda+k-1})^2 + \sum_{n=1}^{k-1} 2\,\mathrm{cov}(T^{\lambda+n-1}, T^{\lambda+k-1}) + 2\,\mathrm{cov}(T^{\lambda-1}, T^{\lambda+k-1})$$

$$= (\sigma_u^{rs,\lambda+k-1})^2 \square$$

Appendix C

Proofs in Chapter 5 are given in this appendix as follows.

Proposition 5.1. In S-FIFO networks, the path travel time satisfies the S-FIFO property:

$y_1 < y_2 \Rightarrow \Phi^{-1}_{Y_s(y_1)}(\lambda) = y_1 + \Phi^{-1}_{T_u^{rs}(y_1)}(\lambda) < \Phi^{-1}_{Y_s(y_2)}(\lambda) = y_2 + \Phi^{-1}_{T_u^{rs}(y_2)}(\lambda), \forall \lambda \in (0,1)$.

Proof. Without loss of generality, we consider a path $p_u^{rs} = a_{ri} \oplus a_{is}$.

According to Theorem 5.1, we have

$y_1 < y_2 \Rightarrow \Phi^{-1}_{Y_i(y_1)}(\lambda) < \Phi^{-1}_{Y_i(y_1)}(\lambda), \forall \lambda \in (0,1)$

$\Rightarrow \Phi_{Y_i(y_1)}(b) - \Phi_{Y_i(y_2)}(b) > 0, \forall b \in R^+$

$\Rightarrow \int_0^b (\Phi_{Y_i(y_1)}(b-y) - \Phi_{Y_i(y_2)}(b-y)) f_{T_{is}(y)}(y)dy > 0, \quad \forall b \in R^+$

$\Rightarrow \int_0^b \Phi_{Y_i(y_1)}(b-y) f_{T_{is}(y)}(y)dy - \int_0^b \Phi_{Y_i(y_2)}(b-y) f_{T_{is}(y)}(y)dy > 0, \forall b \in R^+$

$\Rightarrow \Phi_{Y_s(y_1)}(b) - \Phi_{Y_s(y_2)}(b) > 0, \quad \forall b \in R^+$. □

Proposition 5.2 (FSD condition) Given two paths $p_u^{ri} \neq p_v^{ri} \in P^{ri}$ in a S-FIFO network, p_u^{ri} dominates p_v^{ri} if they satisfy $\Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{T_v^{ri}(y_r)}(\lambda), \forall \lambda \in (0,1)$.

Proof. $\Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{T_v^{ri}(y_r)}(\lambda), \forall \lambda \in (0,1)$

$\Rightarrow \Phi^{-1}_{Y_i^u(y_r)}(\lambda) = y_r + \Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{Y_i^v(y_r)}(\lambda) = y_r + \Phi^{-1}_{T_v^{ri}(y_r)}(\lambda), \forall \lambda \in (0,1)$

$\Rightarrow \Phi_{Y_i^u(y_r)}(b) > \Phi_{Y_i^v(y_r)}(b), \forall b \in R^+$

$\Rightarrow \int_0^b (\Phi_{Y_i^u(y_r)}(b-y) - \Phi_{Y_i^v(y_r)}(b-y)) f_{T^{iw}(y)}(y)dy > 0, \quad \forall b \in R^+, \quad \forall p^{iw} \in P^{iw}, \quad \forall w \in N$

$\Rightarrow \int_0^b \Phi_{Y_i^u(y_r)}(b-y) f_{T^{iw}(y)}(y)dy - \int_0^b \Phi_{Y_i^v(y_r)}(b-y) f_{T^{iw}(y)}(y)dy > 0, \forall b \in R^+, \quad \forall p^{iw} \in P^{iw}, \forall w \in N$

$\Rightarrow \Phi_{Y_w^u(y_r)}(b) - \Phi_{Y_w^v(y_r)}(b) > 0, \quad \forall b \in R^+, \quad \forall p^{iw} \in P^{iw}, \quad \forall w \in N$

$\Rightarrow \Phi^{-1}_{Y_w^u(y_r)}(\lambda) < \Phi^{-1}_{Y_w^v(y_r)}(\lambda), \quad \forall \lambda \in (0,1), \quad \forall p^{iw} \in P^{iw}, \forall w \in N$

$$\Rightarrow y_r + \Phi^{-1}_{T_u^{rw}(y_r)}(\lambda) < y_r + \Phi^{-1}_{T_v^{rw}(y_r)}(\lambda),\ \ \forall\lambda \in (0,1),\ \ \forall p^{iw} \in P^{iw}, \forall w \in N$$

$$\Rightarrow \Phi^{-1}_{T_u^{rw}(y_r)}(\alpha) < \Phi^{-1}_{T_v^{rw}(y_r)}(\alpha), \forall p^{iw} \in P^{iw},\ \ \forall w \in N$$

Therefore, according to Definition 2, p_u^{ri} dominates p_v^{ri} □

Proposition 5.3. Given a path p_u^{ri} and a adjacent link a_{ij}, $p_u^{rj} = p_u^{ri} \oplus a_{ij}$ satisfies $\Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{T_u^{rj}(y_r)}(\lambda),\ \ \forall\lambda \in (0,1)$.

