

FINAL REPORT

Reliable Routing in Transit Networks

July 2nd, 2013

Yu (Marco) Nie, Qianfei Li and Peng Chen



Reliable Routing in Transit Networks

Prepared by

Marco (Yu) Nie, Qianfei Li, Peng (Will) Chen
Department of Civil & Environmental Engineering
Northwestern University

July 2, 2013

DISCLAIMER *The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.*

Contents

Acknowledgement	1
Disclaimer	2
Summary	3
1 Introduction	4
1.1 Background	4
1.2 Objectives	5
1.3 Potential impacts	6
1.4 Organization	7
2 Literature review	8
3 Methodology	11
3.1 Basic common-lines problem	11
3.2 Special cases	13
3.3 Representation of transit network	14
3.4 Algorithms for finding optimal hyperpaths in a network	16
4 Data and statistical analysis	21
4.1 Data sources	21
4.2 Transit data viewer	23

4.2.1	Example 1: Route 130 North	24
4.2.2	Example 2: Route 125 North	27
5	Headway distribution fitting	38
5.1	Illustration of the analysis	38
5.2	Headway distributions for the weekday morning peak period	42
5.2.1	Statistics of headway data	42
5.2.2	Distribution fitting results	43
5.3	Summary	45
6	Numerical experiments	47
6.1	Impacts of service regularity on route choice	47
6.2	Determination of the attractive set	48
6.3	Simulation results of route choice	49
6.4	An Illustrative hyperpath routing example	54
7	Case study of the CTA bus network	57
7.1	Illustrative example	57
7.2	Average performance in morning peak	58
7.3	Algorithm comparison	62
7.3.1	Results by two greedy methods	62
7.3.2	Efficiency	62
A	Derivation of the relationship between waiting time and headway distributions	64
B	VNET Manual	67
B.1	Installation	67
B.2	A Quick Tutorial	67
B.3	Getting Started	68

B.4	Network Panel	69
B.4.1	Select network type	69
B.4.2	Load network file	70
B.5	Applications Panel	70
B.5.1	Generic panel	71
B.5.2	GTFS Applications	72
B.5.3	NETGTFS Applications	74

List of Tables

5.1	Statistic values of different GOF tests	39
5.2	Number of best fits for various distributions	45
5.3	Average percentage deviation of the statistic value from best	45
6.1	Attributes of two transit lines	48
6.2	Route choice results for various service regularity levels	48
6.3	Line Attributes	49
6.4	Numerical results for various combinations of lines	49
6.5	Line attributes	51
6.6	The Exponential headway case	51
6.7	The Erlang headway case	52
6.8	The Deterministic headway case 1	52
6.9	The Deterministic headway case 2	53
6.10	The Deterministic headway case 3	53
6.11	The Deterministic headway case 2 with different running length	53
6.12	The Deterministic headway case 3 with different running lengthn	54
6.13	Line attributes of the hypothetic network	55
7.1	Trip performance from Ashland and Irving Park to Michigan and Grand	59
7.2	Average performance in North Suburbs to Downtown Chicago scenario	61
7.3	Average performance in South Suburbs to Downtown Chicago scenario	61
7.4	Average performance in West Suburbs to Downtown Chicago scenario	61

7.5	Average performance in Downtown Chicago to Downtown Chicago scenario	61
7.6	Computation time for Exponential headway distribution	62
7.7	Computation time for Erlang headway distribution	63

List of Figures

1.1	CTA BusTracker Web Interface	5
2.1	Illustration of the hyperpath concept in a three-stop transit network	9
3.1	Network representation of a basic transit stop served by two lines	15
4.1	Illustration of data sources for the case study	22
4.2	Route 130	25
4.3	Route 130 time-spatial trajectory	26
4.4	Route 130 segment analysis	28
4.5	Spatial headway analysis	29
4.6	Spatial headway analysis for the first segment	29
4.7	Route 130 on-time analysis	30
4.8	Spatial on-time analysis for Route 130	31
4.9	Stop dwell time analysis for Route 130	31
4.10	Spatial dwell time analysis for Route 130	32
4.11	Route 125	33
4.12	Time-spatial diagram for Route 125 North	34
4.13	Segment statistics for Route 125	34
4.14	Example of irregular segments on Route 125	35
4.15	Spatial headway for Route 125	35
4.16	Route 125 on-time analysis	36

4.17	Spatial on-time analysis for Route 125	37
4.18	Route 125 dwell time analysis	37
5.1	Fit comparison for different fitted headway distributions	40
5.2	Q-Q plots for different fitted headway distributions	41
5.3	Summary statistics of headway observations in weekday morning peak	44
6.1	Interface of the route choice simulation	50
6.2	A three-stop transit network	54
6.3	Shortest hyperpath under different headway assumptions	56
7.1	Shortest hyperpaths from Ashland and Irving Park to Michigan and Grand	58
7.2	Four representative zones in the greater Chicago area	60
B.1	User interface of VNET	69
B.2	Help window	70
B.3	VNET Object Selector window	72
B.4	Graph and Objects Properties window	73
B.5	All-to-one tree	74
B.6	All-to-one tree	75
B.7	Parameter setting window for transit routing	76
B.8	Parameter setting window for data viewer	77
B.9	Parameter setting window for data viewer	78

Acknowledgement

This work was funded by the Center for the Commercialization of Innovative Transportation Technology at Northwestern University, a University Transportation Center Program of the Research and Innovative Technology Administration of USDOT through support from the Safe, Accountable, Flexible, Efficient Transportation Equity Act (SAFETEA-LU).

Disclaimer

The contents of this report reflect the views of the authors, who are responsible for the facts and the accuracy of the information presented herein. This document is disseminated under the sponsorship of the Department of Transportation University Transportation Centers Program, in the interest of information exchange. The U.S. Government assumes no liability for the contents or use thereof.

Summary

The objectives of this project are (1) to make use of the newly emerging transit data sources for evaluating the variations in transit services (especially headway), and (2) to help passengers find optimal routing strategies to hedge against these service variations.

The project first develops a data analysis tool (`Transit Data Viewer`) that is capable of building and visualizing empirical distributions of key transit operational parameters (headway, segment running time, dwell time and deviation from schedule) using space-time trajectories of transit vehicles. Statistical analysis is then performed to fit the processed headway data using various distributions.

Based on the best fitted headway distribution, the project develops and implements a transit routing tool built on the notion of *hyperpath* (`Transit Router`). Note that transit systems are affected by variations in road traffic conditions and demand patterns, as well as major disruptions caused by extreme weather conditions, serious traffic accidents, unforeseeable mechanical failures and human errors. To cope with uncertainty, the proposed routing tool aims at finding an optimal hyperpath to minimize the *expected* journey time.

The proposed tools are evaluated in a large-scale case study built from real data provided by the Chicago Transit Authority.

Chapter 1

Introduction

1.1 Background

Travel reliability is a critical dimension in user experience of public transportation services. A recent survey of commuters from the Chicago metropolitan area (Nie et al. 2010) reveals that reliability is the second most important factor that affects commuters' route choice, next only to travel time. Transit systems are affected by variations in road traffic conditions and demand patterns, as well as major disruptions caused by extreme weather conditions, serious traffic accidents, unforeseeable mechanical failures and human errors. While these uncertainties could adversely disrupt transit services, their overall impacts are rarely documented and understood in existing systems. As a result, neither transit operators nor transit users are able to make proactive decisions to ensure travel reliability. Ignoring the impacts of uncertainties often result in misallocation of limited resources in the transit system. From the user point of view, the lack of reliability either encourages overly conservative risk-averse behavior or leads to uncomfortable, sometimes disastrous, disruptions. Not surprisingly, almost half of the commuters in the aforementioned survey describe their transit service as “unreliable”.

Thanks to the revolution in information technology, many transit agencies now have the capability to track their entire fleets, make short-term projections, archive the data and distribute passenger information, all in real time. Figure 1.1 shows the web interface of the Bus Tracker Application provided by Chicago Transit Authority (CTA), which employs the GPS-based auto-

matic vehicle location (AVL) data to project the arrival times of the next transit vehicle at any stop on any route. Similar passenger information applications can be found in other major US cities,

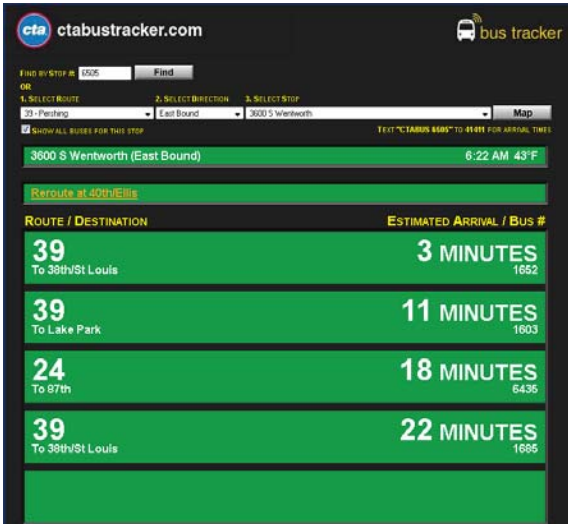


Figure 1.1: CTA BusTracker Web Interface

such as New York and Washington D.C. These new systems not only enable passengers to use transit information in the real time to improve their journey experience, but also make available a large amount of operational data that can be incorporated into better decision making for both passengers and agencies.

1.2 Objectives

The overarching goal of this project is to (1) make use of these newly emerging data sources for evaluating the variations in transit services (especially headway), and (2) help passengers find optimal routing strategies to hedge against these service variations. To this end, this project first develops a data analysis tool (Transit Data Viewer) that is capable of building and visualizing empirical distributions of key transit operational parameters (headway, segment running time, dwell time and deviation from schedule) using space-time trajectories of transit vehicles. Statistical analysis is then performed to fit the processed headway data using various distributions.

Based on the best fitted headway distribution, the project develops and implements a transit routing tool built on the notion of *hyperpath* (Transit Router). A *hyperpath* represents a sequence of routing strategies rather than a simple path consisting of stops. Routing based on hyperpath promises to make better use of availability of alternative routes in the transit systems. It also offers the flexibility to incorporate real-time information, such as the arrival times of all transit vehicles approaching a stop. It is worth noting that the boarding decision at a stop depends on

the waiting time as well as the remaining travel time to the destination once the selected transit line is boarded. This remaining travel time, in turn, is affected by future events such as waiting at subsequent transfers and travel between stops. As a result of the aforementioned system variations, the remaining travel time is not dictated by the published schedule. Accordingly, decisions have to be made according to what are *likely* to happen in the future. In light of this observations, the proposed tool copes with uncertainty by choosing an optimal hyperpath to minimize the *expected* journey time.

It is worth emphasizing that the proposed routing tool fully utilizes the archived operational data obtained from the new passenger information systems. In particular, these data will be used to characterize underlying stochastic properties of transit systems. Of these, the most important for our purpose are the headway distributions of all transit lines, which determine, among other things, the expected waiting time and line boarding probabilities at transfer stops. It has been commonly assumed in the literature that headways are exponentially distributed. With this simplifying assumption all headway distributions can be fully characterized without using any archived data - note that for the exponential distribution the standard deviation and mean are equal and can be reliably estimated using the scheduled headway. Moreover, exponentially distributed headways dramatically reduce the efforts for computing expected waiting times and boarding probabilities. However, our preliminary investigation shows that this assumption not only is poorly supported by empirical evidence, but also leads to sub-optimal route choices.

Finally, the proposed tool will be evaluated using a large-scale case study, which will be built from real data provided by the Chicago Transit Authority.

1.3 Potential impacts

The proposed tool helps transit users save travel time and improve travel reliability by making better informed routing decisions. In the long run, the improved user experience could attract more passengers to use mass transit, and thereby promoting sustainable transportation

by reducing driving.

The results from this project will also help understand the nature of uncertainty in large-scale transit systems. Specifically, statistical analysis will (1) determine the types of distributions that best fit headways and inter-stop travel times obtained from AVL data; and (2) reveal the key contributing factors to headway variations. With the above information, passengers could evaluate the reliability of their trips (such as the likelihood of arriving at their destination on-time), and agencies could assess the overall reliability performance of their systems. Understanding these stochastic properties could also help transit agencies design measures that aim to reduce variations.

1.4 Organization

The report is organized as follows. Chapter 2 briefly reviews the literature of the hyperpath problem. Chapter 3 introduces the methodology, including the concept of hyperpath, the basic analysis of common-lines problem with general headway distributions, and the algorithms for finding optimal paths in a transit network. Chapter 4 describes the data used in the case study, presents various analysis results obtained using *Transit Data Viewer*. Chapter 5 performs a statistical analysis to fit the headway data to various distributions. Various numerical experiments are performed in Chapter 6 to examine the impacts of service regularity on route choice, as well as verifying the analytical results using simulation. The results of a large-scale case study is reported in Chapter 7. A user manual for *Transit Data Viewer* and *Transit Router* is provided in Appendix B.

Chapter 2

Literature review

The concept of hyper-path appears to originate from Chriqui & Robillard (1975)'s study on common bus lines. Using a linear corridor similar to Figure 2.1 (with only two stops), this seminal paper shows that passengers can select a set of attractive lines and board the first arriving bus in that set in order to minimize the expected total journal time. Spiess & Florian (1989) extends this notion of strategy to a general transit network, namely, the choice of an attractive set of lines is considered at each node where boarding occurs.

Nguyen and Pallotino (1988) interpret the above strategy as a hyper-path, which is an acyclic directed graph. They propose both label correcting and label setting algorithms for finding the optimal hyper-paths between a pair of nodes. The above hyper-path routing model has been incorporated by many into transit assignment (see e.g. Nguyen & Pallottino 1988, Spiess & Florian 1989, de Cea & Fernández 1993, Wu et al. 1994, Cominetti & Correa 2001, Cepeda et al. 2006), which focuses on properly modeling the interaction between routing behavior and congestion in transit.

Central to the common-lines problem is the calculation of expected waiting time at stops. Early studies indicate that the expected waiting time for a single transit line can be estimated from the mean and variance of headway (see e.g. Welding 1957, Holroyd & Scraggs 1966, Osuna & Newell 1972, Seddon & Day 1974). When multiple lines are present, the expected waiting time depends on the probability of taking each line (line boarding probability), which in turn is

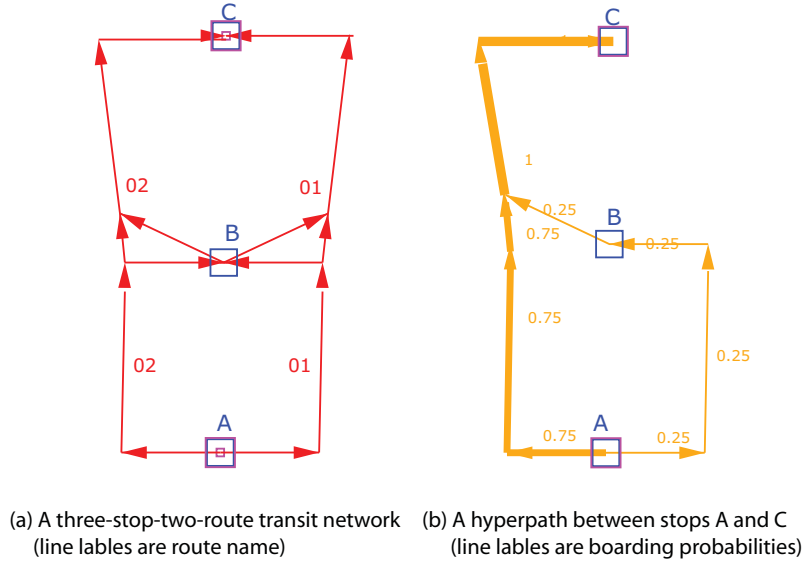


Figure 2.1: Illustration of the hyperpath concept in a three-stop transit network

affected by the availability of information, routing strategy and headway distributions. Of these, the type of headway distributions affects the computation of expected waiting time the most. With Assumption 2 above, headways are exponentially distributed. Consequently, the expected waiting time can be computed in closed form, which greatly improves the computational efficiency in optimal hyper-path search. However, pure random arrival of transit vehicles implies that the expected waiting time for any given line equals its average headway regardless of passengers' arrival time, which seems overly conservative except for bus services with very small headways (O'Flaherty & Mangan 1970).

Other types of headway distributions have been considered by several authors. Marguier & Ceder (1984) analyze the waiting time and line boarding probabilities in a two-line example, assuming the headways follow either power or Gamma distribution. Hickman & Wilson (1995) model bus headway using lognormal distributions in a simulation study. Gendreau (1984) proposed to approximate the line headway distributions by Erlang distribution, which is a special case of Gamma distribution but has better analytical tractability. Erlang distributions are also adopted in Bouzaïene-Ayari et al. (2001) and Gentile et al. (2005) to model route choice in the common-lines problem. More recently, Ruan & Lin (2009) fit a sample of observed headway

data collected in Chicago to four different distributions, and discover that Gamma distribution achieves the highest goodness-of-fit. Theirs appears to be the only recent empirical study on bus headway distributions identified in our reading of literature.

The role of real-time information in the common-lines problem was addressed in Hickman & Wilson (1995), which allow waiting time distributions to vary as passengers gather more information after arriving at the stop. Accordingly, passengers would only board the transit vehicle arriving at time t if the expected travel time upon boarding is smaller than the minimum expected travel time if the vehicle is skipped. Because each boarding decision invokes an evaluation of expected travel time, computing boarding probability is not analytically tractable. Instead, simulations are used to evaluate the value of information in a rather small case study. Gentile et al. (2005) assumes that the arrival time of next transit vehicles can be accurately projected once passengers arrive at the stop. Accordingly, passengers will always board the transit vehicle that has the minimum total travel time (projected waiting + expected travel time upon boarding). The assumption of “accurate real-time information” is adopted in this study because of its tractability, in terms of both computation and calibration. The impact of real-time information on transit routing is not considered in this project.

Chapter 3

Methodology

3.1 Basic common-lines problem

The basic common-lines problem that forms the cornerstone of the classical hyper-path models is built on the following assumptions Gentile et al. (2005).

Assumption 3.1 *Transit vehicles of different lines arrive at stops randomly (i.e. following a poisson distribution), and independent of each other.*

Assumption 3.2 *Bus line travel time is deterministic.*

Assumption 3.3 *Passengers arrive at stops randomly (i.e. following a poisson distribution). This implies that they do not adjust their arrival time according to published schedule, either because they have no access to or have no confidence in such information.*

Assumption 3.4 *Passengers do not have any real-time information (e.g. arrival times of the next vehicle), but can reliably estimate the remaining line travel time, i.e., the expected travel time from the stop to the destination, once boarded a vehicle of the line.*

Assumption 3.5 *Passengers aim to minimize their total expected travel time to the destination.*

Our methodology will adopt all but the first assumption. The reason why the first assumption is excluded is that it poorly aligns with empirical evidence, as this report will show later.

We consider route choice at a stop where the number of available transit lines is denoted as $L = \{1, 2, \dots, M\}$. Without loss of generality, we assume any line $l \in L$ can reach the passenger's destination D , regardless of the actual path from the stop onward. Let s_l be the expected waiting time upon boarding line l , and \tilde{h}_l and \tilde{w}_l be the random headway and waiting time associated with line l . The probability density functions of \tilde{h}_l and \tilde{w}_l are denoted as $g_l(\cdot)$ and $f_l(\cdot)$, respectively. Accordingly, $G_l(\cdot)$ and $F_l(\cdot)$ represent the cumulative distribution function of \tilde{h}_l and \tilde{w}_l . A passenger at stop i should first determine which lines are attractive, i.e., they should be considered in decision-making. Finally, let $R \in \Omega$ denote an attractive set, where Ω is the collection of all subsets of L .

When real-time information is not available, passengers will simply board the first arriving vehicle in R . Thus, the probability to board line l when $\tilde{w}_l = t$, denoted as $\gamma_l(t)$, is

$$\gamma_l(t) = \Pr(\tilde{w}_l = t) \prod_{k \in R/\{l\}} \Pr(\tilde{w}_k \geq t) = f_l(t) \prod_{k \in R/\{l\}} (1 - F_k(t)) \quad (3.1)$$

Hence, the overall probability of boarding l , denoted as π_l , is given by

$$\pi_l = \int_0^\infty \gamma_l(t) dt \quad (3.2)$$

The probability that the waiting time at the stop equals t , denoted as $W_R(t)$, is

$$\theta(t) = \sum_{l \in R} \gamma_l(t) \quad (3.3)$$

Therefore, the expected waiting time at the stop, denoted as $E[W_R]$, can be obtained as

$$E[W_R] = \int_0^\infty t \theta(t) dt \quad (3.4)$$

The expected total travel time to the destination, starting at the stop and corresponding to an attractive set R is then

$$u_R = E[W_R] + \sum_{l \in R} \pi_l s_l. \quad (3.5)$$

Since the line waiting time \tilde{w}_l is difficult to observe, its PDF is typically estimated from that of the random headway \tilde{h}_l using the following relationship (Larson & Odoni 1981)

$$f_l(t) = \lambda_l (1 - G_l(t)) \quad (3.6)$$

where $\lambda_l = 1/E(\tilde{h}_l)$, or the average arrival rate. Equation (3.6) may be derived using different approaches. Appendix A details one derivation based on renewal function.

Given $f_l(t)$, it is easy to see that the corresponding $F_l(t)$ can be obtained by

$$F_l(t) = \int_0^t f_l(w) dw \quad (3.7)$$

Therefore, for a general distribution, the calculation of π_l and $E[W_R]$ may involve a two-dimension integration (cf. Equation (3.1) and (3.7)). Since the calculation of attractive set R involves enumerating all possible combination of lines at the stop, the two-dimension integration is a computational challenge for large-scale network.

Passengers will always choose the best attractive set R^* such that their expected travel time u_{R^*} to the destination is minimized, i.e.,

$$u_{R^*} \leq u_R = E[W_R] + \sum_{l \in R} \pi_l s_l, \forall R \in \Omega \quad (3.8)$$

Searching R^* can be a computationally challenge task for large M since the size of Ω is 2^M . For special headway distributions, the efficiency of the search process may be significantly improved, as shown later.

