Scenario Analyzer for Real-time Dynamic Transportation Assignment (DTA) Systems

by

Yihang Sui

B.Sc. in Resources Environment & Urban and Rural Planning

Submitted to the Department of Civil and Environmental Engineering in partial fulfillment of the requirements for the degree of Master of Science in Transportation at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2018

© Massachusetts Institute of Technology 2018. All rights reserved.

Signature redacted

Author ............................................

Department of Civil and Environmental Engineering
May 11, 2018

Signature redacted

Certified by. ..............

Moshe E. Ben-Akiva

Edmund K. Turner Professor of Civil and Environmental Engineering
Thesis Supervisor

Signature redacted

Accepted by ..............

Jesse Kroll

Professor of Civil and Environmental Engineering
Chair, Graduate Program Committee
DISCLAIMER NOTICE

Due to the condition of the original material, there are unavoidable flaws in this reproduction. We have made every effort possible to provide you with the best copy available.

Thank you.

The images contained in this document are of the best quality available.
Scenario Analyzer for Real-time Dynamic Transportation Assignment (DTA) Systems

by

Yihang Sui

Submitted to the Department of Civil and Environmental Engineering on May 11, 2018, in partial fulfillment of the requirements for the degree of Master of Science in Transportation

Abstract

The optimization of network control strategies using real-time Dynamic Traffic Assignment systems typically utilizes short-term predictions of the network state within a rolling horizon framework. However, there exist several network control instruments (such as incentive schemes under daily budget constraints) whose optimization necessitate generating predictions beyond the “roll period” and for the entire day. This work addresses the aforementioned problem by proposing a “Scenario Analyzer” to extend the horizon for the optimization problem by providing relatively accurate predictions and forecasting results for the extended horizon.

The Scenario Analyzer module adopts a data driven approach, and is formulated as a matching problem utilizing an archived historical database. The archived historical database includes the data from DTA systems as master data table, daily run table and historical scenario table. The matching algorithms use the historical scenario table and master data table to pair the current run feature(s) with historical runs feature(s); after finding a match, the current run will be stored at the daily run table. The matching problem may be solved using different statistical or machine learning algorithms, in terms of: 1) single time step feature matching 2) multiple time steps features matching.

The performance of the proposed scenario analyzer is examined for the optimization of an app-based travel incentive scheme to reduce system wide energy consumption (referred to as Tripod) in the Boston CBD network. The k-NN and KL divergence matching algorithms are tested for a simulation period of 6 AM - 9 PM. Results indicate that the scenario analyzer with k-NN outperforms KLD algorithm probably because KLD need more data points to fully-develop the time-series properties. Among all the traffic features using in the matching algorithms, the cumulative energy consumption is the best indicator for similarity comparison.

Thesis Supervisor: Moshe E. Ben-Akiva
Title: Edmund K. Turner Professor of Civil and Environmental Engineering
Acknowledgments

During the two years I have spent at MIT I have been very fortunate to have received the help and support of numerous people.

First, I would like to express my sincere gratitude to my advisor, Professor Moshe Ben-Akiva. Thank you for all your support. It’s a great pleasure to work with you as a Research Assistant for one and a half year. I learned a lot from you and enhance my research skills.

Thanks also go to Dr. Carlos Lima Azevedo, Ravi Seshadri and Sankaran Ramakrishnan, for all your inspirations and guidance through my masters research. The topic is quite challenging and thank you for your encouragement. All of you taught me how to narrow down to your research from a large and novel topic and how to be a better researcher.

Following that, I want to thank other members of the ITS Lab, Haizheng Zhang, Xiang Song, Mazen Danaf, Bilge Atasoy, Peiyu Jing, Andrea Araldo, Eytan Gross, Ajinkya R. Ghorade, Isabel Viegas de Lima, Yundi Zhang, Samarth Gupta, Yifei Xie and Siyu Chen.

Also thanks to the friends I met at MIT, who are considerate and helpful and made my life at MIT colorful and enjoyable. Thank you, Yan Liu, Xizi Zeng, Jiachen Mao, Xuanan Ni, Xinhui Li, Xiaoyan Shen, Yilin Xie, Fengjiao Peng, Xin Huang, Jian Wen, Gokce Solakoglu and Rounaq Basu.