Proof. $\Phi_{Y_j(y_r)}(b) = \int_0^b f_{Y_i(y_r)}(y)\Phi_{T_{ij}(y)}(b-y)dy < \int_0^b f_{Y_i(y_r)}(y)dy = \Phi_{Y_i(y_r)}(b),\ \ \forall b \in R^+$.

$$\Rightarrow\ \ \Phi^{-1}_{Y_i(y_r)}(\lambda) = y_r + \Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{Y_j(y_r)}(\lambda) = y_r + \Phi^{-1}_{T_u^{rj}(y_r)}(\lambda),\ \ \forall\lambda \in (0,1).$$

$$\Rightarrow\ \ \Phi^{-1}_{T_u^{ri}(y_r)}(\lambda) < \Phi^{-1}_{T_u^{rj}(y_r)}(\lambda),\ \ \forall\lambda \in (0,1).\ \square$$

Appendix D

In this appendix, two solution algorithms for solving all-to-all reliable shortest path problems referred in Chapter 7 are given.

The steps of label-selection label-setting reliable shortest path algorithm (*RSPP-LSLS*) are given below.

Algorithm *RSPP-LSLS*

Inputs: an origin node r, $|M|$ user classes

Returns: reliable shortest paths rooted at the origin for all user classes

Step 1. Initialization.

Create a path p_i^{rr} from r to itself, and set $t_i^{rr} := 0$, $(\sigma_{t,i}^{rr})^2 := 0$ and $\Phi_{rs,i,m}^{-1} := 0$. Add p_i^{rr} into label-vector P^{rr} and the list of candidate labels SE.

Step 2. Label selection.

If $SE = \phi$, then goto Step 4; otherwise goto Sep 3.

Take label $p_i^{ru} \in P^{ru}$ with minimum travel time budget $\Phi_{ru,i,m}^{-1}$ at node u from SE.

Step 3. Path extension.

For every outgoing link a of chosen node u (v denotes a successor node of node u)

Step 3.1 Generate a new label $p_i^{rv} \in P^{rv}$. Set $t_i^{rv} := t_i^{ru} + t_a$,

$$(\sigma_{t,i}^{rv})^2 := (\sigma_{t,i}^{ru})^2 + \left(\sigma_t^a\right)^2 \text{ and } c_i^{rv} := t_i^{rv} + z_{\hat{\alpha}}\sigma_{t,i}^{rv}.$$

Step 3.2 If $p_i^{rv} \in P^{rv}$ is acyclic, then goto Step 3.3; otherwise scan next link

Step 3.3 If p_i^{rv} is a non-dominated path under M-B dominance condition, then insert p_i^{rv} into P^{rv} and SE, and remove all paths M-B dominated by p_i^{rv} from P^{rv} and SE.

End for.

Goto Step 2.

Step 4. Determine the reliable shortest path for each O-D pair and user class. Stop.

The steps of node-selection label-correcting reliable shortest path algorithm (*RSPP-NSLC*) are given as follows.

Algorithm *RSPP-NSLC*

Inputs: an origin node r, $|M|$ user classes

Returns: reliable shortest paths rooted at the origin for all user classes

Step 1. Initialization:

Create a path p_i^{rr} from r to itself; set $t_i^{rr} := 0$, $(\sigma_{t,i}^{rr})^2 := 0$ and $\Phi_{rs,i,m}^{-1} := 0$; and add p_i^{rr} into label-vector P^{rr}.

Add label-vector P^{rr} into the list of candidate labels SE.

Step 2. Node selection:

If $SE = \phi$, goto Step 4; otherwise, continue.

Take label-vector P^{ri} from SE in a FIFO order.

Step 3. Path extension:

For every outgoing link a of chosen node u (v denotes a successor node of node u)

Call procedure $P_N^{rv} := Update(P^{ru}, a_{uv}, P^{rv})$ (referred to Update procedure of *RSPP-NA** algorithm in Chapter 3).

If $P_N^{rv} \neq \phi$, then add label-vector P^{rv} into SE.

End for

Go back to Step 2.

Step 4. Determine the reliable shortest path for each O-D pair and user class. Stop.