3.2 Special cases

There are three special cases when closed form formulae are available. When \tilde{h}_l is exponentially distributed, we have closed form formulae for line boarding probabilities and expected waiting time by following the above equations (see e.g. Spiess & Florian 1989).

$$\pi_l = \frac{\lambda_l}{\sum_{k \in R} \lambda_k} \quad (3.9)$$

$$E[W_R] = \frac{1}{\sum_{k \in R} \lambda_k} \quad (3.10)$$

When the headway \tilde{h}_l is deterministic, the PDF of waiting time distribution can be calculated as

$$f_l(t) = \begin{cases} \lambda_l, & \forall 0 \leq t \leq 1/\lambda_l \\ 0, & \text{otherwise} \end{cases}$$

Then the route choice probability and expected waiting time is

$$\pi_l = \lambda_l \int_0^\infty \prod_{j \in L_n \setminus \{l\}} (1 - \lambda_j t) dt \quad (3.11)$$

$$E[W_R] = \int_0^\infty \prod_{j \in L_n} (1 - \lambda_j t) dt \quad (3.12)$$

When the headway \tilde{h}_l follows Erlang distribution, we have closed form for waiting time distribution. Note that Erlang distribution is a special case of Gamma distribution with shape parameter k_l being an integer number and rate parameter λ_l . Given that, the PDF of the waiting time distribution can be calculated

$$f_l(t) = \frac{\lambda_l}{k_l} e^{-\lambda_l t} \sum_{n=0}^{k_l-1} \frac{1}{n!} (\lambda_l t)^n \quad (3.13)$$

Accordingly, the CDF of waiting time can be calculated by integration.

$$\begin{aligned} F_l(t) &= \int_0^t \frac{\lambda_l}{k_l} e^{-\lambda_l w} \sum_{n=0}^{k_l-1} \frac{1}{n!} (\lambda_l w)^n dw \\ &= 1 - e^{-\lambda_l t} \sum_{n=0}^{k_l-1} \frac{1}{n!} \left(\left(1 - \frac{n}{k_l}\right) (\lambda_l t)^n \right) \end{aligned} \quad (3.14)$$

Given CDF and PDF of the waiting time distribution, the route choice probability can be calculated in Equation 3.2 with one-dimension integration.

3.3 Representation of transit network

A stop in a transit network is represented in this study as shown Figure 3.1. At the center of the stop layout is the so-called transfer node, represented using a solid blue square in the figure. For each transit line that passes a stop, two dummy nodes are created. The dummy nodes are corresponding to alighting and boarding maneuvers. For easy reference, the node associated with alighting is called a “transit node”, represented as a solid blue circle in the figure. The node associated with boarding is called “dwell node”, represented as solid grey circle in the figure. Accordingly, there are five types of links as detailed below.

Boarding link Connecting a transfer node to a dwell node

Alighting link Connecting a transit node to a transfer node

Dwell link Connecting a transit node to a dwell node

Transit link Connecting dwell node to the transit node at the next stop

Walking link Connecting one transfer node to a neighboring node through walking. Walking links are created between stops that are deemed by the modeler as “close enough” for walking at a normal speed (e.g. 5 km/hour).

In this study, we always assume a journey starts at a transfer node for simplicity.

The sets of transit, dwell and transfer nodes will be referred to as N_t , N_d and N_s , respectively.

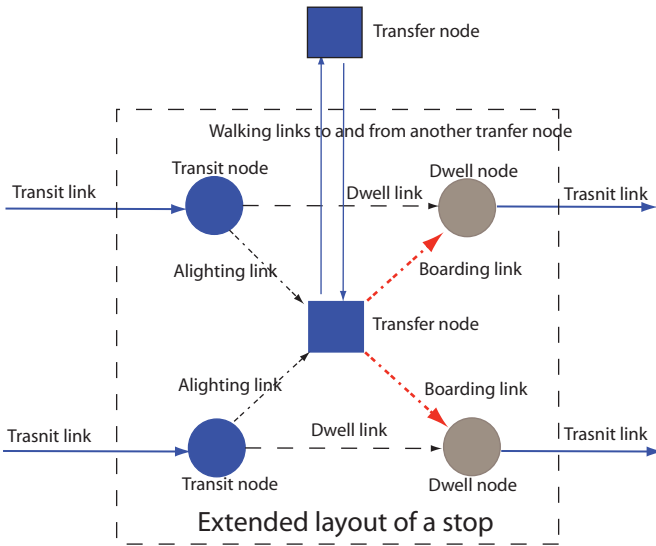


Figure 3.1: Network representation of a basic transit stop served by two lines

The set of all nodes $N = N_t \cup N_d \cup N_s$.

The sets of boarding, alighting, dwell, transit, and walking links are referred to as A_b , A_a , A_d , A_t and A_w , respectively, and the set of all links $A = A_b \cup A_a \cup A_d \cup A_t \cup A_w$. Furthermore, each link a is associated

with three attributes: τ_l as the running time, and h_l and v_l as the mean and

variance of the headway of the transit line associated with the line when

applicable. Importantly, these attributes are set accordingly to the link type, as explained below:

$a \in A_b$ h_a and v_a are mean and variance of the headway of the transit line associated with the line when applicable. $\tau_a = 0$.

$a \in A_a$ $h_a = 0$, $v_a = 0$. τ_a is set to the lost time due to alighting.

$a \in A_d$ $h_a = 0, v_a = 0, \tau_a = 0$.

$a \in A_t$ $h_a = 0, v_a = 0$. τ_a is the expected running time between the corresponding stops.

$a \in A_w$ $h_a = 0, v_a = 0$. τ_a is the walking time.

3.4 Algorithms for finding optimal hyperpaths in a network

We first explain how R^* , the optimal attractive set, may be identified in the common-lines problem. Algorithm 1 details the procedure of a widely adopted algorithm called the greedy method, which adds candidate lines into the attractive set according to an increasing order of s_l . It is clear that the above algorithm has a complexity of $O(M + \log(M))$ ($\log(M)$ refers to the

Algorithm 1 Greedy algorithm for finding R^*

- 1: **initialize:**
 - 2: Rank $L = 1, 2, \dots, M$ such that $s_1 < s_2, \dots, < s_M$.
 - 3: Set $R = \{1\}$. Compute u_R using Equation (3.5). Set $l = 2$.
 - 4: **while** $s_l < u_R$ **do**
 - 5: Set $R = R \cup l$
 - 6: Compute u_R using Equation (3.5).
 - 7: Set $l = l + 1$.
 - 8: **end while**
 - 9: Set $R^* = R$.
-

effort of ranking), which is much better than having to search every element in Ω . Unfortunately, the greedy method guarantees to find the optimal solution only when the headway is exponentially distributed. Note that the correctness of the greedy algorithm depends on the following relationship

$$u_Q < u_R \iff s_{p+1} < u_R, \forall p = \{1, \dots, M-1\}$$

where $R = \{1, \dots, p\}$ and $Q = R \cup \{p+1\}$. When EW_R and π_l are given by (3.10) and (3.9) respectively, we have

$$\begin{aligned} u_Q < u_R &\iff u_Q = \frac{1 + \sum_{l \in P} \pi_l s_l + \lambda_{p+1} s_{p+1}}{\sum_{l \in P} \lambda_l + \lambda_{p+1}} < \frac{1 + \sum_{l \in P} \pi_l s_l}{\sum_{l \in P} \lambda_l} \\ &\iff s_{p+1} < \frac{1 + \sum_{l \in P} \pi_l s_l}{\sum_{l \in P} \lambda_l} = u_R \end{aligned}$$

Since the above proof relies on (3.10) and (3.9), it does not hold for general expressions of $E(W_R)$ and π_l . In those cases, the greedy algorithm may only be considered a heuristic. Our numerical experiments include examples that show the greedy algorithm fails to identify the optimal solution when the headway distribution is Erlang.

For general headway distributions, the following brute-force enumeration algorithm has to be used to avoid sub-optimal solutions.

Algorithm 2 Enumeration algorithm for finding R^*

```

1: initialize:
2: Set  $u_R^* = \infty$ .
3: for all  $R \in \Omega$  do
4:   Compute  $u_R$  using Equation (3.5).
5:   if  $u_R < u_R^*$  then
6:     Set  $u_R^* = u_R$ ,  $R^* = R$ .
7:   end if
8: end for

```

We are now ready to present the algorithm for finding optimal hyperpaths to a given destination stop, denoted as S , from all other stops in a transit network. Like a standard shortest path problem, this problem can also be solved based on dynamic programming. The main difference is that the decision at a transfer node (i.e. the so-called embedded problem in dynamic programming) may involve choosing a subset of outgoing links, as opposed to just one link as in the standard shortest path problem. Either label-correcting or label-setting methods may be used. We first consider label-correcting algorithm, which is described in Algorithm 3. In the description, Q is the set of candidate nodes; R_i is the current attractive set at link i ; u_i is the optimal total expected travel time starting from stop i ; and $I(i)$ and $O(i)$ denote the set of incoming and outgoing links associated with node i .

A few remarks are in order here about the algorithm. First, the above algorithm is designed to deal with general headway distributions. A more efficient design may be achieved if all headway distributions are exponential. Second, the candidate list Q is implemented as a first-in-first-out queue. Other data structures may be used, such as a combination of two queues or a queue and

Algorithm 3 Label-correcting hyperpath (LCH) algorithm

```

1: initialize:
2:  $\forall i \in N$ , set  $R_i = \emptyset$ ;  $u_i = \inf$ 
3: Set  $u_S = 0$ ;  $Q = \{S\}$ 
4: while  $Q \neq \emptyset$  do
5:   Take node  $j$  from the front end of  $Q$ . Remove  $j$  from  $Q$ .
6:   for all links  $a \in I(j)$  do
7:     Set  $i$  as the tail node of  $a$ .
8:     if  $i \in N_t$  and  $a \in A_b$  then
9:       Construct a common-lines problem as follows.
10:      Set  $L = \{l | \forall l \in O(i) \cap A_b\}$ 
11:      For each  $l \in L$ , the line travel time  $s_l$  is set to  $\tau_l + u_k$  where  $l \in I(k)$ .
12:      Call Algorithm 1 or 2 to get  $R^*$  and  $u_{R^*}$  with  $L$  as the input.
13:      if  $u_{R^*} < u_i$  then
14:        Set  $u_i = u_{R^*}$ ,  $R_i = R^*$ .
15:        if  $i$  is not in  $Q$  then
16:          Insert  $i$  to the end of  $Q$ .
17:        end if
18:      end if
19:    else
20:      Set  $U = \tau_a + u_j$ .
21:      if  $U < u_i$  then
22:        Set  $u_i = U$ ,  $R_i = \{a\}$ .
23:        if  $i$  is not in  $Q$  then
24:          Insert  $i$  to the end of  $Q$ .
25:        end if
26:      end if
27:    end if
28:  end for
29: end while

```

a stack, one of which is used to hold nodes that have been scanned before.

The label-correcting algorithms require repeatedly visiting nodes to update their labels. In the worst case, it requires up to $\|N - 1\|$ visits per node to find the optimal path. This property could become a performance bottleneck as the network become denser (i.e. higher link to node ratio), although label-correcting algorithms are competitive on typical highway networks which has a link-to-node ratio of about 3 to 4. Transit networks, by construction, would have a much higher link-to-node ratio because of the common lines. To improve computational performance, this study also implements a label-setting algorithm, as described in Algorithm 4. Note that the label-setting and label-correcting algorithm differ from each other only on the implementation of Q .

Algorithm 4 Label-setting hyperpath (LSH) algorithm

```

1: initialize:
2:  $\forall i \in N$ , set  $R_i = \emptyset$ ;  $u_i = \inf$ 
3: Set  $u_S = 0$ ;  $Q = \{S\}$ , which is implemented as a binary tree ranked based on  $u_i$ .
4: while  $Q \neq \emptyset$  do
5:   Take the first node  $j$  ( $u_j$  is the minimum among all  $j \in Q$ ). Remove  $j$  from  $Q$ .
6:   for all links  $a \in I(j)$  do
7:     Set  $i$  as the tail node of  $a$ .
8:     if  $i \in N_i$  and  $a \in A_b$  then
9:       Construct a common-lines problem as follows.
10:      Set  $L = \{l \mid \forall l \in O(i) \cap A_b\}$ 
11:      For each  $l \in L$ , the line travel time  $s_l$  is set to  $\tau_l + u_k$  where  $l \in I(k)$ .
12:      Call Algorithm 1 or 2 to get  $R^*$  and  $u_{R^*}$  with  $L$  as the input.
13:      if  $u_{R^*} < u_i$  then
14:        Set  $u_i = u_{R^*}$ ,  $R_i = R^*$ .
15:        if  $i$  is not in  $Q$  then
16:          Insert  $i$  into  $Q$  based on  $u_i$ .
17:        else
18:          Remove  $i$  from  $Q$ , and re-insert it back based on the updated  $u_i$ .
19:        end if
20:      end if
21:    else
22:      Set  $U = \tau_a + u_j$ .
23:      if  $U < u_i$  then
24:        Set  $u_i = U$ ,  $R_i = \{a\}$ .
25:        if  $i$  is not in  $Q$  then
26:          Insert  $i$  into  $Q$  based on  $u_i$ .
27:        else
28:          Remove  $i$  from  $Q$ , and re-insert it back based on the updated  $u_i$ .
29:        end if
30:      end if
31:    end if
32:  end for
33: end while

```

Chapter 4

Data and statistical analysis

4.1 Data sources

Two main data sources from Chicago Transit Authority will be used to construct the case study. The first source is the General Transit Feed Specification (GTFS) data published by CTA through Google's GTFS project. The GTFS data, which contains schedules and associated geographic information, will be used to create the transit network topology. In general, GTFS feeder may contain the following files: agency, stops, routes, trips, stop_times, calendar, calendar_dates, fare_attributes, fare_rules, shapes, frequencies and transfers. A detailed illustration of GTFS feeder can be found at <https://developers.google.com/transit/gtfs/>. Some important concepts are illustrated regarding GTFS files, i.e. pattern, route and trip. Pattern is a geographic sequence of points. Each pattern defines a unique driving path for a bus. Route is the bus number or name. However, one thing worth noting is that one individual route may contain multiple patterns. A route may have rush hour pattern, non rush hour pattern and weekend pattern. Trip, on the other hand, is a specific incidence of bus run. Multiple trips consists of the operation of a bus. The trip ID uniquely defines the starting time and pattern of the bus.

A software tool called GTFSBuilder has been developed to visualize any GTFS data set and convert it to a network representation (NETGTFS) consistent with Figure 3.1. In general, a NETGTFS file includes info, link, node, route, shape, stop and trip. Figure 4.1-(a) shows the CTA stops, bus and metra lines active between 6 AM to 6 PM on weekdays, generated using GTFS-

Builder.

We also note that the same route may run in different patterns during a day, for example, the route may skip some stops during a period of a day. Since the schedule for different patterns of the same route may vary very much, we group the trajectories stopping at the exactly same stops into one pattern. Those trajectories in Figure 4.1-(b) actually correspond to a particular pattern for the bus route 151. Consequently, each stop/pattern pair constructs a headway distribution.

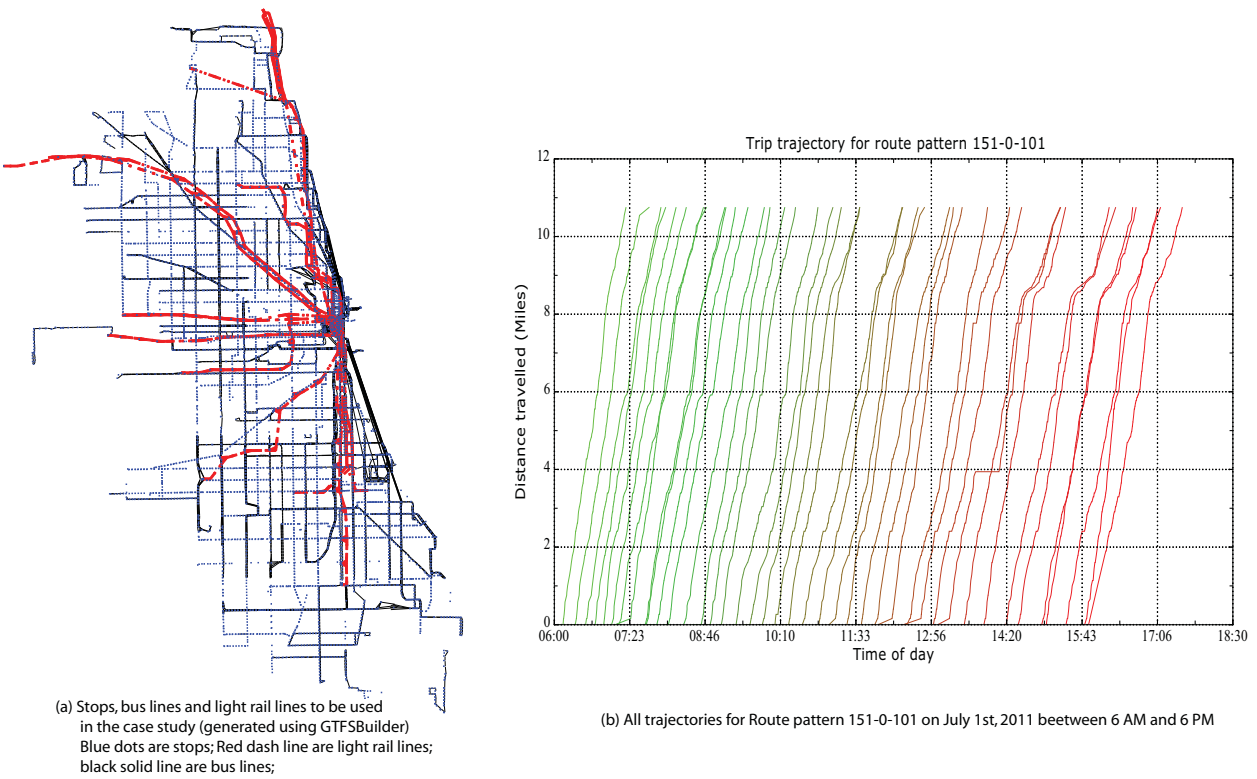


Figure 4.1: Illustration of data sources for the case study

The second source is the automatic vehicle location (AVL) data that feeds CTA’s Bus Tracker Program. The AVL data includes detailed transit vehicle space-time trajectories that can be used to derive various transit performance metrics. The time-space diagram in Figure 4.1-(b) visualizes such trajectories on CTA’s bus route 151 on July 1st, 2011 between 6 AM and 6 PM. The trajectories are from a one-month sample of the CTA’s bus AVL data. Note that from these trajectories the headways at any given stop, the inter-stop travel times, and the dwell times around a stop can all

be retrieved. Since the AVL data is not perfect, for example, some trajectories are incomplete, we have developed some heuristics to extract the headway observations, which may result in some invalid values, e.g., non-positive values. For the purpose of the distribution fitting, we simply delete those invalid headway values. With many days of observations of the same trajectories, probabilistic distributions may be constructed. Figure 4.1-(c) shows the headway distribution on route 151 at the Stockton & Arlington stop based on the data in the July of 2011.

4.2 Transit data viewer

A Transit Data Viewer tool is developed to visualize the data. The tool is developed on the VNET platform, which a simple, flexible and extensible graphic user interface to support a wide variety of network-related applications, ¹. This section will focus on the data processing and visualization functions provided by Transit Data Viewer. A detailed manual of VNET can be found in Appendix B.

Transit Data Viewer can perform the following analysis based on a PostGres database that currently stores all bus running data in the July of 2011.

Time-space trajectory Time-space trajectory plots the bus's position against time for each scheduled bus trip. Both the scheduled and actual trajectories can be plotted and compared with each other. The scheduled time data plots the scheduled time of the vehicle. The vertical distance between lines reflects the headway of the bus.

Segment running time distribution Segment running time distribution plots the probability density function (PDF) and cumulative probability function (CDF) of a selected route segment.

Headway statistics Headway Statistics plot the PDF and CDF of the headway distribution at a selected stop of a route. Spatial headway statistics plot the mean, variance and percentiles of headway at a set of consecutive stops of a route.

¹See http://translab.civil.northwestern.edu/nutrend/?page_id=53 for more details.

On-time statistics On-time probability analysis focuses on the deviation between the scheduled and actual arrival times. The PDF and CDF of such deviations can be plotted. A similar spatial analysis as headway statistics can also be performed for a set of consecutive stops of a route.

Dwell time statistics Dwell time statistics plots CDF and PF of the dwell time at a stop, i.e. the stopping time at a stop to board and alight. Similarly, a spatial analysis can be performed for a set of consecutive stops of a route.

In what follows, two examples are provided for demonstration purpose. The first example is route 130 from Museum Campus to Ogilvie Transportation Center. The second example is route 125 North from Harrison St. to Water Tower Place.

4.2.1 Example 1: Route 130 North

Route 130 North bound is chosen as our second example for demonstration purpose. Figure 4.2(a) is the route schedule from CTA's website. Figure 4.2(b) is a screenshot from VNET, where the dots represent transit stops while the green line is Route 130 North bound.

After loading the route, we select all the trips of July 1st. for demonstration purpose. Figure 4.3(a) shows the time-spatial trajectory of Route 130 North. From the plot, we can see that the bus service is quite regular before 1PM. The lines are evenly distributed. However, irregularity grows obviously as afternoon rush hour approaches. The first observable irregularity happens at 3:45PM. The bus at 3:45PM is late from schedule, see Figure 4.3(b) which plots both the actual and scheduled bus trajectory. The space between the bus and its predecessor becomes wider. Unfortunately, the off-schedule is not adjusted accordingly afterwards. New buses are dispatched off schedule. As a result, serious bunching is observed. When bunching happens, two consecutive buses arrive close to each other. Another two bunchings are observed around 5PM, which might be result from the afternoon rush hour. In particular, the arterial streets of downtown Chicago becomes congested and vehicles are not able to move smoothly.