Last but not least, my deepest gratitude goes to my families and my boyfriend, Shun Xu, for your endless love, encouragement and constant support. Thank you all again for the support.
Contents

1 Introduction
  1.1 Motivation .................................................. 13
  1.2 Thesis Structure ........................................... 15

2 Literature Review ............................................... 17
  2.1 Overview of DTA System ..................................... 17
    2.1.1 Components of DTA Systems .......................... 18
    2.1.2 SODTA System ........................................... 19
    2.1.3 Application ............................................. 21
    2.1.4 Limitation of DTA models ............................ 23
  2.2 Data-driven Methods for Traffic Management .......... 23
    2.2.1 Traffic Simulation: State Estimation, Prediction and Calibration 24
    2.2.2 Traffic Planning: Short-term and Long-term ............ 24
    2.2.3 Traffic Features Selection for Data-driven Methods .......... 25
  2.3 Summary of Precedents and Challenges ................. 26

3 Methodology .................................................... 27
  3.1 DTA System Implementation ............................... 27
    3.1.1 Generic DTA System .................................. 27
    3.1.2 Generic System Optimal Dynamic Traffic Assignment (SODTA) Model Implementation ............................................. 28
  3.2 Scenario Analyzer ............................................ 30
    3.2.1 Matching Algorithm .................................... 32
## List of Figures

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2-1</td>
<td>Generic DTA algorithmic procedure (Chiu et al. [2011])</td>
<td>19</td>
</tr>
<tr>
<td>2-2</td>
<td>Structure of a generic DTA model (Chiu et al. [2011])</td>
<td>20</td>
</tr>
<tr>
<td>3-1</td>
<td>Generic DTA framework (Ben-Akiva et al. [2010a])</td>
<td>28</td>
</tr>
<tr>
<td>3-2</td>
<td>SODTA framework</td>
<td>31</td>
</tr>
<tr>
<td>3-3</td>
<td>Interacting with scenario analyzer</td>
<td>33</td>
</tr>
<tr>
<td>4-1</td>
<td>DynaMIT Framework (Ben-Akiva et al. [2010a])</td>
<td>40</td>
</tr>
<tr>
<td>4-2</td>
<td>Tripod Framework (Azevedo et al. [2018])</td>
<td>41</td>
</tr>
<tr>
<td>4-3</td>
<td>System Optimization Framework</td>
<td>43</td>
</tr>
<tr>
<td>4-4</td>
<td>Boston CBD network</td>
<td>44</td>
</tr>
<tr>
<td>4-5</td>
<td>Boston CBD network in DynaMIT</td>
<td>45</td>
</tr>
<tr>
<td>4-6</td>
<td>Demand for morning peak</td>
<td>46</td>
</tr>
<tr>
<td>4-7</td>
<td>Example of k-NN classification(Bronshtein [2017])^1</td>
<td>48</td>
</tr>
<tr>
<td>4-8</td>
<td>Energy consumption for two similar days (existing KL-divergence sample)</td>
<td>51</td>
</tr>
<tr>
<td>4-9</td>
<td>Scenario Analyzer’s Interface</td>
<td>57</td>
</tr>
<tr>
<td>4-10</td>
<td>Actual and estimated beyond-horizon energy consumption versus time (with CumEC feature)</td>
<td>60</td>
</tr>
</tbody>
</table>
List of Tables

4.1 Master Data: basic information of simulation ............................... 51
4.2 Master Data: OD pair ID and its corresponding origin and destination 52
4.3 Master Data: network information ........................................... 52
4.4 Daily run: state estimation (or prediction) table .......................... 52
4.5 Historical data: information sheet ........................................... 53
4.6 Historical data: result sheet .................................................... 53
4.7 Prediction errors for beyond-horizon energy consumption using the k-
    NN matching algorithm ...................................................... 58
4.8 Prediction errors for beyond-horizon energy consumption using the KL
    divergence matching algorithm ............................................ 59