References

Abdel-Aty, M.A. and Abdalla, M.F., 2004, Modeling drivers' diversion from normal routes under ATIS using generalized estimating equations and binomial probit link function. *Transportation*, 31, pp. 327-348.

Abdel-Aty, M.A., Kitamura, R. and Jovanis, P.P., 1995, Investigating effect of travel time variability on route choice using repeated-measurement stated preference data. *Transportation Research Record*, 1493, pp. 39-45.

Ahuja, R.K., Magnanti, T.L. and Orlin, J.B., 1993, *Network flows: theory, algorithms, and applications*, Prentice Hall, New Jersey, USA.

Avineri, E. and Prashker, J.N., 2006, The impact of travel time information on travellers' learning under uncertainty. *Transportation*, 33, pp. 393-408.

Babonneau, F. and Vial, J.P., 2008, An efficient method to compute traffic assignment problems with elastic demands. *Transportation Science*, 42, pp. 249-260.

Bar-Gera, H., 2002, Origin-based algorithm for the traffic assignment problem. *Transportation Science*, 36, pp. 398-417.

Bates, J., Polak, J., Jones, P. and Cook, A., 2001, The valuation of reliability for personal travel. *Transportation Research Part E-Logistics and Transportation Review*, 37, pp. 191-229.

Bell, M.G.H., 1995, Alternatives to Dial's logit assignment algorithm. *Transportation Research Part B-Methodological*, 29, pp. 287-295.

Bell, M.G.H. and Iida, Y., 1997, *Transportation Network Analysis*, John Wiley & Sons, New York.

Bellman, R., 1958, On a routing problem. *Quarterly of Applied Mathematics*, 16, pp. 87-90.

Berdica, K., 2002, An introduction to road vulnerability: what has been done, is done and should be done. *Transport Policy*, 9, pp. 117-127.

Berdica, K. and Mattsson, L.-G., 2007, Vulnerability: a model-based case study of

the road network in stockholm. In: *Critical infrastructure: Reliability and vulnerability*, A.T. Murray and T.H. Grubesic (Eds.), Springer, New York, pp. 81–106.

Bie, J. and Lo, H.K., 2010, Stability and attraction domains of traffic equilibria in a day-to-day dynamical system formulation. *Transportation Research Part B-Methodological*, 44, pp. 90-107.

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

Chabini, I., 1998, Discrete dynamic shortest path problems in transportation applications: complexity and algorithms with optimal run time. *Transportation Research Record*, 1645, pp. 170-175.

Chabini, I. and Lan, S., 2002, Adaptations of the A* algorithm for the computation of fastest paths in deterministic discrete-time dynamic networks. *IEEE Transactions on Intelligent Transportation Systems*, 3, pp. 60-74.

Chan, E.P.F. and Lim, H., 2007, Optimization and evaluation of shortest path queries. *The VLDB Journal*, 16, pp. 343–369.

Chan, K.S., Lam, W.H.K. and Tam, M.L., 2009, Real-time estimation of arterial travel times with spatial travel time covariance relationships. *Transportation Research Record*, 2121, pp. 102-109.

Chang, S.E. and Nojima, N., 2001, Measuring post-disaster transportation system performance: the 1995 Kobe earthquake in comparative perspective. *Transportation Research Part A-Policy and Practice*, 35, pp. 475-494.

Chang, T.S., Nozick, L.K. and Turnquist, M.A., 2005, Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments. *Transportation Science*, 39, pp. 383-399.

Chen, A. and Ji, Z.W., 2005, Path finding under uncertainty. *Journal of Advanced Transportation*, 39, pp. 19-37.

Chen, A., Lee, D.H. and Jayakrishnan, R., 2002, Computational study of

state-of-the-art path-based traffic assignment algorithms. *Mathematics and Computers in Simulation*, 59, pp. 509-518.

Chen, A., Yang, C., Kongsomsaksakul, S. and Lee, M., 2007, Network-based accessibility measures for vulnerability analysis of degradable transportation networks. *Networks & Spatial Economics*, 7, pp. 241-256.

Chen, A. and Zhou, Z., 2009, A stochastic α-reliable mean-excess traffic equilibrium model with probabilistic travel times and perception errors. In: *Transportation and Traffic Theory 2009: Golden Jubilee*, W.H.K. Lam, S.C. Wong and H.K. Lo (Eds.), Springer, New York, pp. 169-195.

Chen, A. and Zhou, Z., 2010, The alpha-reliable mean-excess traffic equilibrium model with stochastic travel times. *Transportation Research Part B-Methodological*, 44, pp. 493-513.

Cheung, R.K., 1998, Iterative methods for dynamic stochastic shortest path problems. *Naval Research Logistics*, 45, pp. 769-789.