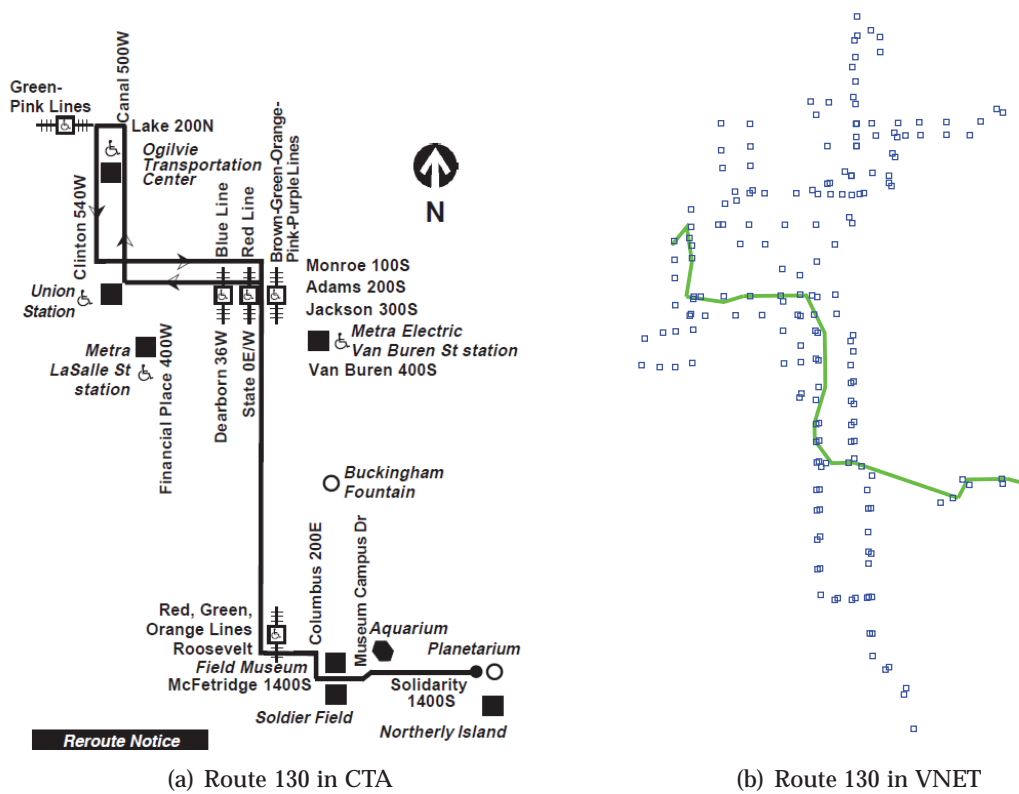
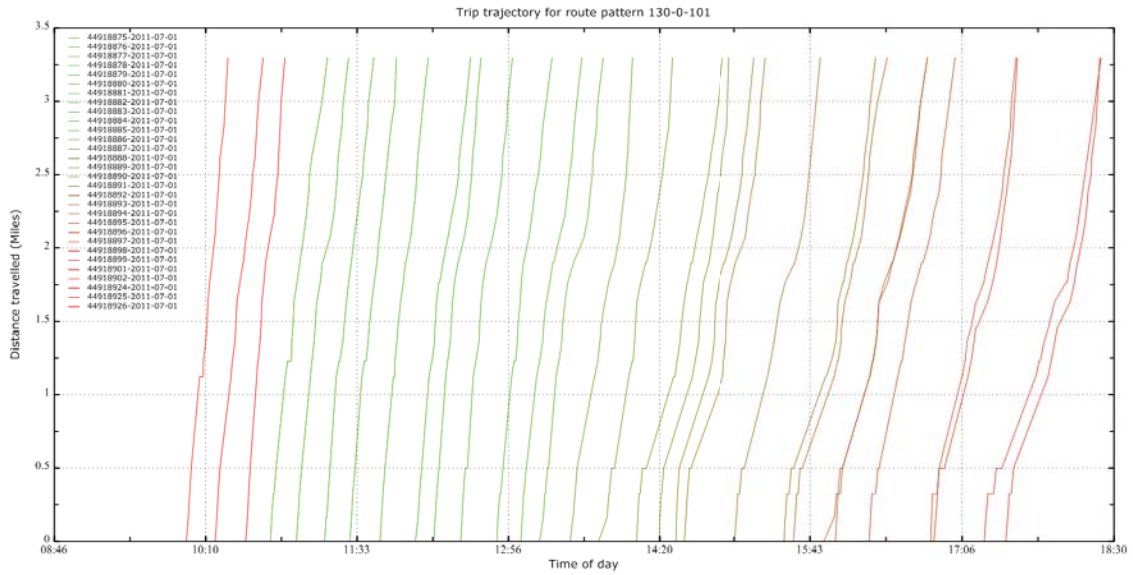
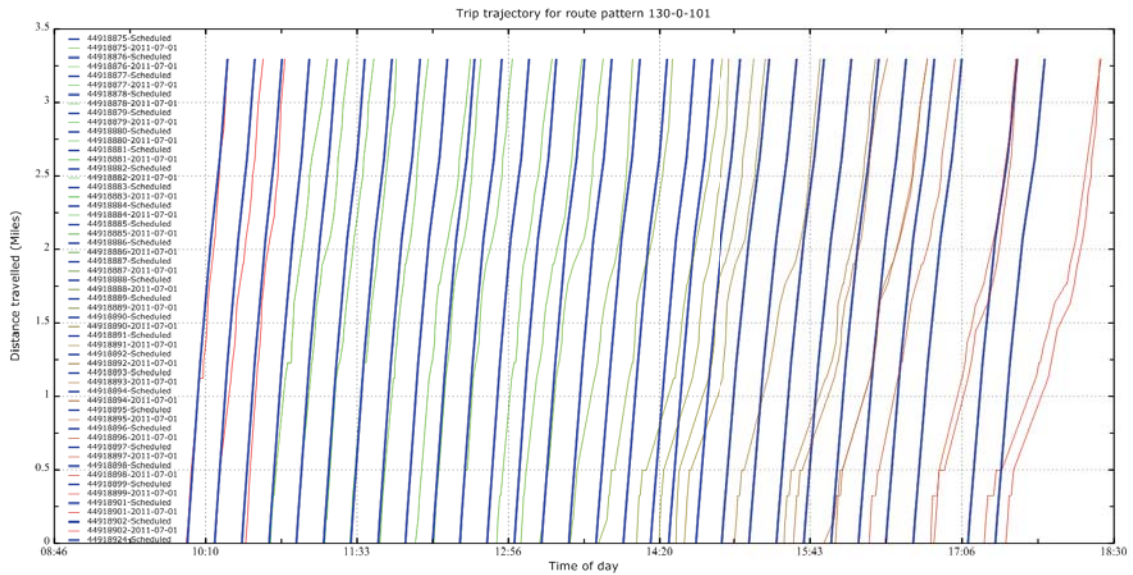


Figure 4.2: Route 130



(a) Actual time-spatial trajectory of Route 130



(b) Schedule and real time-spatial trajectory of Route 130

Figure 4.3: Route 130 time-spatial trajectory

We also conduct analysis on the segment statistics. Figure 4.4(a) represents the overall segment statistics. The travel time PDF and the travel speed PDF is bell-shaped. However, Figure 4.4(b) provides an irregular segment statistics. There is no obvious trend for the PDF of travel time. In addition, The PDF of travel speed is no longer centered but spread out. In fact, depending on the traffic condition, the segment statistics may vary between each other.

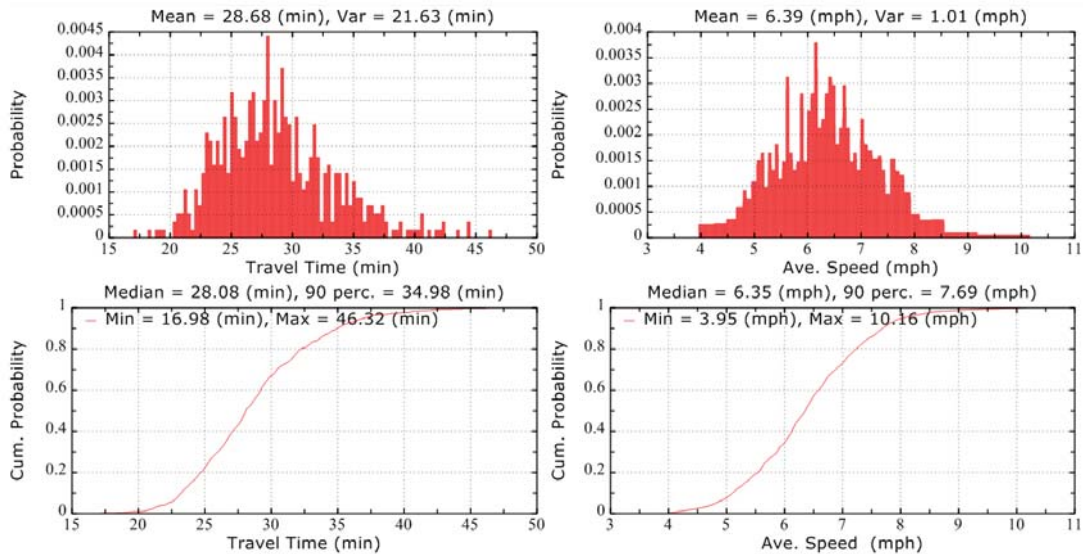
The next plot depicts the relationship between stop headway and stop distance from bus terminal, see Figure 4.5. (To show the plot, sort the stops based on cumulative distance.) One thing worth noting is the trend of the standard deviation (SD). In general, SD is expected to increase as the bus moves from the terminal. An intuitive explanation is due to the intervention of traffic disruption. As the bus moves downstream, the effect of traffic disruption accumulates and results in increasing SD. However, in the plot, a decreasing trend is observed before 0.2 miles. In particular, the decrease takes place on the first segment of the Route 130 North, see Figure 4.6.

Figure 4.7(a)- 4.7(c) plot the on-time probability distribution for three different stops along the route, from upstream to downstream. The mean deviation for the upstream stop is 1.33 mins. The mean deviation for the middle stop is 3.73 mins. The mean deviation for the downstream stop is 7.64 mins. The mean deviation increases as the stops move towards downstream. In the meanwhile, the overall trend of deviation can be plotted with Spatial on-time analysis tool, see Figure 4.8. The mean deviation, as well as percentile deviation, are plotted against the distance from the terminal. The same uphill trend is observed as the stops move towards the downstream.

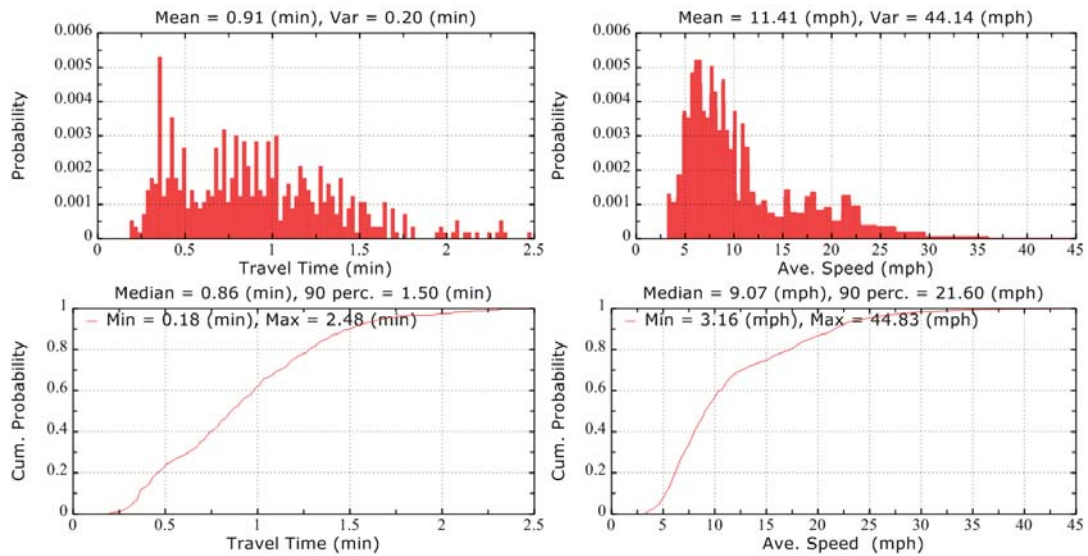
Stop dwell time and spatial dwell time are plotted in Figure 4.9 and Figure 4.10, respectively. No major trend is observed for spatial dwell time analysis. The dwell time at each individual stop is more related to the demand (boarding/alighting passengers). The distance from terminal has little impact on the dwell time.

4.2.2 Example 2: Route 125 North

Route 25 North bound is chosen as our second example. Figure 4.11(a) is the route schedule from CTA's website. Figure 4.11(b) is the screenshot from VNET, where the dots represent transit



(a) Overall segments statistics



(b) An irregular segment statistics

Figure 4.4: Route 130 segment analysis

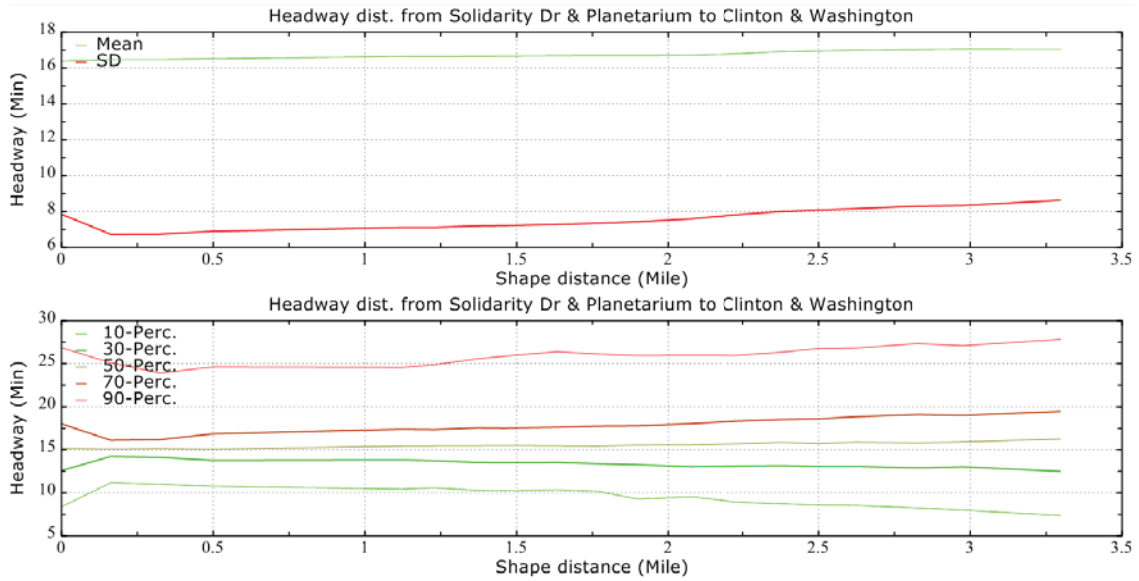


Figure 4.5: Spatial headway analysis

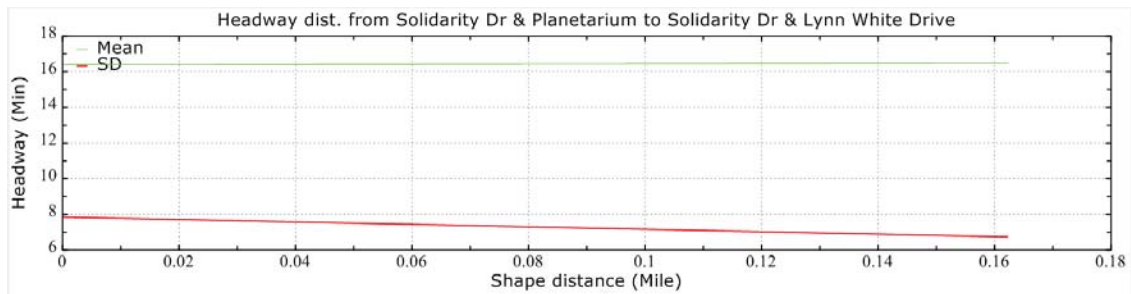
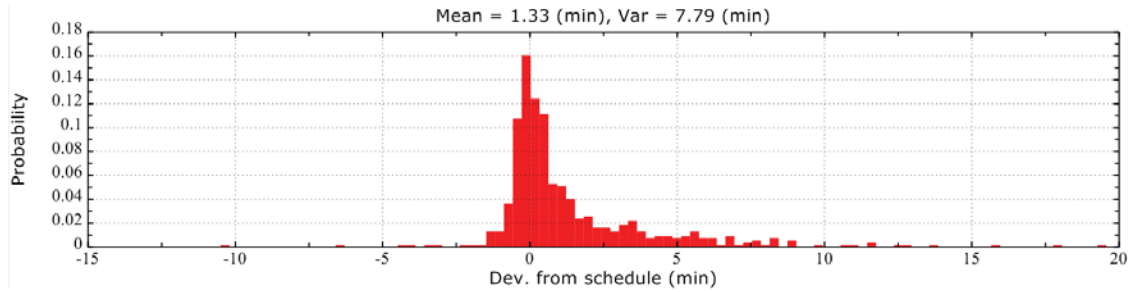
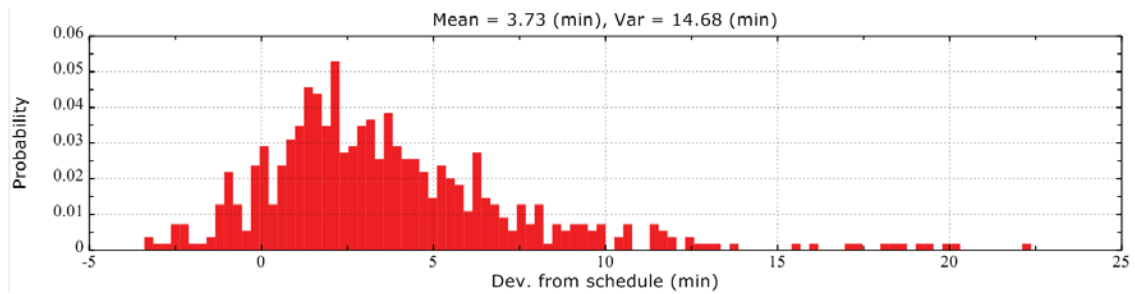


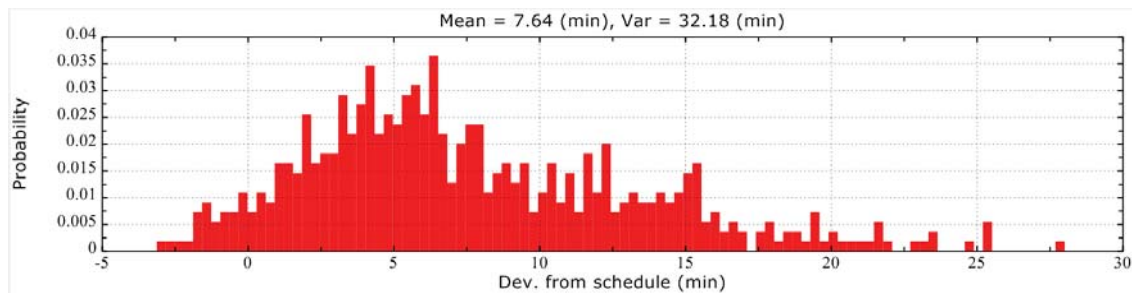
Figure 4.6: Spatial headway analysis for the first segment



(a) Deviation from schedule at an upstream stop



(b) Deviation from schedule at a middle stop



(c) Deviation from schedule at a middle stop

Figure 4.7: Route 130 on-time analysis

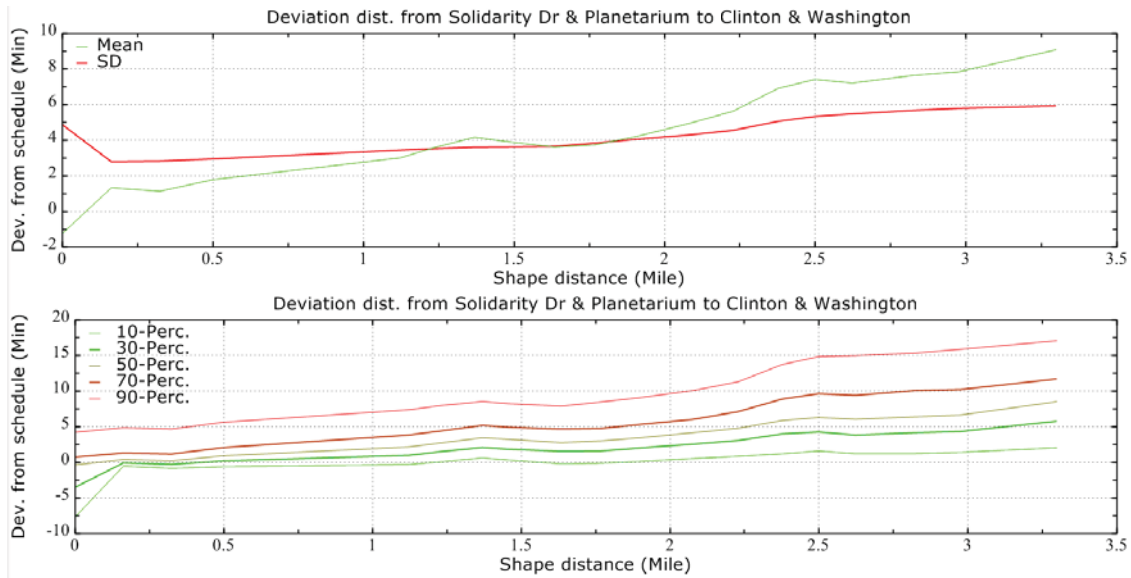


Figure 4.8: Spatial on-time analysis for Route 130

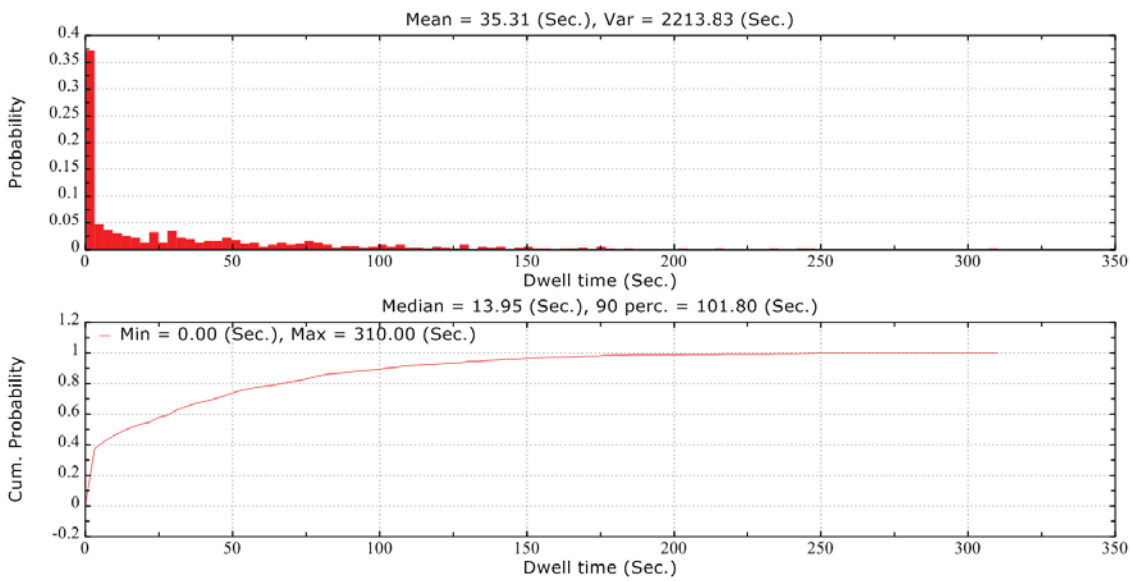


Figure 4.9: Stop dwell time analysis for Route 130

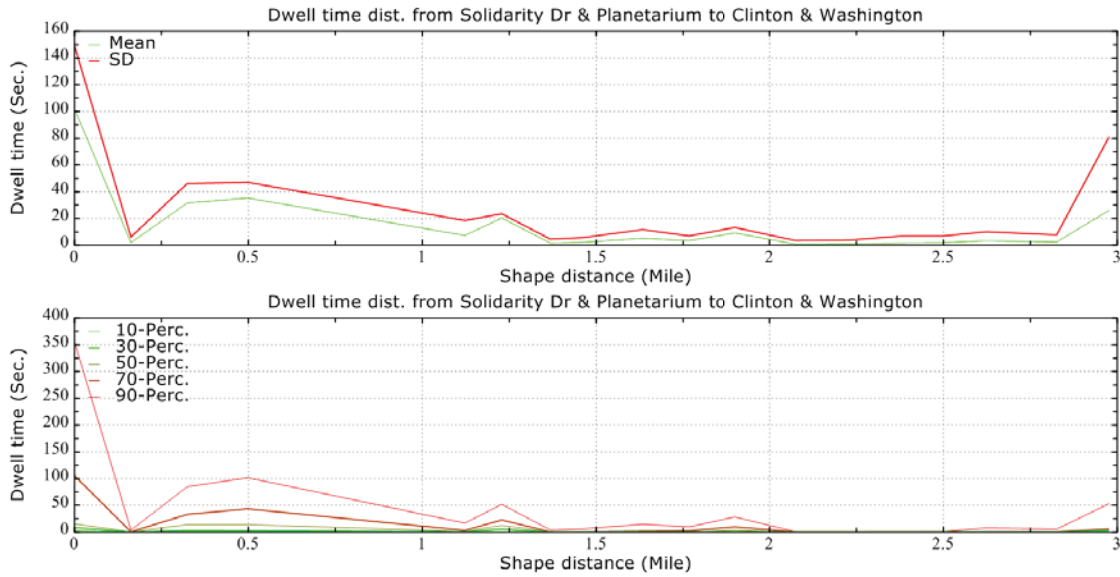


Figure 4.10: Spatial dwell time analysis for Route 130

stops while the green line is Route 125 North. It is noted that Route 125 is an express line which operates in the morning and afternoon. There is no middle day service for route 125. Figure 4.12 clearly depicts the service for Route 125. The figure is draw based on operation data on July 5th. While bus bunchings are observed in the morning (e.g. time-spatial diagram overlaps), the service is quite reliable in the afternoon.