A.1 Master Data: basic information of simulation .............................. 67
A.2 Master Data: OD pair ID and its corresponding origin and destination 68
A.3 Master Data: network information .......................................... 68
A.4 Daily run: metadata for daily run of DynaMIT ............................ 68
A.5 Daily run: state estimation table .......................................... 68
A.6 Daily run: state prediction table ......................................... 69
A.7 Historical scenario: information table .................................... 69
A.8 Historical scenario: result table .......................................... 69
Chapter 1

Introduction

The development of Intelligent Transportation systems and emergence of new data sources has enabled a more efficient and effective control of transportation facilities. To provide accurate and reliable information and guidance to the ITS system, the traffic simulation, including OD estimation, route choice etc., becomes more and more important. From the late 1970s, real-time Dynamic Transportation Assignment (DTA) systems have become an essential tool for dynamic traffic estimation and prediction. With the improvement of computation resources and the emergence of innovative techniques, the DTA models are increasingly discussed and widely used for simulation and operational planning. In this thesis, we propose a novel data-driven method, called Scenario Analyzer, to help improve the optimization quality of real-time DTA systems, in terms of extending the optimization horizon with acceptable accuracy, which is especially useful for ‘real-time’ or management. This chapter describes the motivation for this thesis and presents the thesis outline.

1.1 Motivation

Inefficiency and insufficiency problem in transportation systems has concerned millions of people for decades. Inefficient transportation facilities causes increased use of energy and pollution. Insufficient increase of transportation infrastructure causes severe congestion all over the world.
In order to achieve an efficient transportation system, one of the goals is reducing vehicle miles traveled (VMT). VMT reduction strategies can reduce traffic congestion, enable the use of more efficient vehicles, reduce transportation costs and save time for drivers. However, from 1971 to 2016, the annual VMT in the United States has seen a long rise from 1.13 trillion miles to 3.17 trillion miles, which is a 3.92% annual increase rate. Among these 46 years, 2016 experienced the largest annual increase in VMT since tracking began in 1971 (AFDC [2016]).

State and federal government have tried different methods, such as creating new infrastructures, optimizing signal time, transferring to energy efficient fuel and importing incentives scheme, to manage transportation in a more cost-effective manner. ITS system with the help of real-time DTA systems is a straightforward way using information technology to monitor and influence transportation system. Built upon accurate transportation state simulation, the DTA systems would help provide travel mode, departure time, route and other important information and advisories to travelers for meeting various traffic management and control objectives (Miaou et al. [1999]).

When making strategies for traffic management, system optimal dynamic traffic assignment (SODTA) models, which is an extension of DTA systems, would provide the best network performance by solving suitable optimization problems in real-time and achieving the optimal solution. The SODTA models are useful in the sense that the users in a network are guided by a central controller to effectively travel so as to help maintain system-wide objectives (Lu [2012]). However, one limitation or disadvantage of DTA systems is that, the simulation efficiency is not high enough for relatively long prediction horizon. Thus, for longer-term optimization problem (also called beyond-horizon optimization problem in this thesis), we need to extend the horizon for optimization by providing accurate prediction or forecasting result for the extended horizon.

The goal of this thesis is to develop an efficient data-driven tool, to help extend the optimization period to beyond-horizon, thus formulating and solving the SODTA problem for longer prediction horizons in ideal or general networks. In this thesis,
Scenario Analyzer, a data-driven algorithm, is introduced and fully examined to fulfill the main goal.

1.2 Thesis Structure

In section 1.1, the motivation of this thesis is discussed. We stated that DTA systems could provide traffic state information and network control, which could help improve transportation system efficiency. As state estimation and prediction would be used for system's route guidance, the estimation and prediction ability is essential to the DTA system. Thus, developing a data-driven method that helps with extending the state prediction horizon is important for traffic operations and management.

The outline of the remaining thesis is as follows.

Chapter 2 provides an introduction to DTA systems. The DTA system design, evaluation aspects and practical application will be shown. After that, the SODTA system is reviewed, which provide the optimization functionality to DTA models. In addition, we reviewed the newly discussed data-driven methods for DTA system simulation and their traffic features selected for simulation. The scope of this thesis is narrowed down to optimization problems for real-time DTA systems.

Chapter 3 gives more detailed information of how DTA systems can be integrated with the scenario analyzer for specific optimization problems. Then, the scenario analyzer and control architecture is separately reviewed. Following the system implementation, a general evaluation metric is proposed to help select best scenario analyzer for the specific research purpose.