Chorus, C.G., Molin, E.J.E. and Wee, B.V., 2006, Travel information as an instrument to change car-drivers' travel choices: a literature review. *European Journal of Transport and Infrastructure Research*, 6, pp. 335-364.

Chowdhury, M.A. and Sadek, A.W., 2003, *Fundamentals of intelligent transportation systems planning*, Artech House, Boston, USA.

Connors, R.D. and Sumalee, A., 2009, A network equilibrium model with travellers' perception of stochastic travel times. *Transportation Research Part B-Methodological*, 43, pp. 614-624.

D'Este, G.M. and Taylor, M.A.P., 2001, Network vulnerability: an approach to reliability analysis at the level of national strategic transport networks. In: *1st International Symposium on Transportation Network Reliability*, M.G.H. Bell and Y. Iida (Eds.), Elsevier Science Bv, Kyoto, Japan, pp. 23-44.

Dafermos, S.C. and Sparrow, F.T., 1969, The traffic assignment problem for a general network. *Journal of Research of the National Bureau of Standards*, 73, pp. 91-118.

Daganzo, C.F., 2002, Reversibility of the time-dependent shortest path problem. *Transportation Research Part B-Methodological*, 36, pp. 665-668.

Danczyk, A., Liu, H.X., and Levinson, D., 2009, Unexpected cause, unexpected effect: empirical observations of twin cities traffic behavior after the I-35W bridge collapse and reopening. *Transportation*. Submitted for publication.

Dandy, G.C. and McBean, E.A., 1984, Variability of individual travel time components. *ASCE Journal of Transportation Engineering*, 110, pp. 340-357.

Dial, R.B., 1971, A probabilistic multipath traffic assignment model which obviates path enumeration. *Transportation Research*, 5, pp. 83-111.

Dial, R.B., 2006, A path-based user-equilibrium traffic assignment algorithm that obviates path storage and enumeration. *Transportation Research Part B-Methodological*, 40, pp. 917-936.

Dijkstra, E.M., 1959, A note on two problems in connexion with graphs. *Numerische Mathematic*, 1, pp. 269-271.

Dion, F. and Rakha, H., 2006, Estimating dynamic roadway travel times using automatic vehicle identification data for low sampling rates. *Transportation Research Part B-Methodological*, 40, pp. 745-766.

Du, L., Peeta, S. and Kim, Y.H., 2012, An adaptive information fusion model to predict the short-term link travel time distribution in dynamic traffic networks. *Transportation Research Part B-Methodological*, 46, pp. 235-252.

El Esawey, M. and Sayed, T., 2011, Travel time estimation in urban networks using limited probes data. *Canadian Journal of Civil Engineering*, 38, pp. 305-318.

Fan, Y.Y., Kalaba, R.E. and Moore, J.E., 2005a, Arriving on time. *Journal of Optimization Theory and Applications*, 127, pp. 497-513.

Fan, Y.Y., Kalaba, R.E. and Moore, J.E., 2005b, Shortest paths in stochastic networks with correlated link costs. *Computers & Mathematics with Applications*, 49, pp. 1549-1564.

Fan, Y.Y. and Nie, Y., 2006, Optimal routing for maximizing the travel time reliability. *Networks & Spatial Economics*, 6, pp. 333-344.

FHWA, 2006, Travel time reliability: making it there on time, all the time. Available online at: http://ops.fhwa.dot.gov/publications/tt_reliability/TTR_Report.htm (accessed 11 June 2011).

Florian, M., Constantin, I. and Florian, D., 2009, A new look at projected gradient method for equilibrium assignment. *Transportation Research Record*, 2090, pp. 10-16.

Fosgerau, M. and Karlstr, A., 2010, The value of reliability. *Transportation Research Part B-Methodological*, 44, pp. 38-49.

Frank, H., 1969, Shortest paths in probabilistic graphs. *Operations Research*, 17, pp. 583-599.

Fredman, M.L. and Tarjan, R.E., 1987, Fibonacci heaps and their uses in improved network optimization algorithms. *Journal of the ACM*, 34, pp. 596-615.

Friis-Christensen, A., Lucchi, R., Lutz, M. and Ostländer, N., 2009, Service chaining architectures for applications implementing distributed geographic information processing. *International Journal of Geographical Information Science*, 23, pp. 561–580.

Fu, L.P. and Rilett, L.R., 1998, Expected shortest paths in dynamic and stochastic traffic networks. *Transportation Research Part B-Methodological*, 32, pp. 499-516.

Fu, L.P., 2001, An adaptive routing algorithm for in-vehicle route guidance systems with real-time information. *Transportation Research Part B-Methodological*, 35, pp. 749-765.

Fu, L., Sun, D. and Rilett, L.R., 2006, Heuristic shortest path algorithms for transportation applications: state of the art. *Computers & Operations Research*, 33, pp. 3324-3343.