Segment statistics are plotted in Figure 4.13. The mean travel time is 31.87 mins while the mean speed is 5.69 mph. The overall performance of the route is reliable, i.e. the PDF of travel time and speed are bell-shaped. However, some segments are irregular. Figure 4.14(a) (between Canal & Harrisan and Canal & Van Buren) and Figure 4.14(b)(Between Michigan & Ontario and Michigan & Huron) list two examples of irregular travel time and speed.

Figure 4.15 plots the bus stop headway over distance. In general, the route is quite reliable. The mean headway remains around 11 mins. The SD increases slightly over distance from 5 to 7.

Figure 4.16(a) and 4.16(b) give two examples of on-time analysis at two stops, Ohio & State and Canal & Harrison, respectively. It is observed that the distribution of Canal & Harrison is more centralized compared to Ohio & State. The observation matches our intuitive in the way that Ohio & State is Chicago downtown and thus bus headway is subject to more traffic

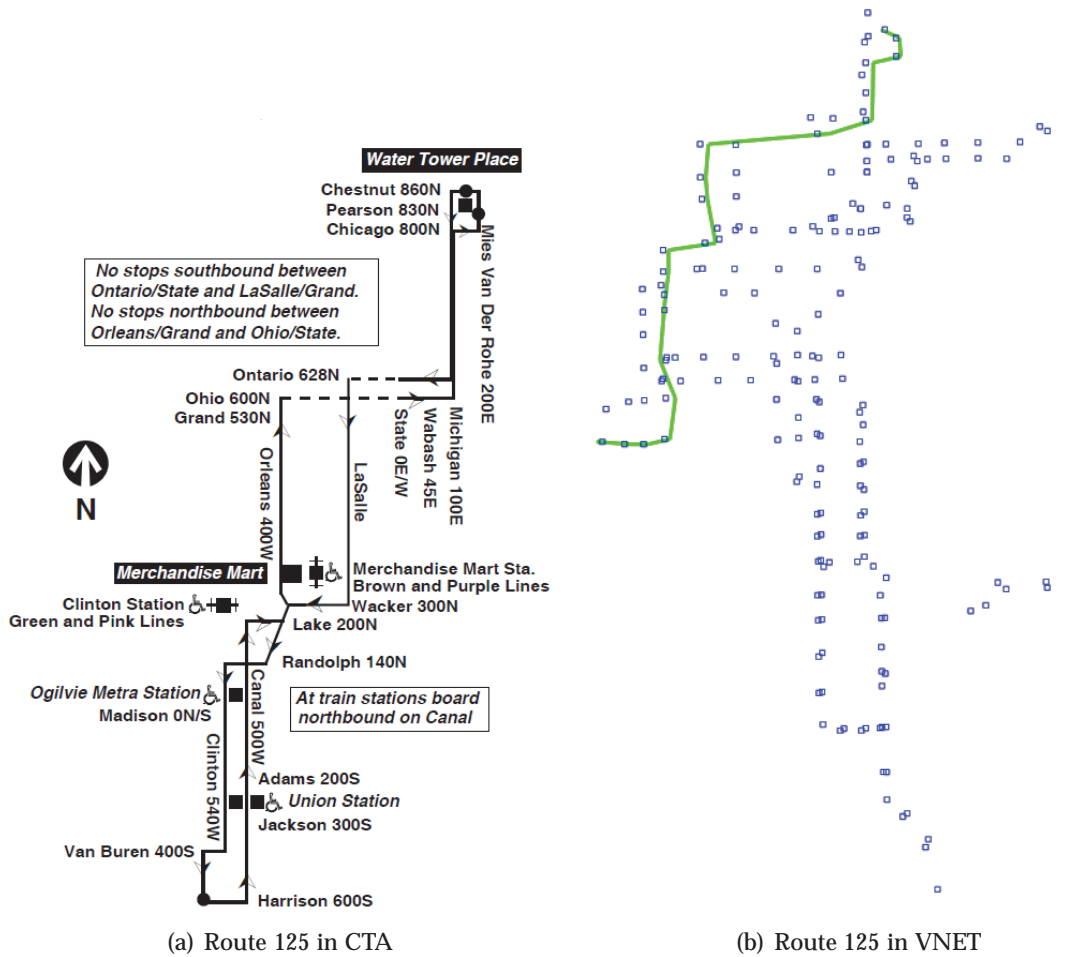


Figure 4.11: Route 125

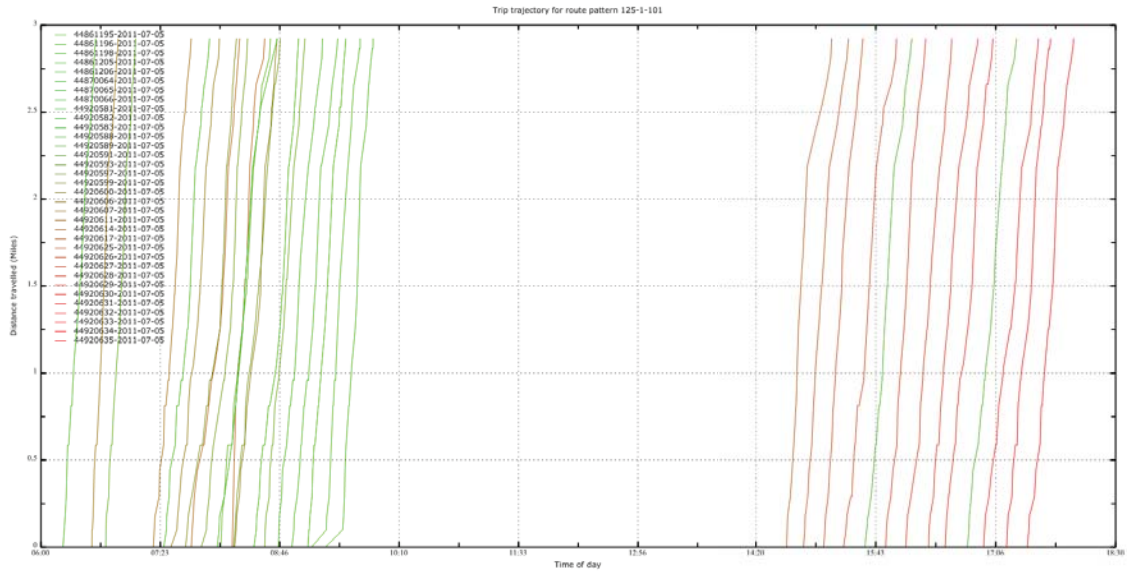


Figure 4.12: Time-spatial diagram for Route 125 North

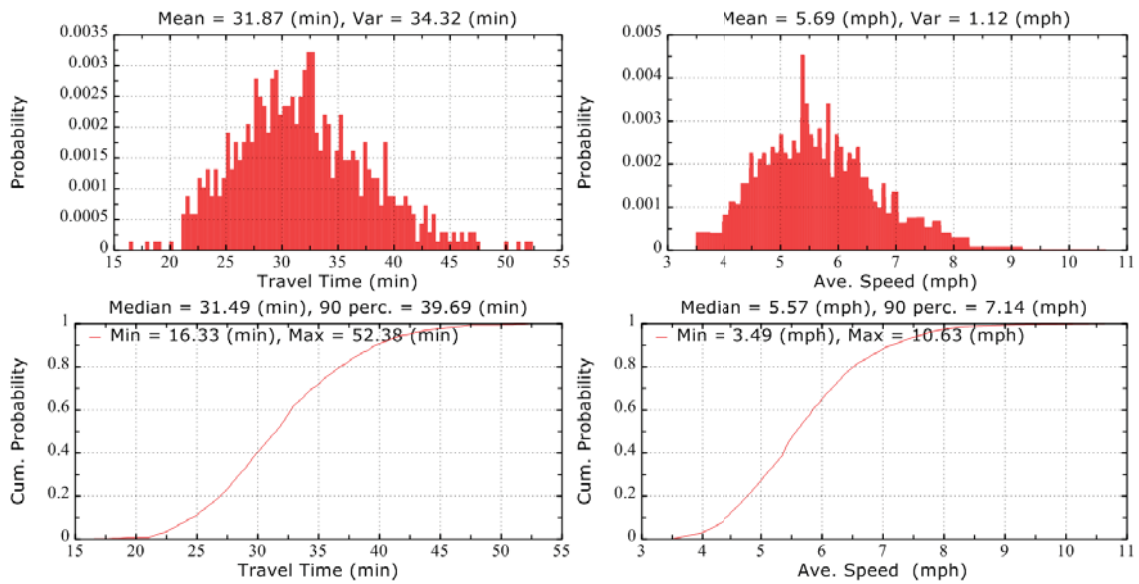


Figure 4.13: Segment statistics for Route 125

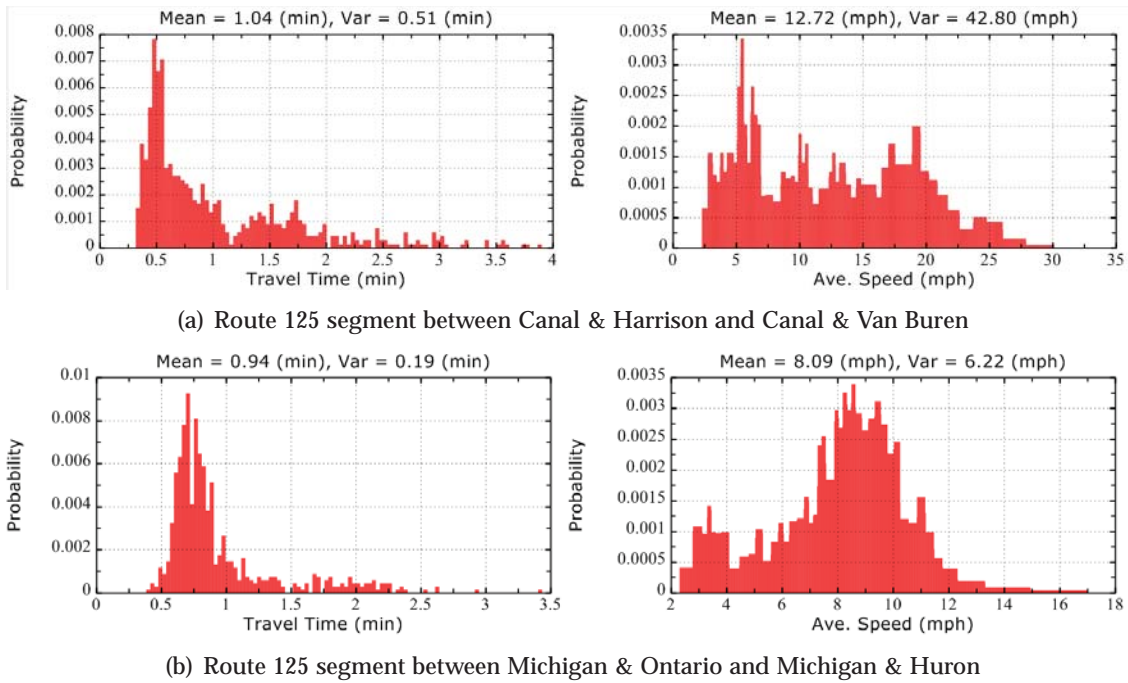


Figure 4.14: Example of irregular segments on Route 125

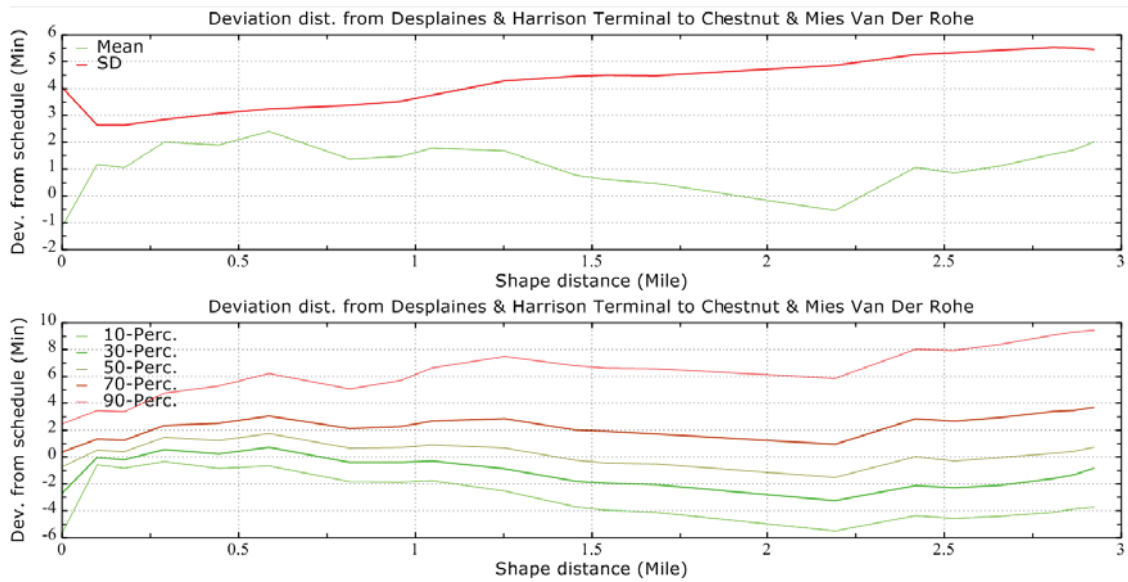


Figure 4.15: Spatial headway for Route 125

disruption in downtown. Spatial on-time analysis is provided in Figure 4.17. The mean deviation from schedule varies largely depending on the specific location. An uphill trend of standard deviation is also observed in the plot.

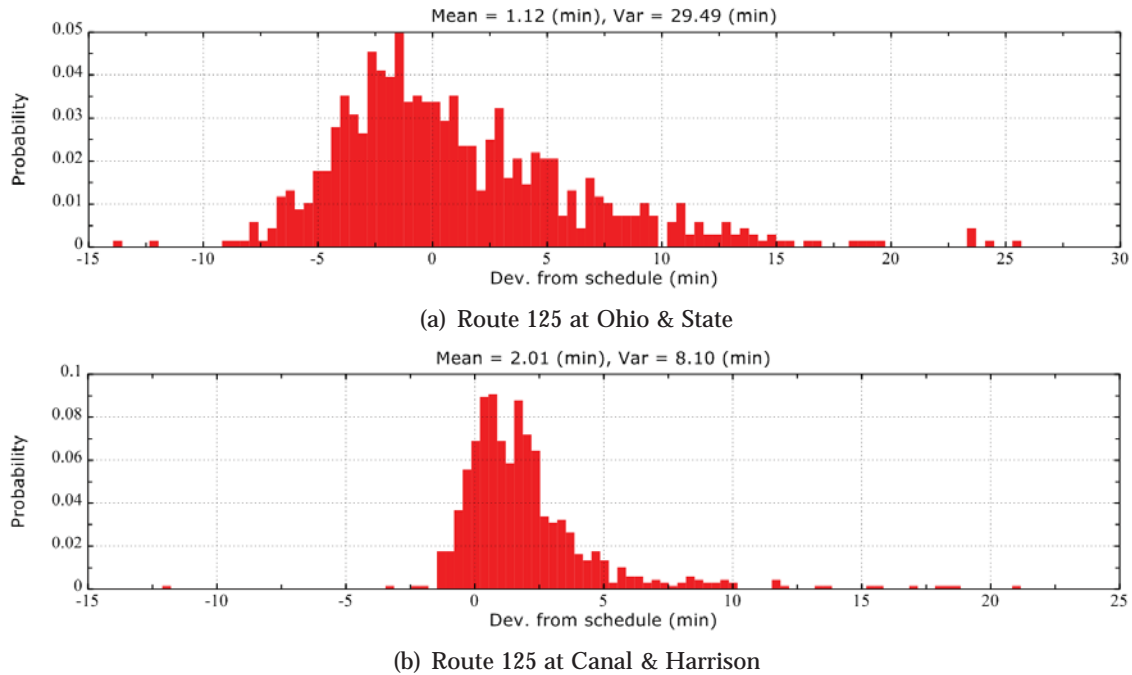


Figure 4.16: Route 125 on-time analysis

Figure 4.18(a) and 4.18(b) provides two example of dwell time analysis at Canal & Adams and Ohio & State, respectively. The mean dwell time at Canal & Adams is 41.52 secs while it is only 7.88 secs at Ohio & State. Canal & Adams is close to the Union Station. Therefore, higher passenger demand is expected which results in longer dwell time for passenger boarding/alighting.

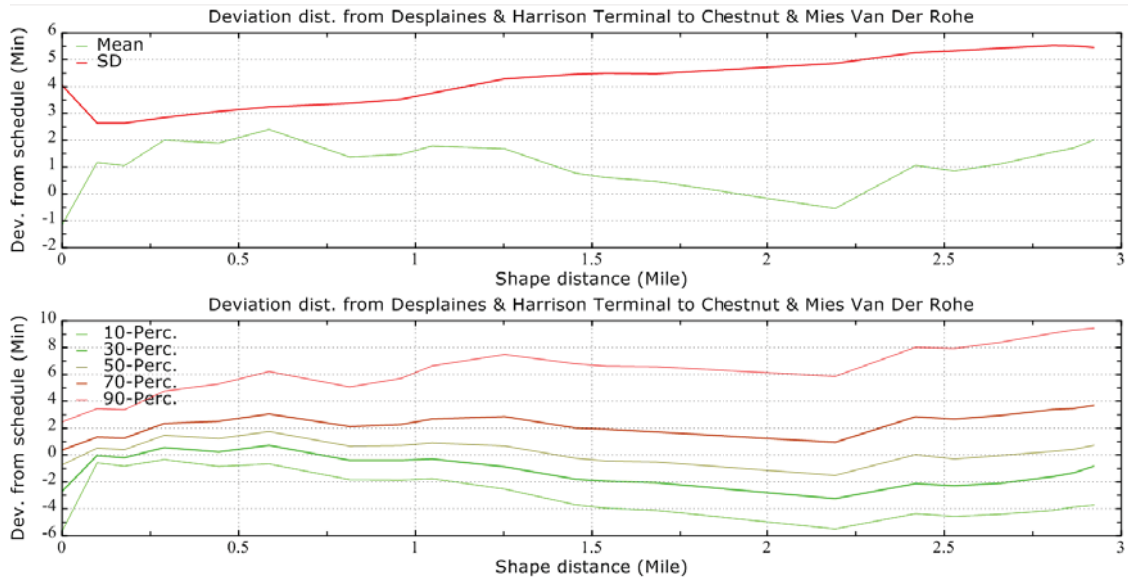
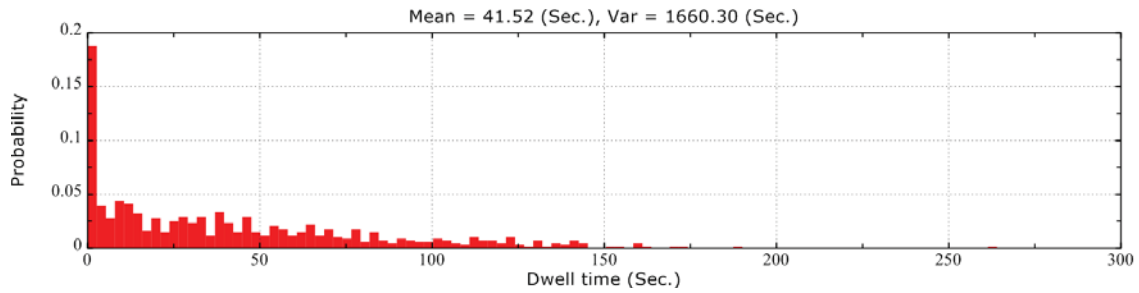
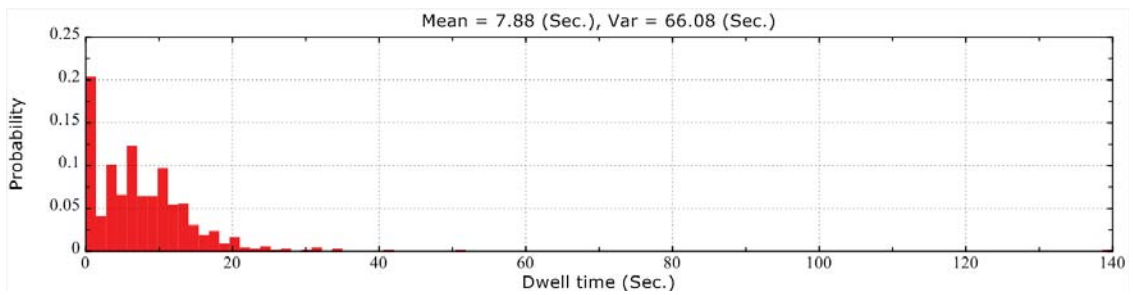


Figure 4.17: Spatial on-time analysis for Route 125



(a) Dwell time at Canal & Adams



(b) Dwell time at Ohio & State

Figure 4.18: Route 125 dwell time analysis

Chapter 5

Headway distribution fitting

In this chapter, we will fit the headway data obtained from *Transit Data Viewer* to various distributions using the distribution fitting functions provided by @Risk. @Risk¹ is a commercial software embedded into Microsoft Excel to analyze risk and uncertainty in a wide variety of industries. Since there is no underlying physical basis implying which type of distribution would suit for the bus headway distribution, we fit six commonly used distributions for positive random variables, which are Exponential, Erlang, Gamma, Weibull, LogLogistic and LogNormal distributions.