In Chapter 4, a case study of Boston Central Business District (CBD) network is conducted. We examine two algorithms for the scenario analyzer that works for beyond-horizon energy optimization problem with DynaMIT (a real-time DTA system) and Tripod (a sustainable travel incentives scheme). Then we compare the performance of algorithms and select the best algorithm for scenario analyzer. Finally, the performance of the best algorithm and DynaMIT simulation is compared.

In Chapter 5, we finally discuss the overall contributions of the research. In
addition, we propose potential directions for future work.
Chapter 2

Literature Review

In the context of this thesis, we are interested in reviewing the work previously done that contributes to our research on developing a data-driven tool to help with the optimization problem within DTA systems. Since the model is based on the framework of DTA systems, we will review the literature concerning the different aspects of DTA systems in section 2.1. Then, data-driven methods for traffic simulation are summarized in section 2.2.

2.1 Overview of DTA System

For the purpose of transportation planning and traffic simulation, travel forecasting models are used to evaluate the impact of land use, transportation facilities or transportation demand on the performance of local and statewide transportation system. In the process, traveler behavior is introduced and highly emphasized.

Traditional four-step model process analyze groups of homogeneous travelers in aggregate trip-based models. Thereafter, more advanced processes which represent travel choices on the individual travelers basis are motivated. Among those, DTA system represents time-varying network and dynamic demand interaction, thus each individual traveler make their travel choices.

Since the late 1970s, DTA system has evolved substantially. The methodology of DTA systems and the application of DTA systems were mostly discussed. Analytical
approaches formulated DTA models as mathematical programming problems, so as to achieve the dynamic user equilibrium (DUE). With the development and improvement of DTA models, the simulation-based DTA models become popular. Simulation-based DTA models could not only address several modeling issues that are troublesome in analytical formulations, but also captures the complex vehicle interactions, thereby evaluating the non-linear objective function satisfactorily compared to idealized cost functions (Peeta and Ziliaskopoulos [2001]).

The simulation-based DTA systems can represent route choices and system controls as expected. However, the system optimization problems that could be solved under the simulation-based DTA structure was not fully developed. Thus, a more appropriate structure, system optimization DTA (SODTA) model, to optimize system-wide benefits was introduced.

2.1.1 Components of DTA Systems

DTA models is a complex dynamic traffic system, which determines time-dependent network conditions by modeling individual route choices and temporal dynamics in network supply. The goal of traffic assignment is to determine the network traffic flows and conditions that result from the mutual interactions among the route choices that travelers make in traversing from their origins to their destinations, and the congestion that results from their travel over the network (Chiu et al. [2011]).

Miaou et al. [1999] summarized the key DTA system variables that are likely to be considered in the design: (1) ATIS type and market penetration rate; (2) Travel-mode, departure-time, and route-choice behavior distributions; (3) Travel demand level, elasticity and variability; (4) Network size, configuration, and functional composition; (6) Surveillance system deployment rates; (7) Reliability of surveillance data; (8) Incidents; (9) Traffic controllers.

Chiu et al. [2011] summarized the most common method of finding equilibrium in DTA and the structure of a generic DTA model (shown in Figure 2-1 and Figure 2-2).

The demand and supply modules are the main input components of DTA system, which includes OD flows, route choice, network loading, traffic control and etc. After
the iterative three step procedure, a stopping criterion is met and the traffic conditions and routing decisions are output to the system.

2.1.2 SODTA System

System optimal approach minimizes the generalized cost for the system holistically (Sheffi [1985]). Aiming at providing the best network performance, the SODTA models optimize total system performance on individual level. Thus, the individual route choice and traffic conditions are affected and may be different from traditional DTA systems.

The SODTA systems provide heated topic for finding a fully developed structure and algorithmic models to solve the optimization problem. Most of the research on SODTA focused on finding a possible and efficient mathematical solution for the system optimal problem. For example, Merchant and Nemhauser [1978] presented a model to minimize cost using a linear cost function assumption (LCFA) by decomposing it into a piecewise problem. Waller et al. [2013] introduced a Linear Programming (LP) formulation to solve system optimal DTA problem that captures strategic route choice, travel behavior and the most time saving path under demand uncertainty. Lu [2012] formulated and dealt with the path-based SODTA problem with uncertain de-
mands using a column generation-based algorithmic framework with scaled gradient projection algorithm.