Gajewski, B.J. and Rilett, L.R., 2003, Estimating link travel time correlation: an application of bayesian smoothing splines. *Journal of Transportation and Statistics*, 7, pp. 53-70.

Gao, S. and Chabini, I., 2006, Optimal routing policy problems in stochastic

time-dependent networks. *Transportation Research Part B-Methodological*, 40, pp. 93-122.

Guerriero, F. and Musmanno, R., 2001, Label correcting methods to solve multicriteria shortest path problems. *Journal of Optimization Theory and Applications*, 111, pp. 589-613.

Gutierrez, E. and Medaglia, A.L., 2008, Labeling algorithm for the shortest path problem with turn prohibitions with application to large-scale road networks. *Annals of Operations Research*, 157, pp. 169-182.

Hall, R.W., 1986, The fastest path through a network with random time-dependent travel-times. *Transportation Science*, 20, pp. 182-188.

Hart, E.P., Nilsson, N.J. and Raphael, B., 1968, A formal basis for the heuristic determination of minimum cost paths. *IEEE Transaction on System Science and Cybernetics*, SSC-4, pp. 100-107.

Huang, B., Wu, Q. and Zhan, F.B., 2007, A shortest path algorithm with novel heuristics for dynamic transportation networks. *International Journal of Geographical Information Science*, 21, pp. 625-644.

Hunt, J.D., Brownlee, A.T. and Stefan, K.J., 2002, Responses to the centre street bridge closure: where the 'disappearing' traffic went. *Transportation Research Record*, 1807, pp. 51-58.

Hutson, K.R. and Shier, D.R., 2009, Extended dominance and a stochastic shortest path problem. *Computers & Operations Research*, 36, pp. 584-596.

Jackson, W.B. and Jucker, J.V., 1981, An empirical study of travel time variability and travel choice behaviour. *Transportation Science*, 16, pp. 460-475.

Jayakrishnan, R., Tsai, W.K., Prashker, J.N. and Rajadhyaksha, S., 1994, Faster path-based algorithm for traffic assignment. *Transportation Research Record*, 1443, pp. 75-83.

Jenelius, E., Petersen, T. and Mattsson, L.G., 2006, Importance and exposure in road network vulnerability analysis. *Transportation Research Part A-Policy and Practice*, 40, pp. 537-560.

Ji, Z., Kim, Y.S. and Chen, A., 2011, Multi-objective a-reliable path finding in stochastic networks with correlated link costs: a simulation-based multi-objective genetic algorithm approach (SMOGA). *Expert Systems with Applications*, 38, pp. 1515-1528.

Jing, N., Huang, Y.W. and Rundensteiner, E.A., 1998, Hierarchical encoded path views for path query processing: an optimal model and its performance evaluation. *IEEE Transactions on Knowledge and Data Engineering*, 10, pp. 409-432.

Kaparias, I., Bell, M.G.H. and Belzner, H., 2008, A new measure of travel time reliability for in-vehicle navigation systems. *Journal of Intelligent Transportation Systems*, 12, pp. 202-211.

Khosravi, A., Mazloumi, E., Nahavandi, S., Creighton, D. and van Lint, J.W.C., 2011, Prediction Intervals to Account for Uncertainties in Travel Time Prediction. *IEEE Transactions on Intelligent Transportation Systems*, 12, pp. 537-547.

Knight, T.E., 1974, An approach to the evaluation of changes in travel unreliability: a safety margin hypothesis. *Transportation*, 3, pp. 393-408.

Knoop, V., van Zuylen, H. and Hoogendoorn, S., 2008, The influence of spillback modelling when assessing consequences of blockings in a road network. *European Journal of Transport and Infrastructure Research*, 8, pp. 287-300.

Knoop, V.L., Snelder, M. and Van Zuylen, H.J., 2007, Comparison of link-level robustness indicators. In: *3th International Symposium on Transportation Network Reliability*, Hague, Netherlands.

Koenig, S., Likhachev, M. and Furcy, D., 2004, Lifelong planning A*. *Artificial Intelligence*, 155, pp. 93-146.

Kurauchi, F., Uno, N., Sumalee, A. and Y., S., 2009, Network evaluation based on connectivity vulnerability. In: *Transportation and Traffic Theory 2009: Golden Jubilee*, W.H.K. Lam, S.C. Wong and H.K. Lo (Eds.), Springer, New York, pp. 637-649.

Lam, W.H.K. and Huang, H.J., 1992, A combined trip distribution and assignment model for multiple user classes. *Transportation Research Part B-Methodological*, 26, pp. 275-287.