5.1 Illustration of the analysis

In this section, the Michigan & Cullerton stop on Route 1 is used to demonstrate how to perform headway distribution fitting and interpret the results.

524 valid headway observations for the stop have been extracted from the trajectories between 6 AM to 6 PM in the July of 2011. The headway sample has a mean of 14.85 minutes and a standard deviation of 6.17 minutes. Table 5.1 presents the statistic values of three popular goodness-of-fit (GOF) tests as well as the parameters of the fitted distributions. The distribution parameters are estimated using the maximum likelihood estimation (MLE) method. For each GOF test, fitted distributions are ranked according to the statistic value where the smaller the statistic value, the better the fit. Note that different GOF tests do not necessarily generate con-

¹see <http://www.palisade.com/risk/>

sistent results in terms of the rank. From the table, we see that Chi-Square test ranks Erlang as the best fitted distribution, while both Anderson-Darling (A-D) and Kolmogorov-Smirnov (K-S) tests rank LogLogistic as the best fitted distribution. It seems that there is no much difference among Erlang, Gamma and LogLogistic in terms of the statistic value. But the Exponential distribution has a much larger statistic value, which indicates it is not preferable compared with other distributions. Also worth noting is that the fitted Exponential, Erlang and Gamma distributions have the same mean value as the real data. In the real application that the mean of the headway distribution is a key influencing factor of the outcome, Erlang or Gamma distributions may suit better than the LogLogistic distribution.

Table 5.1: Statistic values of different GOF tests

Distribution	Chi-Sq	A-D	K-S	Parameters
Erlang	22.2939	1.4573	0.0451	Mean=14.85, Std. Dev.=6.06
Gamma	23.4351	1.8291	0.0465	Mean=14.85, Std. Dev.=6.18
LogLogistic	24.0496	0.9336	0.0269	Mean=15.24, Std. Dev.=7.34
LogNormal	43.8893	4.3047	0.0727	Mean=15.02, Std. Dev.=7.07
Weibull	54.5115	5.5060	0.0794	Mean=14.83, Std. Dev.=6.36
Exponential	564.1985	91.0035	0.3308	Mean=14.85, Std. Dev.=14.85

Comparing the summary statistic values given by GOF tests is one way to justify the fitted distributions, the graphical tools provided by @Risk are also helpful for visually evaluating the fit. Figure 5.1 shows the histogram of the original headway data versus the probability density function for different fitted distributions. We can see clearly that the first four distributions according to the Chi-Sq rank fit generally well, while Weibull distribution is lack of fit in the middle range of the headway value and Exponential distribution fits extremely bad in this particular case. To see where the lack of fit occurs, we can use the quantile-quantile (Q-Q) plot which is shown in Figure 5.1. If the fit is perfect, the curve should be very close to an approximate 45 degree line. Deviation from the line indicates not only the lack of fit, but also where it occurs. From the Q-Q plots, Erlang and Gamma distributions seem to fit better than LogLogistic and LogNormal distributions which are lack of fit in the right tail. The plot for Weibull distribution deviates from the 45 degree line in the middle. Exponential distribution still fits very bad.

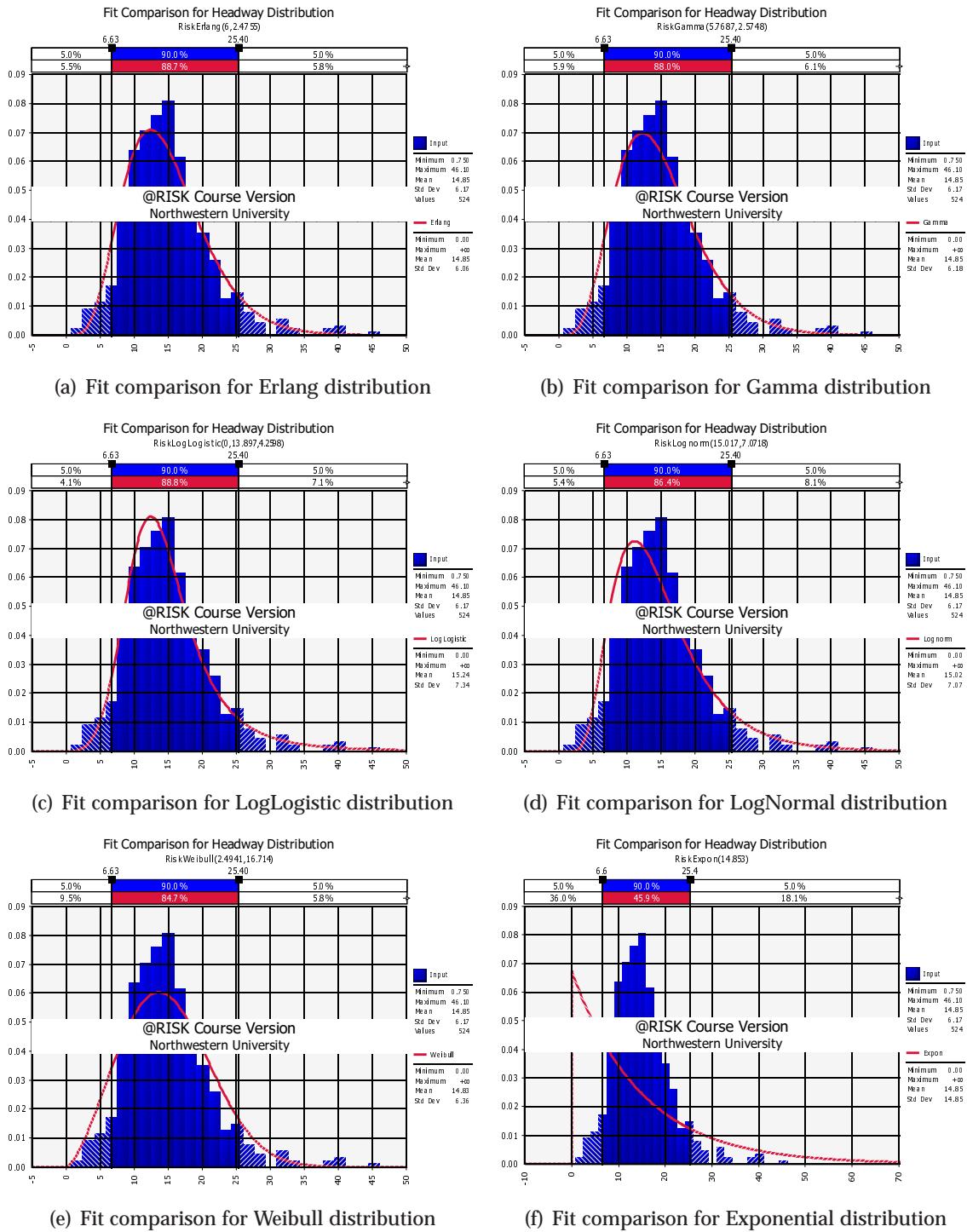


Figure 5.1: Fit comparison for different fitted headway distributions

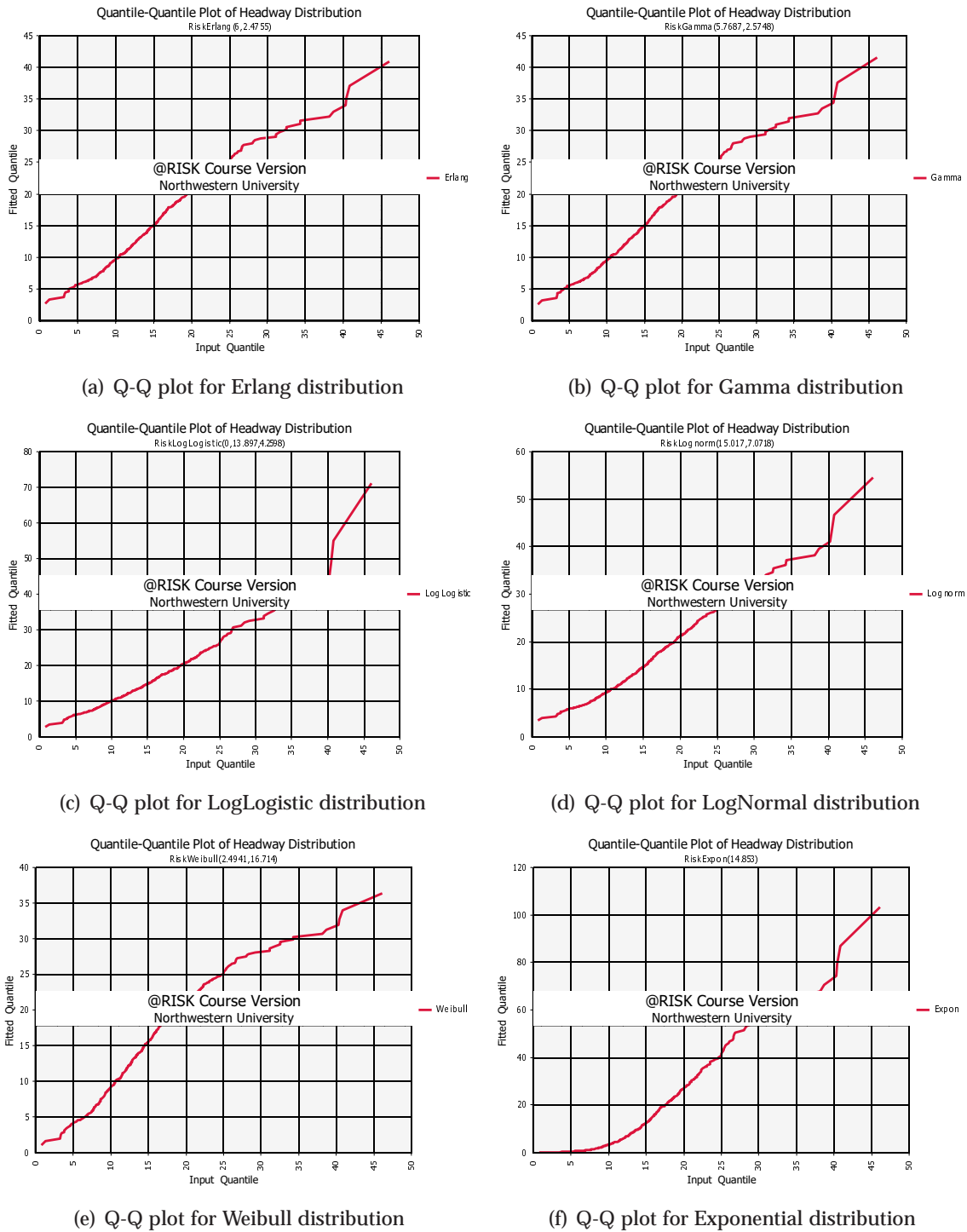


Figure 5.2: Q-Q plots for different fitted headway distributions

Now we need to fit the headway distributions for the whole CTA network, which consists of more than 10,000 stop/pattern pairs, it is not realistic to visually investigate each headway distribution individually. What we can do is automating the fitting process and obtaining the statistic values for different GOF tests, then ranking the fitted distributions accordingly. Thanks to the Excel Object Library included in @Risk, we have written a VBA program to automate the fitting process. The parameters for different fitted distributions could also be obtained for further use in executing the hyperpath algorithm. Our initial analysis suggests that a finer time resolution is needed for better routing support and more accurate performance evaluation, because CTA's bus schedules change significantly during a day. In light of this, the proposed case study will create and test the scenario in weekday morning peak hours, namely between 6:00 AM to 10:00 AM.

5.2 Headway distributions for the weekday morning peak period

We now perform a large-scale fitting using all the headway data retrieved from the trips between 6 AM and 10 AM on weekdays in the July of 2011. This data set will be used to perform the case study for transit routing in the next section.

The processed GTFS network has 125 routes, 1,577 patterns and 11,179 stops. Each pattern consists of a number of stops in a route. Thus, for each stop in a pattern, we have a headway distribution to fit. In total, we have 15,081 headway distributions, i.e., 15,081 pattern/stop pairs, to fit. In the following two subsections, we first present the summary statistics of the headway data, then report the distribution fitting results.

5.2.1 Statistics of headway data

The number of observations for each headway distribution ranges from 113 to 634 depending on how many trajectories we have for the corresponding pattern. The sample size histogram is shown in Figure 5.3(a). The sample size for all headways is larger than 100, which is adequate to qualify for the distribution fitting. Figure 5.3(b) shows the histogram of the headway

mean for all pattern/stop pairs. It shows that 11,606 out of 15,081 headway distributions (about 77.0%) have a mean headway between 10 and 20 minutes. No headway distribution have a mean headway greater than 30 minutes in weekday morning peak. The average value of the headway mean is about 13.9 minutes. The histogram of the headway standard deviation for all headway distributions is shown in Figure 5.3(c). We see that the majority of headway distributions have a standard deviation less than 10 minutes. The average headway standard deviation is about 5.1 minutes. To better demonstrate the variability of the headway distribution, we plot the coefficient of variation histogram in Figure 5.3(d), where the coefficient of variation is defined as the ratio of the standard deviation to the mean. Most of headway distributions' coefficient of variation is less than 0.6. The average value is about 0.38. Note that the coefficient of variation is 0 for deterministic distribution, and 1 for exponential distribution. The value of coefficient of variation suggests that the assumption that bus headway follows exponential distribution in most previous literature seems not valid in reality.

5.2.2 Distribution fitting results

Table 5.2 shows the number of best fits for different fitted distributions according to three GOF tests. For example, Erlang distribution is ranked best (has the smallest statistic value) by Chi-Sq test in 1,561 out of 15,081 cases. The number of failed fits for those distributions is also reported. Failure of fitting the distribution means the MLE method cannot converge to get the parameters of the fitting distribution for some reason, for example, Erlang distribution fails to fit the data in 6 out of 15,081 cases. From the results of Table 5.2, LogLogistic distribution seems to fit best in general among the six distributions. However, having the smallest statistic value is not equivalent to that the distribution is the best fit in all aspects. Remember that in the example we show before, the LogLogistic distribution has the smallest statistic value for both A-D and K-S tests, but the Q-Q plot shows that Erlang and Gamma distributions fit better. Besides, the statistic value of Erlang or Gamma distribution may be only slightly larger than that

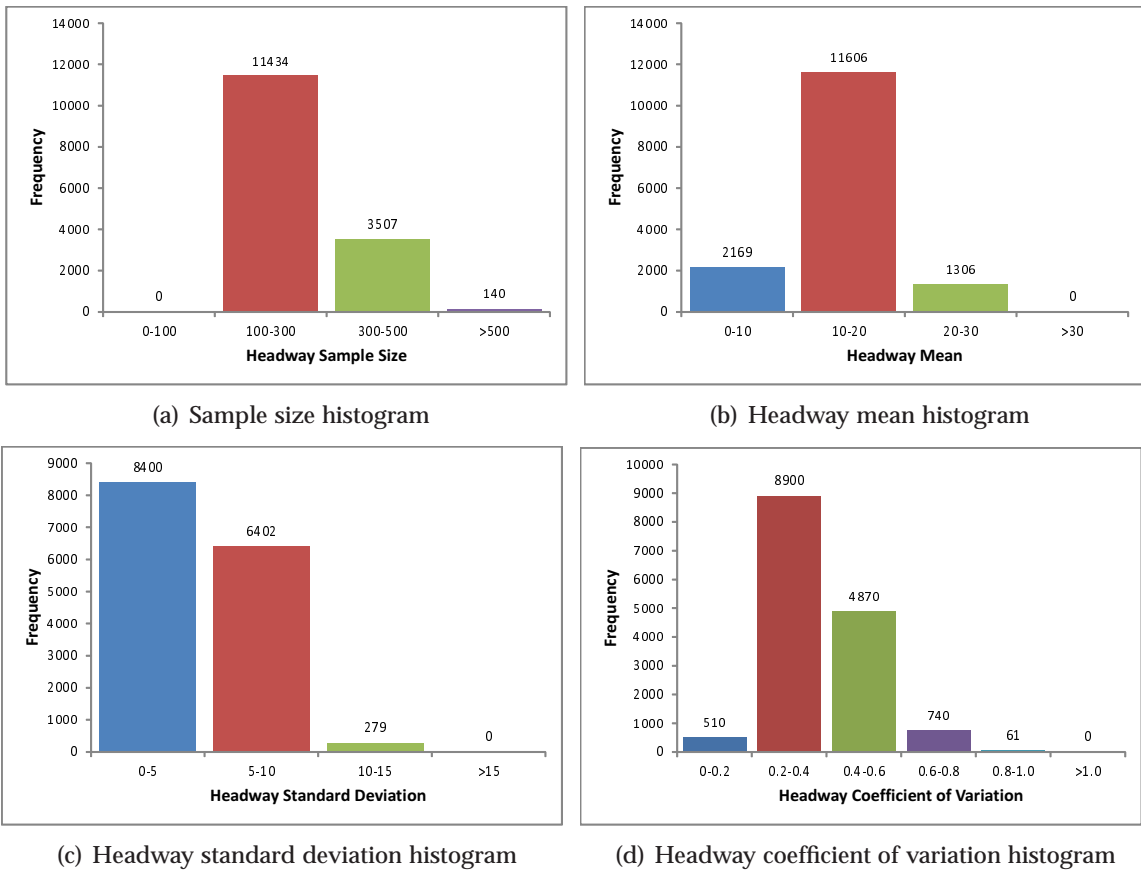


Figure 5.3: Summary statistics of headway observations in weekday morning peak

of LogLogistic distribution. In order to see how much the other fitted distributions deviate from the best fitted one, we calculate the percentage deviation of the statistic value from best (i.e., the smallest statistic value) for all six distributions and for each fit. The average percentage deviation across all successful fits are reported in Table 5.3. From this perspective, LogLogistic is still the best fit overall. Erlang and Gamma are the second best fits. Also worth emphasizing is that fitted Exponential, Erlang and Gamma distributions have the same mean headway as the real data. The fact that Gamma actually includes Erlang and Exponential distributions suggests that we should prefer Gamma to Erlang and Exponential. The reason we may be in favor of Erlang distribution is that it provides more flexibility than Exponential distribution (commonly used in most previous literature) and is more computationally tractable than Gamma distribution as we will see later.

Table 5.2: Number of best fits for various distributions

Distributon	Chi-Sq	A-D	K-S	Number Of Failed Fits
Erlang	1561	605	503	6
Gamma	1350	545	466	6
LogLogistic	6807	9193	10394	215
LogNormal	1355	529	513	0
Weibull	4007	4209	3205	0
Exponential	1	0	20	0

Table 5.3: Average percentage deviation of the statistic value from best

Distribution	Chi-Sq (%)	A-D (%)	K-S (%)
Weibull	122.63	399.94	77.86
Gamma	53.15	143.4	52.22
Erlang	54.17	149.85	53.08
Expon	1353.42	4481.54	496.82
LogLogistic	32.5	46.3	7.19
LogNormal	104.2	281.26	77.4

5.3 Summary

Both the summary statistics of the headway observations and headway distribution fitting results show that headway distribution is unlikely to be exponential in reality. The LogLogistic

distribution is the best fitted distribution in general for CTA's bus headway, followed by Erlang and Gamma distributions. Some other aspects also need to be considered when choosing the headway distribution type for the real application. For instance, fitted Gamma and Erlang distributions have the same mean as the real data. Besides, computational tractability is another important factor, especially for the large-size problem. Erlang distribution has a significant advantage over Gamma and LogLogistic distributions in terms of computational efficiency, because the waiting time distribution can be computed in closed-form when the headway follows Erlang headway (see Section 3.2). Taking all of these into account, Erlang is an overall good choice for our large-scale case study, which balances between the accurate representation of the headway distributions and the reasonable computational efforts.

Chapter 6

Numerical experiments

6.1 Impacts of service regularity on route choice

In most previous studies, the transit headway distribution is assumed to be exponentially distributed, which implies highly irregular transit service. However, as the CTA headway data revealed, this is hardly the case in the real world. To illustrate the impact of the service regularity on the route choice, we first analyze a small example with two lines, whose attributes are shown in Table 6.1. The two lines are assumed to have Erlang headway distributions. For simplicity, both lines have the same Erlang shape parameter k , i.e., the standard deviation of the headway distribution is $E[h]/\sqrt{k}$. So as k increases, the standard deviation decreases, which means the service is more regular. Note that the Erlang distribution with shape parameter $k = 1$ is equivalent to the exponential distribution with the same mean, and the Erlang distribution with shape parameter $k = +\infty$ is equivalent to the deterministic distribution.

The results of the route choice probability and expected travel time for different levels of service regularity are reported in Table 6.2. Some interesting trends can be observed from the results. First, as k increases, i.e., the service becomes more regular, the expected travel time keeps going down to the lower bound which corresponds to the deterministic headway distribution. Second, the probability of choosing the first line, i.e., the faster line, at first decreases as k increases, and then jumps to 1 as k reaches to a breakpoint, which means the second line, i.e., the slower line, is excluded from the attractive set. An explanation is that the passengers can take ad-

vantage of the higher frequency of the slower line when the service is not regular enough. When the service becomes very regular, the passengers will stick to the faster line since the decreased line travel time in this example outweighs the increased waiting time of doing so.

Table 6.1: Attributes of two transit lines

Line i	Headway Mean $E[h_i]$ (min)	Travel Time s_i (min)
1	20	20
2	10	30

Table 6.2: Route choice results for various service regularity levels

k	π_1	π_2	$E[W]$	$E[S]$	$E[T]$
1	0.333	0.667	6.67	26.66	33.33
2	0.315	0.685	5.37	26.85	32.22
3	0.303	0.697	4.96	26.97	31.93
4	0.295	0.705	4.76	27.05	31.81
10	1	0	11.00	20.00	31.00
50	1	0	10.20	20.00	30.20
∞	1	0	10.00	20.00	30.00

6.2 Determination of the attractive set

It is well known that for the exponential headway, the attractive set could be determined using a greedy method, which ranks all the available lines in the increasing order of the line travel time and successively adds the line into the attractive set until the line travel time of the next line exceeds the current expected travel time (Nguyen 1989). However, this greedy method may not produce the right attractive set for other headway distributions, see Gentile et al. (2005) for example. Gentile et al. (2005) also proposed another greedy method which rearranges the order of the lines according to the expected travel time of the line considered separately and stop as soon as the addition of the next line increases the expected travel time. Gentile et al. (2005) reported that they were neither able to prove that the correctness of the proposed strategy, nor able to find a counterexample. In this following, we will show this strategy does not guarantee the correct identification of the attractive set, using a counterexample.

Consider the example given in Table 6.3, the headway distributions are assumed to be exponential for all lines. The route choice probability and expected travel time of all possible combinations of the three lines are reported in Table 6.4. From the results, we can see that the line order should be $\{1, 3, 2\}$ according to the expected travel time of each line. Consequently, the resulting attractive set by executing the greedy method is $\{1, 2, 3\}$, while the right attractive set is $\{1, 2\}$.