Traditionally, the SODTA system considers the travel time cost as the main source of system cost. While environment issues become more and more important in our life, the environmental and energy optimization problem come into discussion. Aziz and Ukkusuri [2012] integrated emission-based objective (CO emissions) into the traditional travel time based DTA framework and the results indicated changes in route choice behavior compared with traditional DTA system.

However, under the DTA system structure, most of the optimization is restricted by the prediction horizon. For traditional simulation-based DTA systems, the prediction horizon is 15 to 30 minutes. In this thesis, we will introduce a method to ‘extend’ the horizon for optimization to longer time, for example, beyond the prediction horizon until the end of the day.
2.1.3 Application

The DTA and SODTA systems are widely used on transportation planning, demand forecasting, real-time operational control and other transportation related areas.

1. Traffic simulation (including estimation and prediction)

Existing simulation-based DTA models fall in 3 categories due to different levels of detail in terms of presenting traffic dynamics: microscopic, mesoscopic and macroscopic (Lu et al. [2013]).

Microscopic DTA models have the most detailed simulation, while macroscopic DTA models denote the least detailed simulation. The existing simulation-based microscopic systems include AIMSUN2 (Barceló and Casas [2005]), MITSIMLab (Ben-Akiva et al. [2010b]), VISSIM (PTV [2015a]). Macroscopic DTA models include Visum (PTV [2015b]), EMME (INTO [2015]) and etc. Mesoscopic models are a combination of macroscopic and microscopic models, with the aim of balance between efficiency and accuracy. Examples include DynaMIT (Ben-Akiva et al. [2010a]), DYNASMART (Mahmassani et al. [1998]) and etc.

2. Operational planning

An informative simulation-based DTA system is equipped with traffic state estimation and prediction functions, which helps making planning decisions for major operations, construction, or demand management actions, however all of them are off-line application.

Various DTA frameworks have been used to help with the operational planning. For example, Kamga et al. [2011] examined the distribution of travel time of origin-destination (OD) pairs on a transportation network under incident conditions and proves that an effective DTA system for traffic management and information center could help with traffic incident control.

Congestion problems are also caused by rapid urbanization in most countries. Several papers in the literature have used DTA models to control the traffic system
under congestion situations. Ben-Akiva et al. [2012] used DynaMIT-P, a mesoscopic
dynamic traffic assignment simulation system, to manage the severe congestion in a
sub-area of Beijing. Under the condition of traffic events (such as severe weather,
large traffic accidents etc.), congestion would also happen. For that, Du et al. [2018]
proposed a network flow based two-layer DTA model (NTDM), thus largely improving
the operational efficiency of an expressway network after events occurring.

DTA models are also applied to support modeling efforts related to lane manage-
ment. Shelton et al. [2008] focused on the DTA model development and the evaluation
of managed lane strategies applied to freeway ramps in a test network. Kuhn et al.
[2008] investigated the application of different demand management strategies, in-
cluding DYNASMART-P (a DTA system), to mainlane ramps and managed lane
ramp operations during the peak period.

3. Optimization problems within DTA systems

With the help of DTA models and intelligent transportation systems, researchers
start to integrate and solve optimization problems within DTA systems, which also
motivates the development of SODTA systems. The optimization problems include,
for example, environmental objectives, traffic optimization, resources allocation and
etc.

Long et al. [2016] accommodated environmental objectives into SODTA models
and use the link transmission model (LTM) to minize total system emissions (TSE)
in single destination networks. Papatzikou and Stathopoulos [2015] proposed an
optimization method for combining DTA and network control to optimize traffic sig-
nalization in urban areas. Wang et al. [2016] proposed a real-time toll optimization
system based on predicted traffic conditions from DTA systems. Long et al. [2016]
accommodated environmental objectives into SODTA models and use the link trans-
mission model (LTM) to minimize total system emissions (TSE) in single destination
networks. Papatzikou and Stathopoulos [2015] proposed an optimization method of
traffic signalization in urban areas, which combines DTA and network control by
minimizing the risk of potential loss. Wang et al. [2016] proposed a real-time toll
optimization system based on predicted traffic conditions from DTA system.