Lam, T.C. and Small, K.A., 2001, The value of time and reliability: measurement from a value pricing experiment. *Transportation Research Part E-Logistics and Transportation Review*, 37, pp. 231-251.

Lam, W.H.K., Chan, K.S. and Shi, J.W.Z., 2002, A traffic flow simulator for short-term travel time forecasting. *Journal of Advanced Transportation*, 36, pp. 265-291.

Lam, W.H.K., Shao, H. and Sumalee, A., 2008, Modeling impacts of adverse weather conditions on a road network with uncertainties in demand and supply. *Transportation Research Part B-Methodological*, 42, pp. 890-910.

Latora, V. and Marchiori, M., 2001, Efficient behavior of small-world networks. *Physical Review Letters*, 87, p. 4.

Lee, W.H., Tseng, S.S. and Tsai, S.H., 2009, A knowledge based real-time travel time prediction system for urban network. *Expert Systems with Applications*, 36, pp. 4239-4247.

Li, Z.C., Huang, H.J. and Lam, W.H.K., 2010, Modelling heterogeneous drivers' response to route guidance and parking information system in stochastic and time-dependent networks. *Transportmetrica*. Article in press.

Lim, S., Balakrishnan, H., Gifford, D., Madden, S. and Rus, D., 2008, Stochastic motion planning and applications to Traffic. In: *8th International workshop on the algorithmic foundations of Robotics*, Springer, Guanajuato, Mexico, pp. 483-500.

Lleras-Echeverri, G. and Sanchez-Silva, M., 2001, Vulnerability analysis of highway networks, methodology and case study. *Proceedings of the Institution of Civil Engineers-Transport*, 147, pp. 223-230.

Lo, H.K. and Chen, A., 2000, Reformulating the traffic equilibrium problem via a smooth gap function. *Mathematical and Computer Modelling*, 31, pp.

179-195.

Lo, H.K., Yip, C.W. and Wan, K.H., 2003, Modeling transfer and non-linear fare structure in multi-modal network. *Transportation Research Part B-Methodological*, 37, pp. 149-170.

Lo, H.K., Luo, X.W. and Siu, B.W.Y., 2006, Degradable transport network: travel time budget of travellers with heterogeneous risk aversion. *Transportation Research Part B-Methodological*, 40, pp. 792-806.

Müller-Hannemann, M. and Weihe, K., 2004, On the cardinality of the pareto set in bicriteria shortest path problems. *Annals of Operation Research*, 147, pp. 185-197.

Martins, E.Q.V., 1984, On a multicriteria shortest path problem. *European Journal of Operational Research*, 16, pp. 236-245.

Mazloumi, E., Rose, G., Currie, G. and Sarvi, M., 2011, An integrated framework to predict bus travel time and its variability using traffic flow data. *Journal of Intelligent Transportation Systems*, 15, pp. 75-90.

Miller-Hooks, E. and Mahmassani, H.S., 1998, Optimal routing of hazardous materials in stochastic, time-varying transportation networks. *Transportation Research Record*, 1645, pp. 143-151.

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

Miller-Hooks, E. and Yang, B.Y., 2005, Updating paths in time-varying networks given arc weight changes. *Transportation Science*, 39, pp. 451-464.

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

Miller, H.J. and Shaw, S.L., 2001, *Geographic information systems for transportation: principles and applications*, Oxford University Press, New York.

Mirchandani, P.B., 1976, Shortest distance and reliability of probabilistic networks. *Computers & Operations Research*, 3, pp. 347-355.

Montgomery, F.O. and May, A.D., 1987, Factors affecting travel times on urban radial routes. *Traffic Engineering and Control*, 28, pp. 452-458.

Ng, S.T., Cheu, R.L. and Lee, D.H., 2006, Simulation evaluation of the benefits of real-time traffic information to trucks during incidents. *Journal of Intelligent Transportation Systems*, 10, pp. 89-99.

Nie, Y. and Fan, Y.Y., 2006, Arriving-on-time problem - discrete algorithm that ensures convergence. *Transportation Research Record*, 1964, pp. 193-200.

Nie, Y. and Wu, X., 2009a, Reliable a priori shortest path problem with limited spatial and temporal dependencies. In: *Transportation and Traffic Theory 2009: Golden Jubilee*, W.H.K. Lam, S.C. Wong and H.K. Lo (Eds.), pp. 169-195.

Nie, Y. and Wu, X., 2009b, Shortest path problem considering on-time arrival probability. *Transportation Research Part B-Methodological*, 43, pp. 597-613.

Nikolova, E., 2006, Stochastic shortest paths via quasi-convex maximization. In: *Algorithms - ESA 2006 Proceedings*, Y. Azar and T. Erlebach (Eds.), Springer, New York, pp. 552-563.