Table 6.3: Line Attributes

Line i	Headway Mean $E[h_i]$ (min)	Travel Time s_i (min)
1	20	30
2	15	40
3	10	44.8

Table 6.4: Numerical results for various combinations of lines

Line Set	π_1	π_2	π_3	EW	ER	ET
1	1	0	0	20.00	30.00	50.00
2	0	1	0	15.00	40.00	55.00
3	0	0	1	10.00	44.80	54.80
1 2	0.429	0.571	0	8.57	35.72	44.29
1 3	0.333	0	0.667	6.67	39.86	46.53
2 3	0	0.400	0.600	6.00	42.88	48.88
1 2 3	0.231	0.308	0.461	4.62	39.90	44.52

Therefore, we conclude that no greedy method is exact in the case of general headway distribution. For exponentially distributed headway, the original greedy method (based on ranking line travel time) can ensure the right attractive set. For all other headway distributions, either greedy method may be used as an approximation method. We note also that the enumeration method may be considered as an alternative when the number of lines is small.

6.3 Simulation results of route choice

As discussed before, the waiting time distribution for a specific line is an asymptotical distribution when the bus operating time approaches infinity. Therefore, the resulting route choice probability is only valid at the steady state. In this section, we use a discrete event simulation to

verify the analytical results at the steady state. The simulation is implemented in Microsoft Excel using VBA, see the interface in Figure 6.1. The user can specify the running length, the number of replications, the number of transit lines at the stop as well as the attributes of each line, e.g., headway mean and headway standard deviation. The results of the route choice probability and the trip performance measures will show automatically in the results region after the simulation is completed. A 95% confidence interval of each measure is also calculated as well as the relative error, which is the half width of the confidence interval divided by the midpoint of the confidence interval.

Transit Route Choice Simulation								
Num of Lines:		3	Run Length (min):		10000	Num of Replications:		100
Parameters						Results		
Line	Headway Mean	Headway Var	Line Travel Time	Distribution Type	Info. Aval.	Probability	Conf. Interval	Rel. Error
1	20	44.44444444	10	Erlang	No	0.203473144	0.000515405	0.002533
2	15	25	15	Erlang	No	0.291439529	0.000210117	0.000721
3	10	11.11111111	16	Erlang	No	0.505087327	0.000409274	0.0008103

Figure 6.1: Interface of the route choice simulation

In the simulation, each line is dispatched according to the headway distribution specified by the user. Passengers' arrival at the stop is modeled as a Poisson arrival. The passengers will board the first coming line. The probability of boarding each line is calculated as the number of passengers who board that line divided by the total number of passengers arriving at the stop. We set the running length to be 100,000 minutes in each replication (unless otherwise specified) to estimate the route choice probability at the steady state. The number of replications is set as 100 to ensure that the relative error is less than 5%. Note that the passengers do not calculate the attractive set in the simulation, they simply board the first coming line. In order to compare the simulation results with the analytical results, we choose a three-line example in which all of the three line are included in the attractive set. The line attributes are shown in Table 6.5.

Table 6.6 and Table 6.7 report both simulation results and analytical results for the Exponential headway case and Erlang headway case, respectively. For Erlang headway case, all three lines

Table 6.5: Line attributes

Line i	Headway Mean (min)	Travel Time (min)
1	20	10
2	15	15
3	10	16

are assumed to have the same shape parameter $k = 9$, which means the standard deviation is one third of the headway mean. The 95% confidence interval is also reported. The relative difference is the percentage difference of the analytical result from the simulation result. The results show that the analytical probabilities at the steady state are very accurate for both the Exponential headway case and the Erlang headway case. The results for the Deterministic headway case are presented in Table 6.8. The relative differences are much larger than those of the Exponential and Erlang headway cases. Since the headway is deterministic, with the headway values in Table 6.5, multiple lines may arrive at the stop at the same time, e.g., both line 1 and line 3 will arrive at the stop at time $t = 20$ minutes. If there are more than one line arriving at the stop at the same time, the waiting passengers will be evenly allocated to the arriving lines. If we change line 1's headway mean to 20.01 or 19.99 minutes, then line 1 and line 3 will rarely arrive at the same time within the 100,000 minutes interval. We also report the results for these two cases in Table 6.9 (line 1' headway = 20.01 minutes, case 2) and Table 6.9 (line 1' headway = 19.99 minutes, case 3), respectively. Two surprising facts are worth pointing here: (1) The results change dramatically from case 1 for both case 2 and case 3, and they are closer to the analytical results; (2) Case 2 and case 3 have very close results.

Table 6.6: The Exponential headway case

	Simulation Result	Analytic Result	Relative Difference (%)
Line 1 Probability	0.231 ± 0.001	0.231	0
Line 2 Probability	0.308 ± 0.001	0.308	0
Line 3 Probability	0.461 ± 0.001	0.461	0
Expected Waiting Time	4.61 ± 0.01	4.62	0.2
Expected Trip Time	18.92 ± 0.01	18.92	0

Regarding the first fact, the reason why the analytical formula is not accurate for case 1 is

Table 6.7: The Erlang headway case

	Simulation Result	Analytic Result	Relative Difference (%)
Line 1 Probability	0.203±0.001	0.203	0
Line 2 Probability	0.290±0.001	0.290	0
Line 3 Probability	0.507±0.001	0.507	0
Expected Waiting Time	3.43±0.01	3.43	0
Expected Trip Time	17.92±0.01	17.92	0

that it does not consider the situation in which multiple lines arrive at the stop at the same. Simply speaking, the assumptions behind the analytical formula do not state which line to board if multiple arrivals at the same time occurs. In our simulation, we assume that each line will get the equal share of the waiting passengers. However, the best strategy for passengers should be boarding the line with the minimum line travel time among those arriving lines. If we enforce this rule in the simulation, we can expect that the resulting route choice probability will be different for case 1. Another interesting point regarding the first fact is that the expected waiting time in case 1 is much larger than that in both case 2 and case 3. This is not hard to understand. Since in both case 2 and case 3, line 1 and line 3 rarely arrives at the same time, the probability of having at least one line arrival at a given period increases. Consequently, passengers' expected waiting time decreases.

Table 6.8: The Deterministic headway case 1

	Simulation Result	Analytic Result	Relative Difference (%)
Line 1 Probability	0.180±0.001	0.194	7.8
Line 2 Probability	0.306±0.001	0.278	-9.1
Line 3 Probability	0.514±0.001	0.528	2.7
Expected Waiting Time	4.17±0.01	3.33	-20.1
Expected Trip Time	18.78±0.01	17.89	-4.7

The second fact that case 2 and case 3 have almost the same results seems counter-intuitive at first glance. For case 2, because of the 20.01 minutes headway, a line 1 will arrive a little bit later than a line 3 at first, so the line 1 will get very few passengers since most waiting passengers already board line 3. Thus, we should expect the probability of boarding line 1 decreases in case 2 and increases in case 3, compared with case 1. But if the running length is long enough, this effect

Table 6.9: The Deterministic headway case 2

	Simulation Result	Analytic Result	Relative Difference (%)
Line 1 Probability	0.209±0.001	0.194	-7.2
Line 2 Probability	0.271±0.001	0.278	2.6
Line 3 Probability	0.520±0.001	0.528	1.5
Expected Waiting Time	3.54±0.01	3.33	-5.9
Expected Trip Time	18.02±0.01	17.89	-0.7

Table 6.10: The Deterministic headway case 3

	Simulation Result	Analytic Result	Relative Difference (%)
Line 1 Probability	0.208+0.001	0.194	-6.7
Line 2 Probability	0.271+0.001	0.278	2.6
Line 3 Probability	0.521+0.001	0.528	1.3
Expected Waiting Time	3.54+0.01	3.33	-5.9
Expected Trip Time	18.02+0.01	17.89	-0.7

may be eliminated as the inter arrival time between a line 1 and a line 2 changes periodically. To confirm this, we report the results for both case 2 and case 3 with different running length in Table 6.11 and Table 6.12. As we can see, with running length 10,000 minutes, the probability of taking line 1 in case 2 is smaller than that in case 3. As the running length increases to 100,000 minutes, the probability of taking line 1 is almost the same in both case 2 and case 3.

Even though the simulation results in both case 2 and case 3 are closer to the analytical results compared with case 1, the relative difference is still larger if compared with the Expon or Erlang distribution case. Also noticing that the simulated expected trip time for deterministic distribution is a little larger than that for the Erlang distribution. It indicates that the deterministic headway dispatching does not necessarily guarantee the minimum expected trip time if the headways for each line are badly chosen.

Table 6.11: The Deterministic headway case 2 with different running length

Running Length (mins)	10,000	100,000	1,000,000
Line 1 Probability	0.125+0.001	0.209+0.001	0.2079+0.0001
Line 2 Probability	0.271+0.001	0.271+0.001	0.2711+0.0001
Line 3 Probability	0.604+0.001	0.520+0.001	0.5210+0.0001
Expected Waiting Time	3.54+0.01	3.54+0.01	3.542+0.001
Expected Trip Time	18.52+0.01	18.02+0.01	18.023+0.001

Table 6.12: The Deterministic headway case 3 with different running length

Running Length (mins)	10,000	100,000	1,000,000
Line 1 Probability	0.291+0.001	0.208+0.001	0.2088+0.0001
Line 2 Probability	0.270+0.001	0.271+0.001	0.2707+0.0001
Line 3 Probability	0.439+0.001	0.521+0.001	0.5205+0.0002
Expected Waiting Time	3.54+0.01	3.54+0.01	3.542+0.001
Expected Trip Time	17.53+0.01	18.02+0.01	18.018+0.001

6.4 An Illustrative hyperpath routing example

In this section, we present a small hypothetical network to illustrate the impact of different headway assumptions on the resulting optimal hyperpaths. The example transit network consists

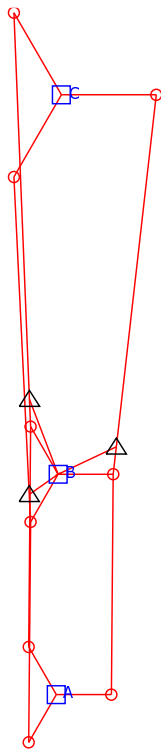


Figure 6.2: A three-stop transit network

of three transit lines which go through the three same stops, namely A, B, and C, but with different attributes on line headway and line travel time. The network topology of the small example, represented by the NETGTFS format, is shown in Figure 6.2, in which the squares represent the transfer stop nodes A, B, and C; the circles represent the transit nodes; the triangles represent the dwell nodes. The line attributes are shown in Table 6.13. To compare the impacts of different headway assumptions, we test the Exponential headway and Erlang headway on this small network. For simplicity, the shape parameter of Erlang headway distribution for all line-stops is set as 9. This implies that the coefficient of variation for Erlang headway distribution is about 0.333, which is very realistic according to our analysis in the previous section. In the test, stop C is the destination, passengers can travel

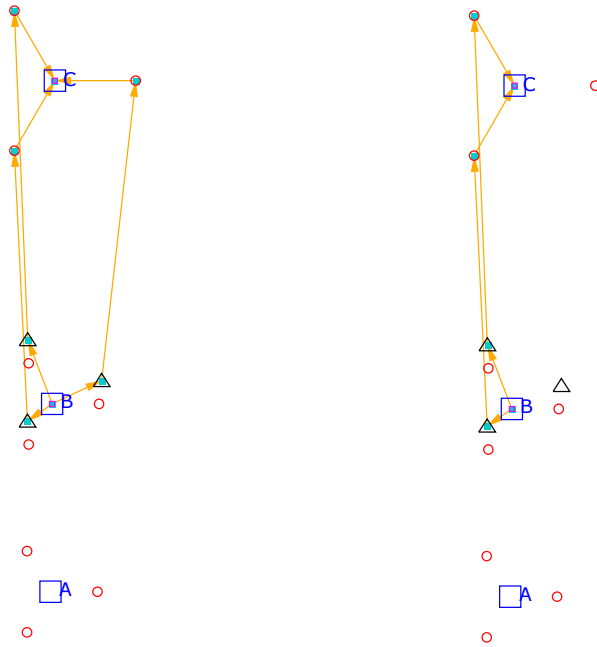
from both stop A and stop B to the destination.

Not to our surprise, our tests generate different shortest hyperpaths for the Exponential and Erlang headway distributions. The results are shown in Figure 6.4. At stop B, the expected total travel time for Exponential headway distribution is 34.67 minutes compared with 31.64 minutes for Erlang headway distribution. Also notice that the shortest hyperpath for Exponential headway distribution includes all three lines while it only includes line 2 and line 3 for Erlang headway distribution. This is because the expected total travel time of the set of line 2 and line 3 decreases significantly as the headway variance decreases, thus, it is not advantageous to take the line 3 with a much longer line travel time. The intuition here is that since the headway is more regular, the shortest hyperpath may exclude the line with longer travel time. At stop A, the expected total travel time for Exponential headway distribution is 68.57 minutes compared with 64.82 minutes for Erlang headway distribution. Passengers may board either line 1 or line 2, whichever comes first, for both Exponential headway distribution and Erlang headway distribution. The interesting thing is that with Erlang headway distribution, if passengers board line 1 at stop A, they will get off line 1 at stop B and wait for either line 2 or line 3, whichever comes first. This strategy will yield a smaller expected travel time.

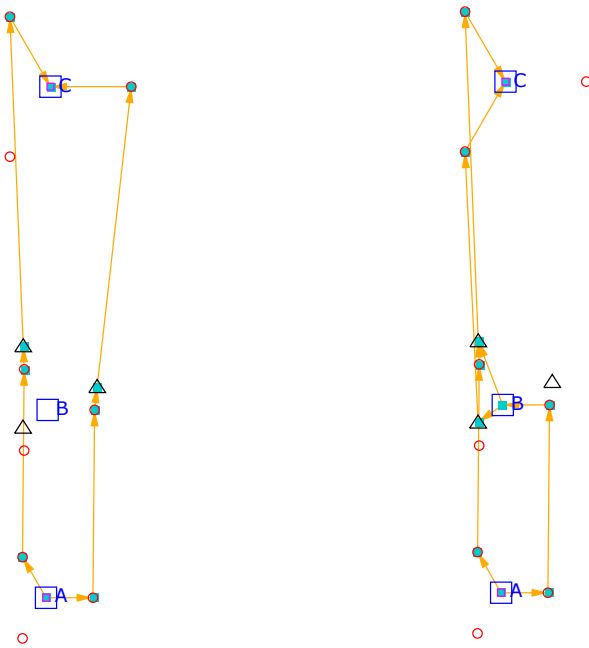
The numerical results with the small network indicate that under different headway assumptions, we could predict quite different shortest hyperpaths, thus quite different assignment results if we implement the transit assignment algorithm. As the network size increases, the difference may be much more significant. In the next section, we will conduct a few tests on the CTA network with the real operation data.

Table 6.13: Line attributes of the hypothetical network

Line i	Headway Mean at Stop A	Travel Time From A to B	Headway Mean at Stop B	Travel Time From B to C
1	10	30	20	34
2	25	30	30	25
3	10	50	15	25



(a) Shortest hyperpath at Stop B for Exponential distribution (b) Shortest hyperpath at Stop B for Erlang distribution



(c) Shortest hyperpath at Stop A for Exponential distribution (d) Shortest hyperpath at Stop A for Erlang distribution

Figure 6.3: Shortest hyperpath under different headway assumptions

Chapter 7

Case study of the CTA bus network

In this section, we conduct experiments on the CTA bus network with the real operation data to examine the performance of the CTA bus service under different bus headway assumptions, i.e., Exponential and Erlang. Both Exponential and Erlang headway distributions are calibrated from the real headway observations. The bus travel time from one stop to another is constant, which is the average of the real travel time observations. Note that we exclude the train and metra routes from the network because we do not have the operation data of those two modes.

7.1 Illustrative example

We use one O-D pair to illustrate that different bus headway assumptions may result in very different strategies of choosing attractive bus routes. In this case, the origin is Ashland and Irving Park, the destination is Michigan and Grand. The headway observations used to fit the distributions are collected between 6:00 AM and 10:00 PM across the July of 2011. The shortest hyperpaths from Ashland and Irving Park to Michigan and Grand for both headway assumptions are shown in Figure 7.1. As we can see, the shortest hyperpath for Erlang headway distribution happens to be a simple path in this case while the shortest hyperpath for Exponential headway distribution is much more complicated. It indicates that some bus routes are excluded from the attractive set when headways are considered as following Erlang distribution, which is consistent with our analysis for the small three-stop example. More details of these two hyperpaths are

reported in Table 7.1. The expected total trip time for Erlang case is -15.03% less than that for Exponential case. The enroute time is actually longer for Erlang case. The waiting time for Erlang case is slightly larger because in this instance the passengers have only one bus to wait at each stop for Erlang case. The increased waiting time for Erlang case is more than offset by the significantly decreased walking time. The results in this example suggest that Erlang headway distribution can generate better overall performance than Exponential headway distribution.

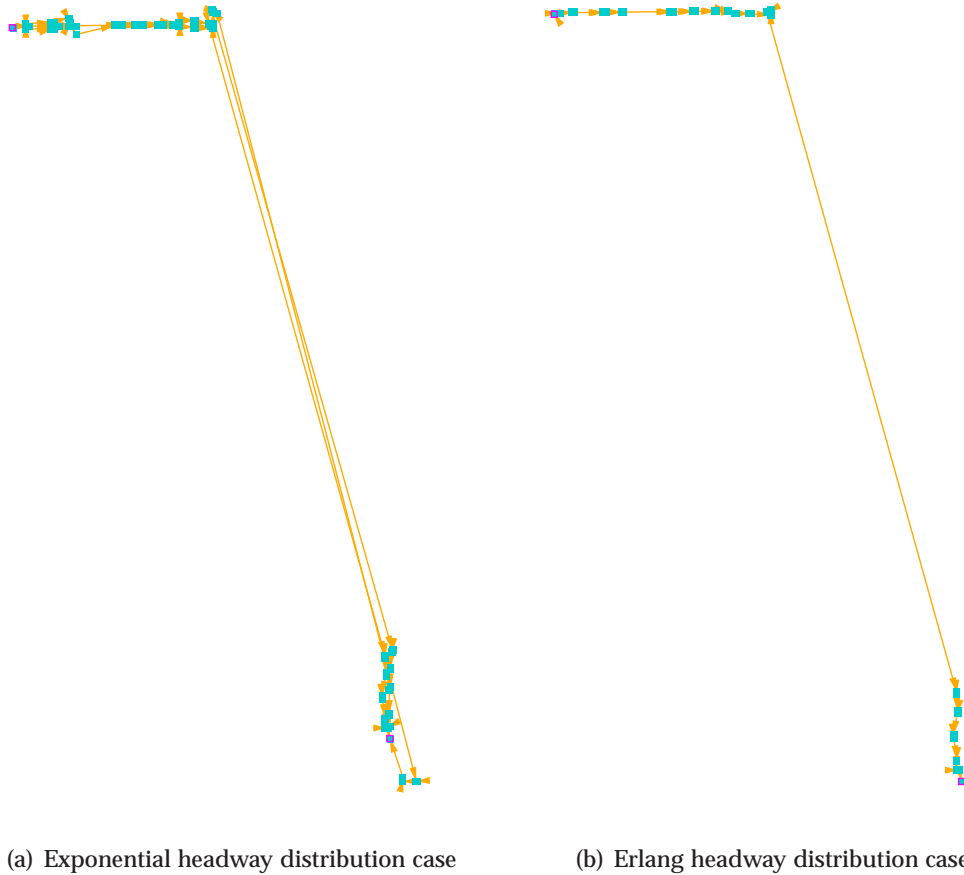


Figure 7.1: Shortest hyperpaths from Ashland and Irving Park to Michigan and Grand

7.2 Average performance in morning peak

In this section, we investigate the average performance of the CTA bus network in the morning peak hours, i.e., between 6:00 AM and 10:00 AM. For this purpose, four representative zones

Table 7.1: Trip performance from Ashland and Irving Park to Michigan and Grand

Expected Performance	Exponential Case	Erlang Case	Difference (%)
TotalTripTime (min)	43.50	36.96	-15.03
EnRouteTime (min)	18.32	22.22	21.29
WaitingTime (min)	11.24	12.93	15.04
WalkingTime (min)	13.49	1.41	-89.55
TransferLostTime (min)	0.45	0.40	-11.11
TotalTripDistance (mile)	6.42	5.83	-9.19
AverageSpeed (mph)	8.85	9.46	6.89

are selected in the greater Chicago area, namely Downtown Chicago, North suburbs, South Suburbs and West Suburbs, as shown in Figure 7.2. In each zone, ten random places are selected as the origins or destinations. Four scenarios are tested for both Exponential and Erlang headway distributions, which are North Suburbs to Downtown Chicago, South Suburbs to Downtown Chicago, West Suburbs to Downtown Chicago and Downtown Chicago to Downtown Chicago. In each scenario, passengers travel from all the origins to all the destinations. Therefore, there are in total 90 trips in Downtown Chicago to Downtown Chicago scenario, and 100 trips for each of the three other scenarios. The average performance of all four scenarios for both headway distribution are reported in Table 7.2–7.5. Note that all performance measures are the averages of expected values across all trips in each scenario. For Erlang headway distribution, the greedy method based on the line travel time is used to generate the attractive set.

From Table 7.5 in Downtown Chicago to Downtown Chicago scenario, the total trip time for Erlang case is only slightly shorter than that for Exponential case. The walking time is more than twice as much as the enroute time. The waiting time at a stop is very low compared with the other scenarios. The average speed is comparable with the walking speed due to short enroute time. For the three other scenarios, the Erlang headway distribution yields better overall performance. The enroute time for both distributions cases are approximately the same while the Erlang case has much shorter waiting and walking time. The average speed seems to be increasing with the trip distance because there will be more enroute time involved in the trip.