2.1.4 Limitation of DTA models

For large-scale dynamic traffic network modeling and traffic simulation, the DTA models have reached sufficient maturity to provide accurate results with enough computation resources. However, the DTA systems still have some limitations in application.

Peeta and Ziliaskopulos [2001] grouped the challenges for DTA system into three categories: 1) real-time deployment, 2) planning and 3) fundamental issues. Chiu et al. [2011] summarized cautions for using DTA models: 1) DTA models are not the universal cure, 2) most existing DTA models focus on route choice and relatively few are implemented for departure time or arrival time choice, 3) DTA models could represent the effects of most existing traffic signal control logics, however the representation is relatively simplistic, and 4) simulation-based DTA models generally do not strictly conform to mathematical properties.

On the operation side, the real-time deployment is the most acknowledgeable issue. Since DTA systems highly depend on the input data source, the data reliability and the data quality accounts for the model controls and performance. Also, the computational power of available machines influence the DTA models running efficiency.

On the planning side, the long-term planning, multi-modes person assignment are the limitation of DTA systems. For example, for long-term planning, the time-dependent demand is intractable problem as the available level of input data required for DTA may not be available.

2.2 Data-driven Methods for Traffic Management

With the emergence of a number of advanced computational approaches, data-driven methods offer the potential for the development of approaches that are more accurate and efficient for capturing characteristics of DTA models. Theses data-driven methods could work for multiple traffic tasks, including traffic state estimation and prediction,
long-term traffic signal planning, and etc.

2.2.1 Traffic Simulation: State Estimation, Prediction and Calibration

Traffic state estimation and prediction is a key problem with considerable implications in intelligent transportation system operational management. Various modeling approaches have been developed and improved to innovate in state estimation and prediction field, such as neural networks, cluster and classification analysis, support vector regression (SVR) and multimodal regression.

The data-driven approaches could not only work for solving the NP hard modeling problems, but also help with the modeling parameters calculation. Antoniou et al. [2013] introduced a two-step approach for short-term prediction of traffic state and local speed, which takes advantage of clustering and Markov chain training methods. Varia et al. [2013] applied the artificial intelligence technique of genetic algorithms (GAs) to obtain the optimal signal setting parameters and path flow distribution factor for DUE (dynamic user equilibrium) condition. Papathanasopoulou and Antoniou [2015] proposed a non-parametric data-driven method, locally weighted regression, for the estimation of car-following models, which is suitable for incorporation into microscopic traffic simulation models.

As calibrating a DTA model is relatively time-consuming, the data-driven methods are used to improve the running efficiency and accuracy. Zhang et al. [2016] implemented a modified Kalman filter (EKF) method and tested it on the Singapore expressway network with synthetic data that replicate real world demand level. Huang [2010] applied a Gradient Descent Algorithm (GD) and a Conjugate Gradient Algorithm (CG) as direct optimization formulation for the Brisa A5 motorway.

2.2.2 Traffic Planning: Short-term and Long-term

Various data-driven methods are used in traffic planning, aiming at providing more accurate and efficient planning methods. Existing attempts include k-nearest neigh-
bors, kernel-based approaches, artificial neural networks (ANNs).

Sun et al. [2017] introduced a fully automatic dynamic procedure kNN (DP-kNN) for short-term traffic forecasting and showed that the DP-kNN perform better than the manually adjusted kNN and other benchmarking methods in terms of accuracy on average. Bacciu et al. [2017] analyzed the feasibility of bike-sharing services by using SVM with Gaussian kernels and Random Forests. Gan et al. [2017] developed a novel algorithm to construct an improved multilayer perceptron (MLP) network for accurate long-term ship speed prediction. Lim et al. [2017] proposed a distance-based eco-driving scheme and optimized the scheme by the quadratic programming method for both long-term and short-term.

In general, the data-driven methods are quite useful in transportation research. First, data-driven methods are shown as more efficient and more accurate than traditional analysis methods. In addition, some NP hard problems could be solved by data-driven methods, such as machine learning algorithms.