Nikolova, E., 2009, Strategic algorithms. Massachusetts Institute of Technology.

Nikolova, E., 2010, High-performance heuristics for optimization in stochastic traffic engineering problems. In: *Large-Scale Scientific Computing*, I. Lirkov, S. Margenov and J. Wasniewski (Eds.), Springer, New York, pp. 352-360.

Nocedal, J. and Wright, S.J., 2006, *Numerical Optimization*, Springer, New York.

Noland, R.B. and Small, K.A., 1995, Travel-time uncertainty, departure time choice, and the cost of morning commutes. *Transportation Research Record*, 1493, pp. 150-158.

Opasanon, S. and Miller-Hooks, E., 2006, Multicriteria adaptive paths in stochastic, time-varying networks. *European Journal of Operational Research*, 173, pp.

72-91.

Orda, A. and Rom, R., 1990, Shortest-path and minimum-delay algorithms in networks with time-dependent edge-length. *Journal of the ACM*, 37, pp. 607-625.

Pal, R. and Sinha, K., 2000, Optimal design of freeway incident response systems. FHWA/IN/JTRP-99/10. US Department of Transportation, Washington, D.C.

Panicucci, B., Pappalardo, M. and Passacantando, M., 2007, A path-based double projection method for solving the asymmetric traffic network equilibrium problem. *Optimization Letters*, 1, pp. 171-185.

Peeta, S., 1994, System optimal dynamic traffic assignment in congested networks with advanced information systems. The University of Texas at Austin.

Peeta, S. and Ziliaskopoulos, A.K., 2001, Foundations of dynamic traffic assignment: the past, the present and the future. *Networks & Spatial Economics*, 1, pp. 233-265.

Polus, A., 1979, A study of travel time and reliability on arterial routes. *Transportation*, 8, pp. 141-151.

Polychronopoulos, G.H. and Tsitsiklis, J.N., 1996, Stochastic shortest path problems with recourse. *Networks*, 27, pp. 133-143.

Rakha, H., El-Shawarby, I., Arafeh, M. and Dion, F., 2006, Estimating path travel-time reliability. In: *2006 IEEE Intelligent Transportation Systems Conference*, Toronto, Canada, pp. 236-241.

Schmöcker, J.D., Fonzone, A., Shimamoto, H., Kurauchi, F. and Bell, M.G.H., 2011, Frequency-based transit assignment considering seat capacities. *Transportation Research Part B-Methodological*, 45, pp. 392-408.

Schrank, D. and Lomax, T., 2009, The 2009 urban mobility report. Texas Transportation Institute, The Texas A&M University.

Sen, S., Pillai, R., Joshi, S. and Rathi, A.K., 2001, A mean-variance model for route guidance in advanced traveller information systems. *Transportation Science*, 35, pp. 37-49.

Shao, H., Lam, W.H.K. and Chan, K.S., 2004, The problem of searching the reliable path for transportation networks with uncertainty. In: *9th Conference of the Hong Kong Society for Transportation Studies*, H. Yang and L.H. K. (Eds.), Hong Kong University Science &Technology, Hong Kong, China, pp. 226-234.

Shao, H., Lam, W.H.K., Meng, Q. and Tam, M.L., 2006a, Demand-driven traffic assignment problem based on travel time reliability. *Transportation Research Record*, 1985, pp. 220-230.

Shao, H., Lam, W.H.K. and Tam, M.L., 2006b, A reliability-based stochastic traffic assignment model for network with multiple user classes under uncertainty in demand. *Networks & Spatial Economics*, 6, pp. 173-204.

Sheffi, Y., 1985, *Urban transportation networks: equilibrium analysis with mathematical programming methods*, Prentice-Hall, New Jersey.

Sherali, H.D., Ozbay, K. and Subramanian, S., 1998, The time-dependent shortest pair of disjoint paths problem: complexity, models, and algorithms. *Networks*, 31, pp. 259-272.

Sherali, H.D. and Hill, J.M., 2009, Reverse time-restricted shortest paths: application to air traffic management. *Transportation Research Part C-Emerging Technologies*, 17, pp. 631-641.

Siu, B.W.Y. and Lo, H.K., 2008, Doubly uncertain transportation network: degradable capacity and stochastic demand. *European Journal of Operational Research*, 191, pp. 166-181.

Skriver, A.J.V. and Andersen, K.A., 2000, A label correcting approach for solving bicriterion shortest-path problems. *Computers & Operations Research*, 27, pp. 507-524.

Spearman, C., 1904, The proof and measurement of association between two things. *The American Journal of Psychology*, 15, pp. 72-101.

Sumalee, A. and Watling, D.P., 2008, Partition-based algorithm for estimating transportation network reliability with dependent link failures. *Journal of*

Advanced Transportation, 42, pp. 213-238.