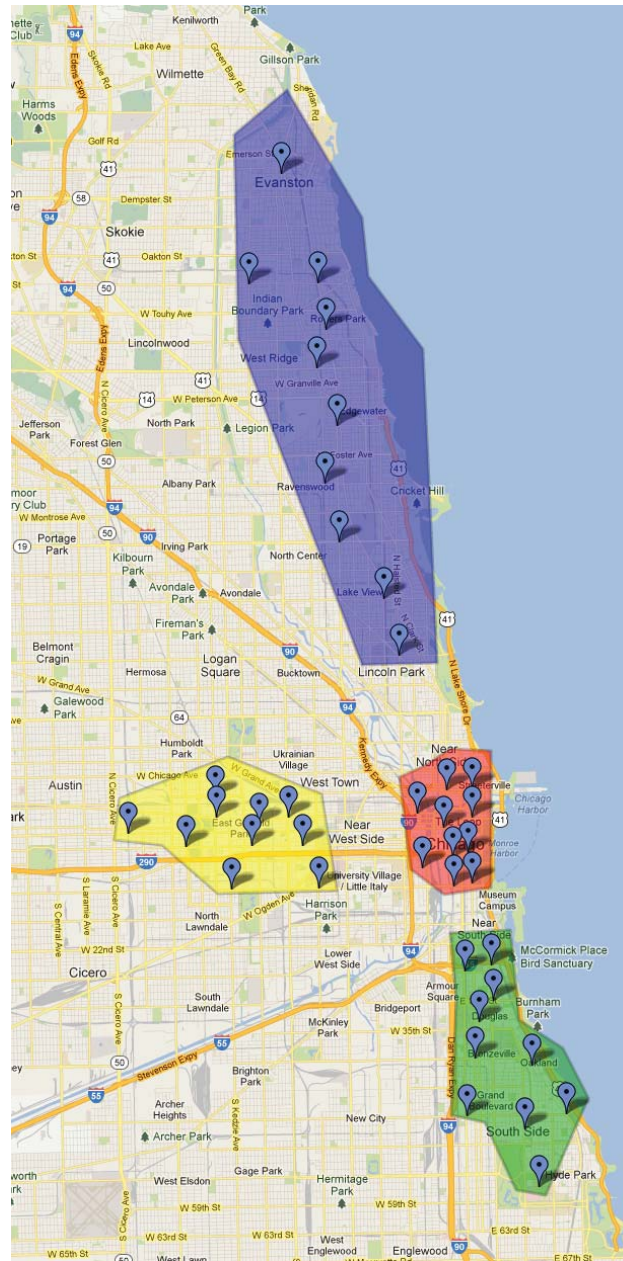


Figure 7.2: Four representative zones in the greater Chicago area

Table 7.2: Average performance in North Suburbs to Downtown Chicago scenario

Avg. Expected Performance	Exponential Case	Erlang Case	Difference (%)
TotalTripTime (min)	66.32	60.86	-8.23
EnRouteTime (min)	38.20	40.09	4.95
WaitingTime (min)	13.28	10.89	-18.00
WalkingTime (min)	14.39	9.43	-34.47
TransferLostTime (min)	0.45	0.45	0.00
TotalTripDistance (mile)	9.28	9.19	-0.97
AverageSpeed (mph)	8.38	8.98	7.16

Table 7.3: Average performance in South Suburbs to Downtown Chicago scenario

Avg. Expected Performance	Exponential Case	Erlang Case	Difference (%)
TotalTripTime (min)	38.65	36.42	-5.77
EnRouteTime (min)	22.37	22.58	0.94
WaitingTime (min)	7.43	6.33	-14.80
WalkingTime (min)	8.58	7.23	-15.73
TransferLostTime (min)	0.27	0.28	3.70
TotalTripDistance (mile)	4.78	4.81	0.63
AverageSpeed (mph)	7.45	7.93	6.44

Table 7.4: Average performance in West Suburbs to Downtown Chicago scenario

Avg. Expected Performance	Exponential Case	Erlang Case	Difference (%)
TotalTripTime (min)	43.79	40.20	-8.20
EnRouteTime (min)	25.98	26.39	1.58
WaitingTime (min)	8.48	6.76	-20.28
WalkingTime (min)	9.06	6.76	-25.39
TransferLostTime (min)	0.27	0.29	7.41
TotalTripDistance (mile)	4.28	4.30	0.47
AverageSpeed (mph)	5.83	6.38	9.43

Table 7.5: Average performance in Downtown Chicago to Downtown Chicago scenario

Avg. Expected Performance	Exponential Case	Erlang Case	Difference (%)
TotalTripTime (min)	16.80	16.51	-1.73
EnRouteTime (min)	4.28	4.76	11.21
WaitingTime (min)	1.45	1.59	9.66
WalkingTime (min)	10.95	10.03	-8.40
TransferLostTime (min)	0.12	0.13	8.33
TotalTripDistance (mile)	1.04	1.05	0.96
AverageSpeed (mph)	3.63	3.72	2.48

7.3 Algorithm comparison

7.3.1 Results by two greedy methods

We have discussed before two greedy methods for generating the attractive set if the headway distribution is not exponential. In this section, we test these two greedy methods for Erlang headway distribution in the aforementioned four scenarios. Hereafter, we denote the greedy method based on ranking the line travel time as Greedy 1, the other greedy method based on ranking the expected travel time of the line considered separately as Greedy 2. After running the four scenarios using both greedy methods, we find that the two greedy methods generate the same results. It implies that neither greedy method is superior to the other one in terms of the results accuracy, at least in this case study.

7.3.2 Efficiency

In this section, we solve the four scenarios using two different algorithms—label correcting and label setting algorithms. In each scenario, we need to solve 10 all-to-one shortest hyperpath problems. We will report the average time to solve a single all-to-one shortest hyperpath problem in the entire CTA network for different algorithms and parameter settings. All the instances are tested on a laptop with Window 7 Home Premium, Intel(R) Core(TM) i7-2630QM CPU@2.00GHz and 8.00 GB memory. The results for Exponential and Erlang headway distribution are shown in Table 7.6 and Table 7.7, respectively. Note that for Exponential headway distribution, we use only Greedy 1 method since it guarantees the optimal attractive set, for Erlang headway distribution, we report the results for both greedy methods. The computational results suggest that label setting algorithm is much faster than the label correcting algorithm. And Greedy 2 method consumes more CPU time than Greedy 1 method, which is expected since it will spend extra time to compute the expected travel time for each line separately.

Table 7.6: Computation time for Exponential headway distribution

	Label Correcting	Label Setting
Average CPU Time (s)	3	<0.2

Table 7.7: Computation time for Erlang headway distribution

	Label Correcting		Label Setting	
	Greedy 1	Greedy 2	Greedy 1	Greedy 2
Average CPU Time (s)	300	370	9	11

Appendix A

Derivation of the relationship between waiting time and headway distributions

Rosenberg (1968) establishes the relationship between waiting time distribution $F(\cdot)$ and headway distribution $G(\cdot)$ by analyzing the underlying discrete event of bus arrival. Let $A_i(t, x)$ denote the event that at least one bus from route i arrives between the interval $[t, t + x]$, where t is the passenger arrival time at the stop and x is the waiting time. Then the waiting time distribution can be expressed as

$$F(t, x) \equiv P\{w(t) \leq x\} = P\left\{\bigcup_{i=1}^N A_i(t, x)\right\} \quad (\text{A.1})$$

where $w(t)$ is the waiting time when passenger arrives at time t . The second equality follows from the equivalence of $w(t) \leq x$ and at least one bus arrives between the interval $[t, t + x]$. By assuming independence of each route, waiting time distribution is derived as

$$F(t, x) = 1 - \prod_{i=1}^N (1 - P\{A_i(t, x)\}) \quad (\text{A.2})$$

In particular, the waiting time distribution on a specific route r is

$$F_r(t, x) \equiv P\{w_r(t) \leq x\} = P\{A_r(t, x)\} \quad (\text{A.3})$$

The computation of $P\{A_i(t, x)\}$ is given in the following equation. Readers are referred to Rosenberg (1968) or Takács (1982) for detailed computation. The result is presented as

$$P\{A_i(t, x)\} = \int_t^{t+x} [1 - G_i(t + x - u)] dM_i(u) \quad (\text{A.4})$$

Appendix A. Derivation of the relationship between waiting time and headway distributions al.

where $G_i(x)$ is the CDF of headway, $M_i(u)$ is the average number of arrivals of a bus from route i in the interval $(0, u]$, which is called renewal function in renewal process. Computation of equation (A.4) involves integration of a CDF. Analytical form can be obtained only for special cases such as exponential distribution. Fortunately, numerical integration is always available to calculate such integration.

Another difficulty of computing equation (A.4) lies on computation of renewal function $M_i(u)$. For convenience, we suppress the subscript i and use $M(t)$ to refer to renewal function. There are extensive studies on renewal process, see Ross (2009), Ross (1996), Tijms & Wiley (2003) and Cox et al. (1962). Some important concepts and results of renewal process are presented below for references.

Definition A.1 (Renewal Process) *If the sequence of nonnegative random variables $\{X_1, X_2, \dots\}$ is independent and identically distributed, then the counting process $\{N(t), t \geq 0\}$ is said to be a renewal process.*

Definition A.2 (Renewal Function) *The average number of events up to time t :*

$$M(t) = E[N(t)] = \sum_{n=1}^{\infty} F^{(n)}(t)$$

where $F^{(n)}(t)$ is the n -fold Stieltjes convolution, which can be calculated recursively, defined as

$$F^{(n)}(t) = \int_0^t F^{(n-1)}(t-u) dF(u) \quad (\text{A.5})$$

In general, calculation of renewal function is not easy. Both numerical approach and approximation approach have been developed to study the behavior of renewal function. In the special case of asymptotic behavior where $t \rightarrow \infty$, renewal function can be easily calculated. There are established theorems to calculate renewal function.

Theorem A.1 (Blackwell's Theorem) *Let μ be the mean interarrival time. If the underlying interarrival distribution F is not lattice, then:*

$$\lim_{t \rightarrow \infty} (M(t+h) - M(t)) = \frac{h}{\mu} \quad (\text{A.6})$$

Theorem A.2 Let μ and σ^2 be the mean and variance of interarrival distribution. Then for large t ,

$$M(t) = \frac{t}{\mu} + \frac{\sigma^2}{2\mu^2} - \frac{1}{2} + o(1) \quad (\text{A.7})$$

Therefore, in the asymptotic case, applying the above theorems leads to the calculation of $dM(t)$, the derivative of renewal function.

$$dM(t) = d\frac{t}{\mu} = \frac{1}{\mu}dt \quad (\text{A.8})$$

Plugging equation (A.8) into equation (A.4) and (A.3) gives the CDF of waiting time distribution on route r .

$$\begin{aligned} F_r(t, x) &= \frac{1}{\mu_r} \int_t^{t+x} [1 - G_r(t + x - u)] du \\ &= \frac{1}{\mu_r} \int_x^0 [1 - G_r(y)] d(-y) \\ &= \frac{1}{\mu_r} \int_0^x [1 - G_r(y)] dy \end{aligned} \quad (\text{A.9})$$

With the above CDF of waiting time distribution, we can easily obtain the PDF as follows.

$$\begin{aligned} f_r(t, x) &= \frac{dF_r(t, x)}{dx} \\ &= \frac{1 - G_r(x)}{\mu_r} \end{aligned} \quad (\text{A.10})$$

where $G_r(x)$ is the CDF of headway distribution for route r . From the equations (A.9) and (A.10), the waiting time distribution for route r is independent of the passenger arrival time t when t is large.

Appendix B

VNET Manual

B.1 Installation

VNET can be downloaded at http://translab.civil.northwestern.edu/nutrend/?page_id=53. The setup package is a .msi file. Simply double click it to launch the setup wizard which will guide you through the installation process. Note that if an older version of VNET exist, it has to be removed before installing a new version. Currently VNET can only be used on Microsoft Windows XP and Windows 7.

B.2 A Quick Tutorial

VNET is very easy to navigate because it has a linear structure. All operations are structured and executed following a well-defined path, as described below.

Step 1: Specify a network type — choose one from the dropdown list.


Step 2: Choose a base file from the set of required files for the selected network type (any file can be used) by clicking .

Step 3: Load and navigate network by clicking .

Step 4: Choose an app – choose one from the dropdown list, which will be built according to network type.

Step 5: Run the app – click .

Step 6: Choose a post-run operation for analysis – choose from the dropdown list, which will be build according to app.

Step 7: Run the post — run operation. Click .

At any point, users may re-select an app or a file name - reselecting a file name necessitates a network re-loading, and reselecting an app necessitates a new app run.

B.3 Getting Started

Once VENT is installed, a VNET shortcut can be found on the desktop. Or it is always accessible through the installation folder. The default folder is "C:\Program Files (x86)\NUTREND\VNET".

User interface of VNET is shown in Figure B.1. The user interface consists of four major panels as follows:

Main panel: display the map or plots generated in VNET

Network panel: select network from files

Application panel: network-related applications and post operations

Log panel: display information in running

 provide help information about VNET. The button pops out the help window of VNET,

Figure B.2.

 switch between map view and plot view

 provide coordinates information of the cursor

 pop out a dedicated log window

 clean the current log window

 exit VNET

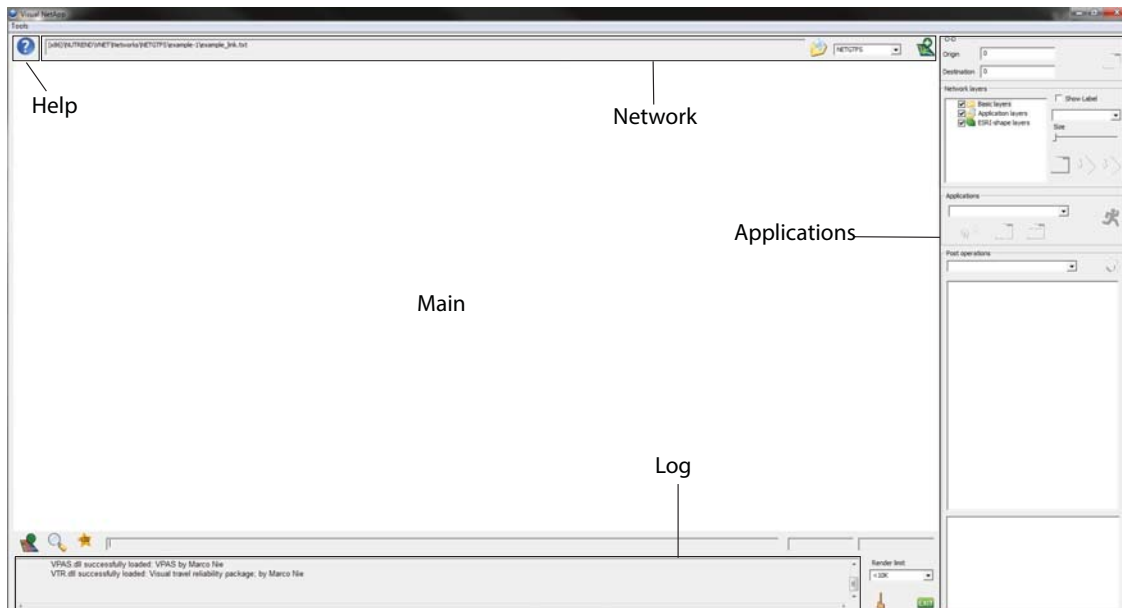


Figure B.1: User interface of VNET

B.4 Network Panel

B.4.1 Select network type

Before loading network files, network type has to be specified. Eight network types are available from the drop list for the current version. A more detailed description about each type can be found in [?](#) → Network types.

- **Empty:** allows users to use applications not directly tied to a defined network object
- **CTR** used for Chicago Reliable Routing project
- **FORT:** a static network format which describe networks using a forward star structure
- **FORTNO:** this is the same as FORT, except it is network topology only
- **TAPAS:** a static network format used by Hillel Bar-Gera
- **TAPASNO:** this is the same as TAPAS, except it is network topology only
- **GTFS:** Google Transit Feeder Specification

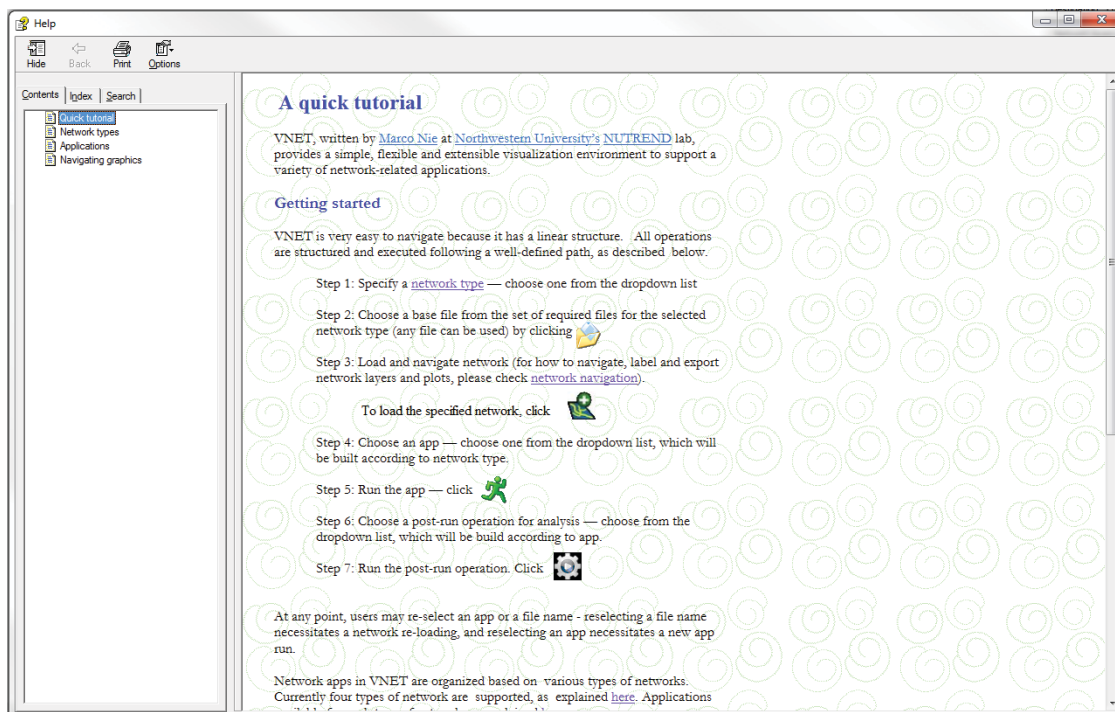




Figure B.2: Help window

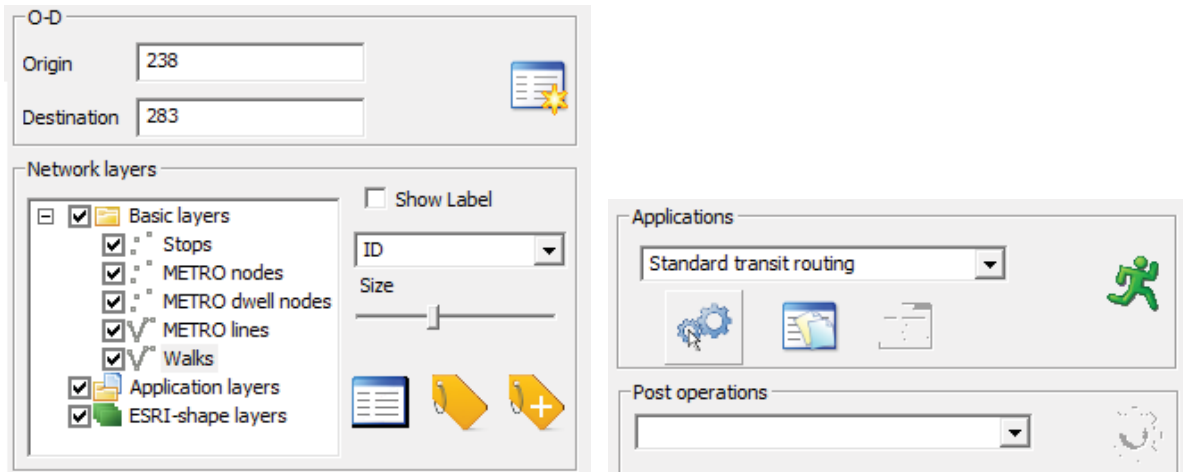
- **NETGTFS**: introduced by Marco Nie as a processed GTFS format

B.4.2 Load network file

Once network type is specified, clicking  to choose the base network file. Note that a valid network may contain multiple files with different extension. Any of them can be chosen to represent the network. Then, click  to load the network into VNET. Once a network is loaded, the main window provides an overview of the loaded network.

B.5 Applications Panel

Applications panel includes two components. The first one is a generic component which deals with the topology of the network (Figure B.3(a)). The second one is network type sensitive (Figure B.3(b)). The applications vary for different network type. Two network types are specifically related to transit, i.e. GTFS and NETGTFS. In the following of this section, the applications of the two types are discussed.





(a) Generic component of Applications Panel

(b) Network type sensitive component of Applications Panel

B.5.1 Generic panel

The generic panel provides universal functionality across all network type. Specifically, it provides functionality to deal with OD pairs and map layers.

1. O-D There are two ways to identify an OD pair.


- **Input box** Enter the OD pair directly into the boxes in the OD panel
- **OD selection window** Click  to open the "VNET Object Selector" window (Figure B.3). The window provides a list of information about the network nodes. Choose "Origin" or "Destination" from the drop list. Select any node from the list and click  to finalize the selection. Once it is done, the OD pairs are updated.

2. Network Layers Display different layers and objects on the map.

- **Show/Hide layers:** Check the box next to each layer to show/hide the layer (Figure B.4(a)).
- **Show label:** Check the box to show the selected label on the map. The label can be selected from the drop list below the check mark (Figure B.4(b)). The size of the label can be adjusted through the slide bar.

ID	Pattern	Type	Length	MeanHeadway	VarHeadway	MeanTravelTime	VarTravelTime
1	01-0-107	Enroute	4.67	0.00	0.00	6.00	0.00
2	01-0-107	Enroute	2.13	0.00	0.00	4.00	0.00
3	01-0-107	Enroute	3.39	0.00	0.00	5.00	0.00
4	01-0-107	Enroute	1.71	0.00	0.00	3.00	0.00
5	01-0-107	Enroute	3.14	0.00	0.00	5.00	0.00
6	01-0-107	Enroute	3.43	0.00	0.00	5.00	0.00
7	01-0-107	Enroute	4.44	0.00	0.00	5.00	0.00
8	01-0-107	Enroute	1.35	0.00	0.00	3.00	0.00
9	01-0-107	Enroute	1.44	0.00	0.00	4.00	0.00
10	01-0-107	Enroute	0.35	0.00	0.00	1.00	0.00
11	01-0-107	Enroute	1.26	0.00	0.00	5.00	0.00
12	01-0-107	Enroute	5.64	0.00	0.00	7.00	0.00
13	01-0-107	Enroute	0.36	0.00	0.00	1.00	0.00
14	01-0-107	Enroute	0.43	0.00	0.00	2.00	0.00
15	01-0-107	Enroute	0.53	0.00	0.00	1.00	0.00
16	01-0-107	Enroute	1.04	0.00	0.00	2.00	0.00
17	01-0-107	Enroute	0.89	0.00	0.00	2.00	0.00
18	01-0-107	Enroute	1.60	0.00	0.00	3.00	0.00
19	01-0-107	Enroute	1.05	0.00	0.00	2.00	0.00
20	01-0-107	Enroute	1.61	0.00	0.00	4.00	0.00
21	01-0-107	Enroute	1.49	0.00	0.00	4.00	0.00
22	01-0-107	Enroute	1.90	0.00	0.00	3.00	0.00
23	01-0-107	Enroute	2.43	0.00	0.00	4.00	0.00
24	01-0-107	Enroute	1.95	0.00	0.00	5.00	0.00
25	01-0-112	Enroute	3.39	0.00	0.00	5.00	0.00
26	01-0-112	Enroute	1.71	0.00	0.00	3.00	0.00
27	01-0-112	Enroute	3.14	0.00	0.00	5.00	0.00
28	01-0-112	Enroute	3.43	0.00	0.00	4.00	0.00
29	01-0-112	Enroute	4.44	0.00	0.00	6.00	0.00
30	01-0-112	Enroute	1.35	0.00	0.00	3.00	0.00
31	01-0-112	Enroute	1.44	0.00	0.00	4.00	0.00
32	01-0-112	Enroute	0.35	0.00	0.00	1.00	0.00
33	01-0-112	Enroute	1.26	0.00	0.00	5.00	0.00
34	01-0-112	Enroute	5.64	0.00	0.00	7.00	0.00
35	01-0-112	Enroute	0.36	0.00	0.00	1.00	0.00
36	01-0-112	Enroute	0.43	0.00	0.00	2.00	0.00
37	01-0-112	Enroute	0.53	0.00	0.00	1.00	0.00
38	01-0-112	Enroute	1.04	0.00	0.00	2.00	0.00
39	01-0-112	Enroute	0.89	0.00	0.00	2.00	0.00
40	01-0-112	Enroute	1.60	0.00	0.00	3.00	0.00
41	01-0-112	Enroute	1.05	0.00	0.00	2.00	0.00
42	01-0-112	Enroute	1.61	0.00	0.00	5.00	0.00
43	02-0-106	Enroute	1.95	0.00	0.00	4.00	0.00
44	02-0-106	Enroute	2.43	0.00	0.00	4.00	0.00
45	02-0-106	Enroute	1.90	0.00	0.00	3.00	0.00

Figure B.3: VNET Object Selector window

- **Edit layer properties:** Double click a layer to open "Graph and Objects Properties" window, Figure B.4.
- **Show layer properties:** Click  to show the layer information

B.5.2 GTFS Applications

The GTFS applications are activated when GTFS is selected as the network type. In the current version, the GTFS applications mainly serves as an interface to transform GTFS format to NETGTFS format. The current version only consists of one application.

1. Build transit network: Follow the next three steps to start the application.

Step 1. Set routing parameters by clicking .

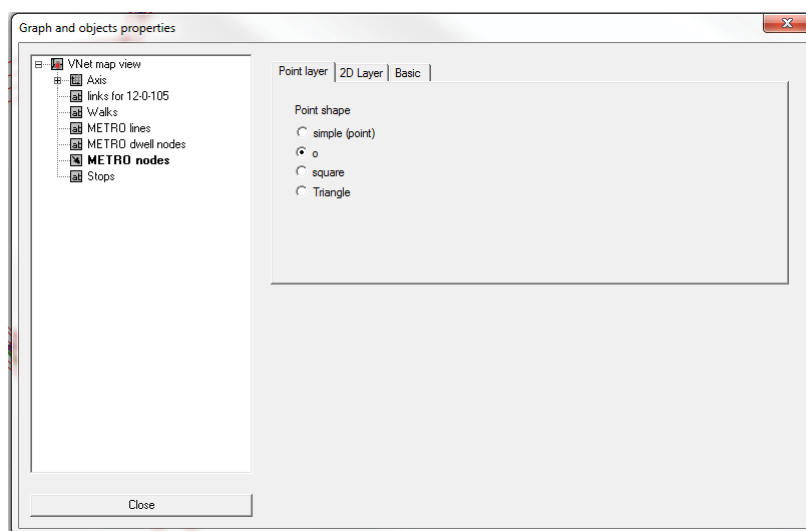
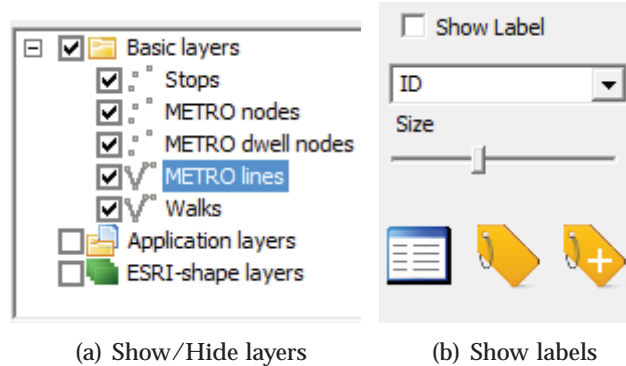



Figure B.4: Graph and Objects Properties window

Step 2. Reload the parameters by clicking . Note that reload must be executed very time the parameters are changed.

Step 3. Run the application by clicking . Once the application is ready, there are two post-operations available.

Once the application is ready, there are two post-operations available.

- **Plot trip trajectories:** Select a trip, then use this post-operation to show the trajectories, see Figure B.5.
- **Export:** export the GTFS format to other geographical format. In the current version, three options are available: (1) ESRI shape; (2) NETGTFS; (3) TAPAS.

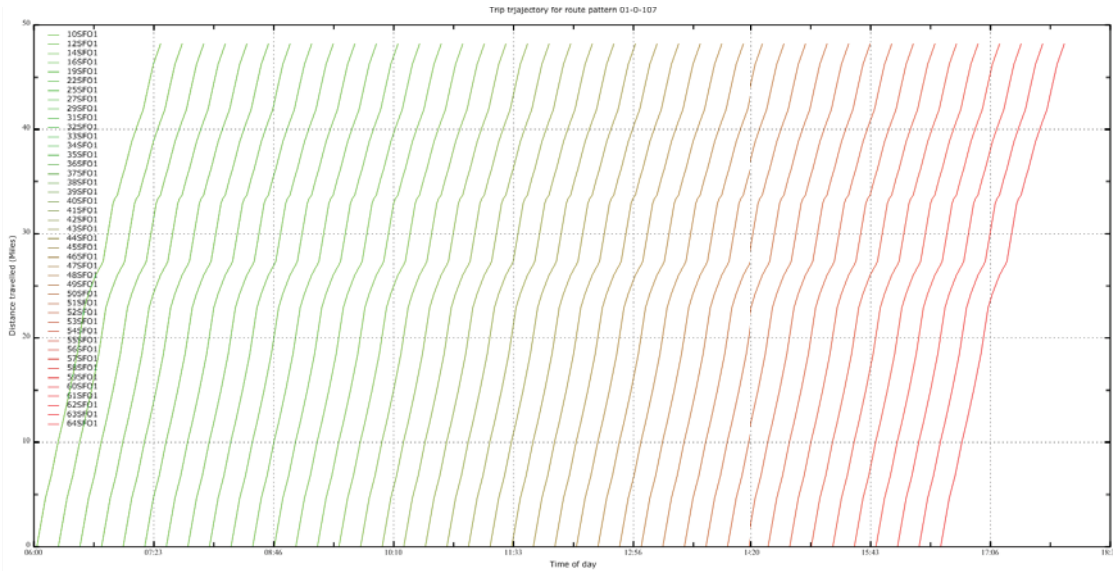


Figure B.5: All-to-one tree


B.5.3 NETGTFS Applications

The NETGTFS applications are activated when NETGTFS is selected as network type. The current version of VNET contains two applications:

- **Standard transit routing:** Transit Router application generates the all-to-one tree rooted on the destination
- **Real-time transit data viewer:** Transit Data Viewer viewer application provides visual aids to analyze transit real-time data

1. Standard transit routing: Follow the next three steps to start transit routing. A all-to-one tree is returned after execution, Figure B.6.

Step 1. Set routing parameters by clicking 

Step 2. Reload the parameters by clicking . Note that reload must be executed very time the parameters are changed.

Step 3. Run the application by clicking 

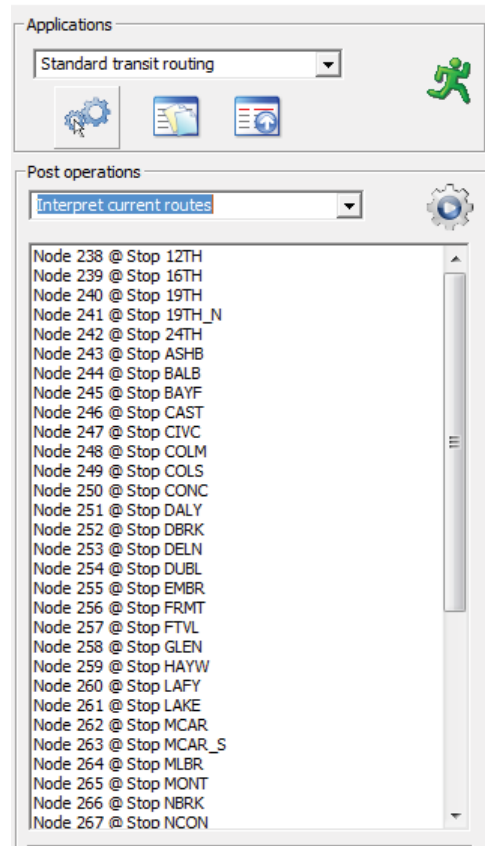


Figure B.6: All-to-one tree

Routing parameters greatly influence the result. The current version provides three routing schemes, Figure B.7.

- **Fastest route:** standard shortest path problem
- **Frequency-based hyperpath:** shortest hyperpath algorithm. Once frequency-based hyperpath is selected, other parameters can be set:
 - **Headway distribution:** specifies the bus headway distribution, three options are available: exponential, erlang and deterministic
 - **Information availability:** three options: (1) no information (passengers have no knowledge about route waiting time) (2) information (passengers know everything including waiting time) (3) partial information
 - **Attractive set method:** either greedy method or enumeration

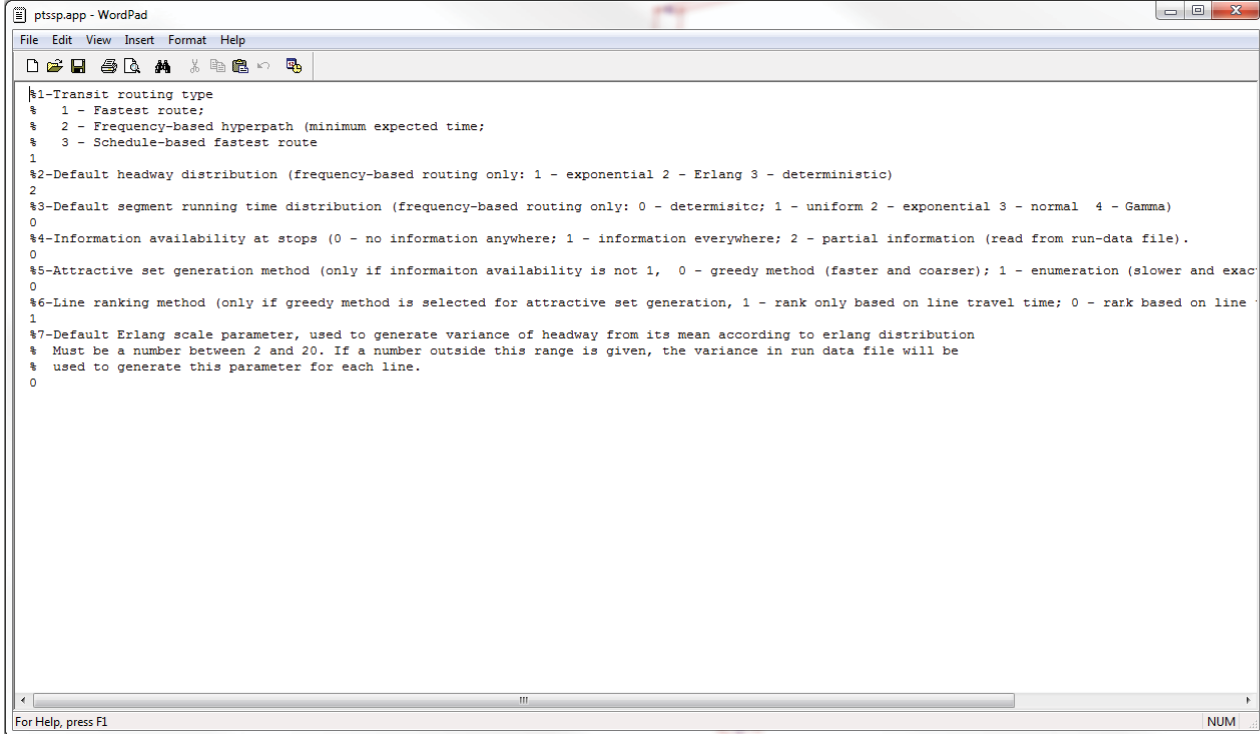




Figure B.7: Parameter setting window for transit routing

- **Schedule-based fastest route:** a new feature to be added in the future

After running transit routing, several post operations are available. Select one post operation and click  to execute.

- **Interpret current routes:** Click  to display the detailed route information.
- **Save current route:** Save current route for post operations.
- **Compare saved routes:** Compare multiple saved routes.
- **Delete saved routes:** Deleted the selected routes from saving list.
- **Interpret saved routes:** Interpret multiple routes.

2. Real-time transit data viewer: Following the next three steps to start real-time transit data viewer. The parameter setting window is shown in Figure B.8.

Step 1. Set routing parameters by clicking 

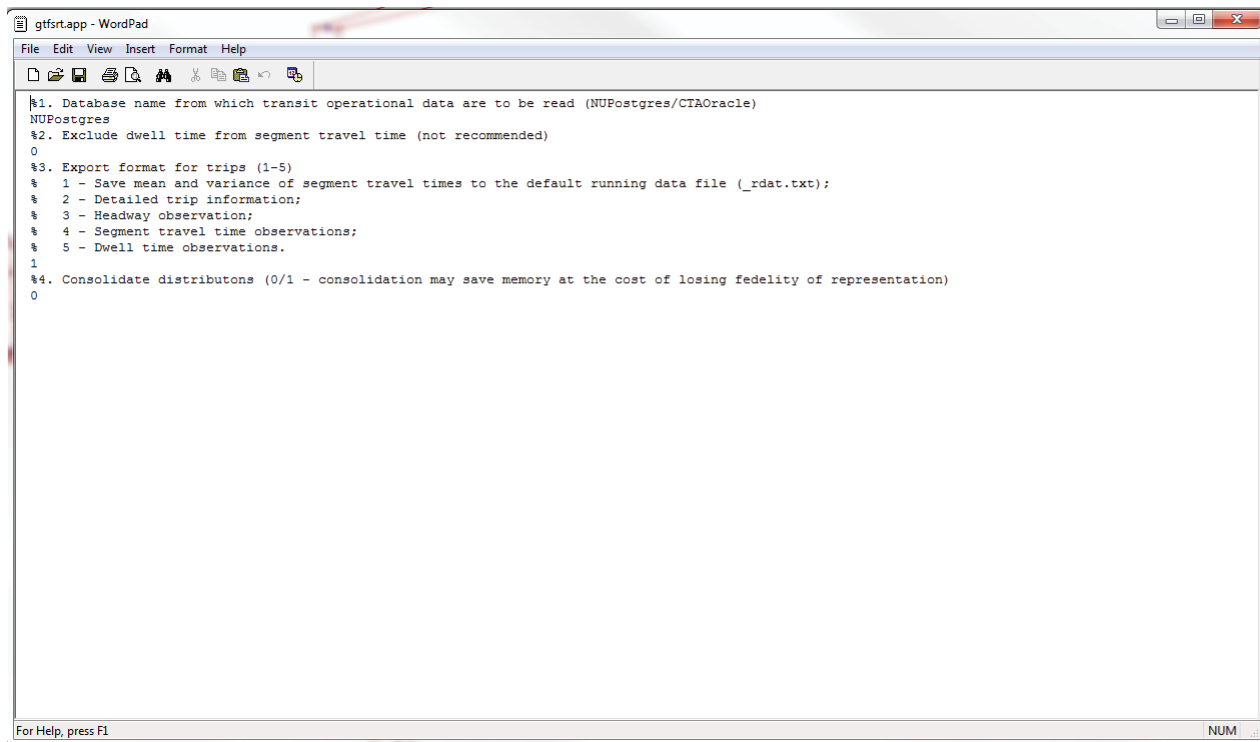




Figure B.8: Parameter setting window for data viewer

Step 2. Reload the parameters by clicking . Note that reload must be executed every time the parameters are changed.

Step 3. Run the application by clicking .

The post operations include the following options. The function of each operation is self-explanatory. Once a option is selected, click  to run the operation. Note that, some operations requires multiple inputs.

- Load trips

As in Figure B.9, the upper block shows the pattern ID. Note that each pattern represent a geographic sequence of stops to form an actual route. However, different route bearing the same route ID may have different patterns. For example, route A may have both weekday pattern and weekend pattern.

Once a pattern is selected, all the trips associated with the pattern are displayed in the

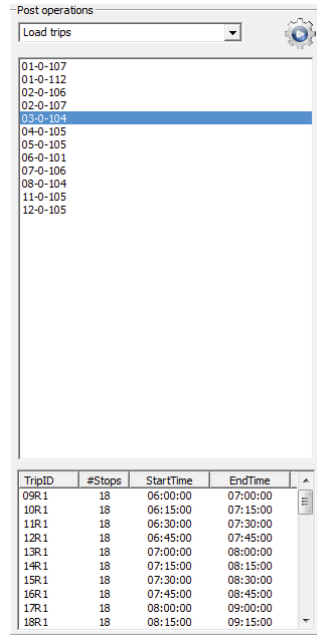


Figure B.9: Parameter setting window for data viewer

lower block. The trip information is also displayed. From here one, VNET applications can be used for analysis.

- **Clear loaded trips:** Delete loaded trips from the memory.
- **Save loaded trips:** Save loaded trips for future comparison
- **Plot trips (loaded):** Once trips are loaded, this tool can be used to plot the real time-spatial trajectory of the trip, see Figure ??.
- **Plot trips (Scheduled):** In contrast to the real time-spatial plot, scheduled time-spatial plot can be drawn with this tool, see Figure ?. It is clear that real plot exhibits variations during the time of operation. In extreme cases, we can observe bunching or even overlapping.
- **Plot trips (both):** We can also plot both in the same graph to examine how the real data deviates from the scheduled data.
- **Segment statistics:** Plot the Probability Density Function (PDF) and Cumulative Density Function (CDF) of travel time and speed for the selected segment, see Figure

??.

- **Headway statistics:** Display the PDF and CDF of the headway distribution over the selected segments, see Figure ??.
- **Spatial headway analysis:** Display headway variation as a function of distance. In most cases, as the distance from the origin increases, the headway variation increases, see Figure ?. A more detailed analysis can be found in the Case study section.
- **On-time statistics:** Display the PDF and CDF of on-time deviation from the schedule, see Figure ??.
- **Spatial on-time analysis:** Display on-time distribution and percentile, see Figure ??.
- **Dwell time statistics:** Display the PDF and CDF of dwell time, see Figure ??.
- **Spatial dwell time analysis:** Display dwell time deviation and percentile, see Figure ??.
- **Export loaded trips:** Export loaded trips into .txt files.
- **Export all trips:** Export all trips into .txt files.
- **Simulation:** To be added in the future version.

Bibliography

- Bouzaïene-Ayari, B., Gendreau, M. & Nguyen, S. (2001), 'Modeling bus stops in transit networks: A survey and new formulations', *Transportation Science* **35**(3), 304–321.
- Cepeda, M., Cominetti, R. & Florian, M. (2006), 'A frequency-based assignment model for congested transit networks with strict capacity constraints: characterization and computation of equilibria', *Transportation Research Part B: Methodological* **40**(6), 437–459.
- Chriqui, C. & Robillard, P. (1975), 'Common bus lines', *Transportation Science* **9**(2), 115–121.
- Cominetti, R. & Correa, J. (2001), 'Common-lines and passenger assignment in congested transit networks', *Transportation Science* **35**(3), 250–267.
- Cox, D., Cox, D., Cox, D. & Cox, D. (1962), *Renewal theory*, Vol. 1, Methuen London.
- de Cea, J. & Fernández, E. (1993), 'Transit assignment for congested public transport systems: an equilibrium model', *Transportation science* **27**(2), 133–147.
- Gendreau, M. (1984), *Étude approfondie d'un modèle d'équilibre pour l'affectation des passagers dans les réseaux de transport en commun*, Montréal: Université de Montréal, Centre de recherche sur les transports.
- Gentile, G., Nguyen, S. & Pallottino, S. (2005), 'Route choice on transit networks with online information at stops', *Transportation science* **39**(3), 289–297.
- Hickman, M. D. & Wilson, N. H. (1995), 'Passenger travel time and path choice implications of real-time transit information', *Transportation Research Part C: Emerging Technologies* **3**(4), 211–226.

- Holroyd, E. M. & Scraggs, D. A. (1966), 'Waiting time for buses in central london', *Traffic Engineering & Control* **8**, 158–160.
- Larson, R. C. & Odoni, A. R. (1981), *Urban operations research*, number Monograph.
- Marguier, P. H. & Ceder, A. (1984), 'Passenger waiting strategies for overlapping bus routes', *Transportation Science* **18**(3), 207–230.
- Nguyen, S. (1989), 'Hyperpaths and shortest hyperpaths', *Combinatorial Optimization* .
URL: <http://www.springerlink.com/index/VP45481420218011.pdf>
- Nguyen, S. & Pallottino, S. (1988), 'Equilibrium traffic assignment for large scale transit networks', *European journal of operational research* **37**(2), 176–186.
- Nie, Y., Wu, X., Zissman, J., Lee, C. & Haynes, M. (2010), Providing reliable route guidance: Phase ii, Technical report, Center for the Commercialization of the Innovative Transportation Technology, Northwestern University.
- O'Flaherty, C. A. & Mangan, D. O. (1970), 'Bus passenger waiting time in central areas', *Traffic Engineering & Control* **11**, 419–421.
- Osuna, E. & Newell, G. (1972), 'Control strategies for an idealized public transportation system', *Transportation Science* **6**(1), 52–72.
- Rosenberg, L. (1968), 'Mean waiting time as a measure of effectiveness', *The American Statistician* **22**(4), 31–34.
- Ross, S. (1996), *Stochastic processes.*, New York, NY: John Wiley & Sons.
- Ross, S. (2009), *Introduction to probability models*, Academic press.
- Ruan, M. & Lin, J. (2009), An investigation of bus headway regularity and service performance in chicago bus transit system, in 'Transport Chicago 2009 Annual Conference, <http://www.transportchicago.org/uploads/5/7/2/0/5720074/intelligentbus-ruanlin.pdf>'.

- Seddon, P. & Day, M. (1974), 'Bus passenger waiting times in greater manchester', *Traffic Engineering and Control* **15**(9), 442–445.
- Spiess, H. & Florian, M. (1989), 'Optimal strategies: A new assignment model for transit networks', *Transportation Research Part B: Methodological* **23**(2), 83–102.
- Takács, L. (1982), *Introduction to the Theory of Queues*, Greenwood Press.
- Tijms, H. & Wiley, J. (2003), *A first course in stochastic models*, Vol. 2, Wiley Online Library.
- Welding, P. (1957), 'The instability of a close-interval service', *OR* **8**(3), 133–142.
- Wu, J., Florian, M. & Marcotte, P. (1994), 'Transit equilibrium assignment: a model and solution algorithms', *Transportation Science* **28**(3), 193–203.