Sung, K., Bell, M.G.H., Seong, M. and Park, S., 2000, Shortest paths in a network with time-dependent flow speeds. *European Journal of Operational Research*, 121, pp. 32-39.

Tam, M.L. and Lam, W.H.K., 2008, Using automatic vehicle identification data for travel time estimation in Hong Kong. *Transportmetrica*, 4, pp. 179-194.

Tam, M.L., Lam, W.H.K. and Lo, H.P., 2008, Modeling air passenger travel behavior on airport ground access mode choices. *Transportmetrica*, 4, pp. 135-153.

Taylor, M.A.P. and D'Este, G.M., 2004, Critical infrastructure and transport network vulnerability: developing a method for diagnosis and assessment. In: *2th International Symposium on Transportation Network Reliability (INSTR04)*, Christchurch and Queenstown, New Zealand, pp. 96–102.

Taylor, M.A.P., Sekhar, S.V.C. and D'Este, G.M., 2006, Application of accessibility based methods for vulnerability analysis of strategic road networks. *Networks & Spatial Economics*, 6, pp. 267-291.

Taylor, M.A.P., 2008, Critical transport infrastructure in urban areas: impacts of traffic incidents assessed using accessibility-based network vulnerability analysis. *Growth and Change*, 39, pp. 593-616.

Tilahun, N.Y. and Levinson, D.M., 2010, A moment of time: reliability in route choice using stated preference. *Journal of Intelligent Transportation Systems*, 14, pp. 179-187.

Tobler, W.R., 1970, A computer movie simulating urban growth in the detroit region. *Economic Geography*, 46, pp. 234-240.

Toledo, T. and Beinhaker, R., 2006, Evaluation of the potential benefits of advanced traveller information systems. *Journal of Intelligent Transportation Systems*, 10, pp. 173-183.

van Lint, J.W.C., 2008, Online learning solutions for freeway travel time prediction. *IEEE Transactions on Intelligent Transportation Systems*, 9, pp. 38-47.

Vanajakshi, L., Subramanian, S.C. and Sivanandan, R., 2009, Travel time prediction

under heterogeneous traffic conditions using global positioning system data from buses. *IET Intelligent Transport Systems*, 3, pp. 1-9.

Waller, S.T. and Ziliaskopoulos, A.K., 2002, On the online shortest path problem with limited arc cost dependencies. *Networks*, 40, pp. 216-227.

Wardrop, J.G., 1952, Some theoretical aspects of road traffic research. *Proceedings of the Institution of Civil Engineers*, 1, pp. 325-378.

Watling, D., 2006, User equilibrium traffic network assignment with stochastic travel times and late arrival penalty. *European Journal of Operational Research*, 175, pp. 1539-1556.

Wijeratne, A.B., Turnquist, M.A. and Mirchandani, P.B., 1993, Multiobjective routing of hazardous materials in stochastic networks. *European Journal of Operational Research*, 65, pp. 33-43.

Wu, X. and Nie, Y., 2009, Implementation issues for the reliable a priori shortest path problem. *Transportation Research Record*, 2091, pp. 51-60.

Yang, H. and Huang, H.J., 2004, The multi-class, multi-criteria traffic network equilibrium and systems optimum problem. *Transportation Research Part B-Methodological*, 38, pp. 1-15.

Yen, J.Y., 1971, Finding the k shortest loopless paths in a network. *Management Science*, 17, pp. 712-716.

Yue, Y. and Yeh, A.G.O., 2008, Spatiotemporal traffic-flow dependency and short-term traffic forecasting. *Environment and Planning B-Planning & Design*, 35, pp. 762-771.

Zeng, W. and Church, R.L., 2009, Finding shortest paths on real road networks: the case for A*. *International Journal of Geographical Information Science*, 23, pp. 531-543.

Zhan, F.B. and Noon, C.E., 1998, Shortest path algorithms: an evaluation using real road networks. *Transportation Science*, 32, pp. 65-73.

Zhong, R.X., 2010, Dynamic assignment, surveillance and control for traffic network with uncertainties. The Hong Kong Polytechnic University.

Zhou, Z. and Chen, A., 2008, Comparative analysis of three user equilibrium models under stochastic demand. *Journal of Advanced Transportation*, 42, pp. 239-263.

Ziliaskopoulos, A.K. and Mahmassani, H.S., 1996, A note on least time path computation considering delays and prohibitions for intersection movements. *Transportation Research Part B-Methodological*, 30, pp. 359-367.

Ziliaskopoulos, A.K., Mandanas, F.D. and Mahmassani, H.S., 2009, An extension of labeling techniques for finding shortest path trees. *European Journal of Operational Research*, 198, pp. 63–72.