### 2.2.3 Traffic Features Selection for Data-driven Methods

As informative as DTA models, many traffic features are accessible from the simulation or before the simulation, which helps the construction of data-driven methods. In order to launch a data-driven methods, most of the researchers need to select one or more hyper-parameters for the model. For example, Lim et al. [2017] use characteristic of a drivetrain and road conditions for his distance-based eco-driving scheme.

Other useful features include: maximum desired speed (Papathanasopoulou and Antoniou [2015]), distance, occupancy (or density) (Antoniou et al. [2013]), OD-specific speed-flow curves for trips (Guzman et al. [2016]), flow rate (Sun et al. [2017]) and etc. In some cases, only one features are using. But mostly, multiple features work together in the algorithm.
2.3 Summary of Precedents and Challenges

From the literature in this chapter, we close with two important comments. First, DTA and SODTA models are maturely developed for system optimization problems. However, most of the works mentioned above were applied to short-term optimization problems. In other words, how to efficiently and accurately solve the long-term optimization problem is a challenge, since the long-term forecasting or prediction is time-consuming and less accurate. In the next chapter, we will discuss about how to solve the problem. Second, there were several data-driven algorithms applied the transportation simulation issue. The most popular methods fall into two categories: 1) advanced machine learning algorithms 2) statistical methods. Based on the research need and data availability, different traffic features are selected as parameters in the data-driven models. Based on the two conclusion, we will conduct the following research to examine a full framework of Scenario Analyzer.
Chapter 3

Methodology

This chapter presents the methodologies for the thesis used for the following efforts: First, how a DTA system works with the scenario analyzer and control system is reviewed. Then, each important part of DTA system, i.e. scenario analyzer and control architecture, is introduced in sequence. Lastly, the evaluation metric for selecting the best algorithm for the Scenario Analyzer is summarized.

3.1 DTA System Implementation

In chapter 2, we already have a brief overview of DTA system. In this section, a generic DTA systems structure is proposed. Then, how Scenario Analyzer and SODTA control system works with DTA model are discussed.

3.1.1 Generic DTA System

The general flow of simulation-based DTA systems includes the dynamic traffic management system, demand and supply modules (Figure 3-1). The dynamic traffic management system works as inputs to demand and supply model components, which interact to predict the networkwide traffic conditions. In the demand modules, parameters work for forecasting travels’ route choice. In the supply modules, data sets, including network data, traffic control, incidents and events, evaluating resulting fu-
ture traffic system conditions. The interaction between supply and demand modules is simulated and evaluated inside the DTA model. Then the traffic conditions, such as traffic flows and routing decisions, will be measured at a timely basis.

The principle of DTA system is that: route choice (from demand models) is derived from predictions of future network conditions (from supply models), and the network conditions (from supply models) affect the users' response to information (from demand models). Demand and supply interaction works for traffic assignment and flow propagation on a disaggregate level. As the traffic conditions are provided, real-time DTA systems could extend the work flow to provide operational control of vehicle traffic systems or generate route guidance information for travelers.

3.1.2 Generic System Optimal Dynamic Traffic Assignment (SODTA) Model Implementation

System Optimal Traffic Assignment (SODTA) model is an extension of DTA systems, which could predict traffic state with optimal networkwide performance and provide a
benchmark for controlling and managing dynamic traffic network (Long et al. [2016]). However, the basic simulation-based DTA models can fairly provide an appropriate structure to optimize route choices and control systems. Therefore, the SODTA model need to be considered and added as an additional module to generic DTA systems to help fulfill optimization purposes.

In general, the optimization problems in SODTA problem could be categorized to three major purposes: (1) minimizing travel time for the whole system; (2) minimizing or maximizing other objectives, e.g. total system emission, for the whole system; (3) minimizing both travel time and other objectives for the whole system in an integrated manner.

For the purpose (1), the cost function of a solution \( r \), \( Cost(r) \) is defined as:

\[
Cost(r) = TT(r, q(\tau))
\]  
(3.1)

For the purpose (2), the cost function is generically defined as:

\[
Cost(r) = Other(r, q(\tau), \ldots)
\]  
(3.2)

For the purpose (3), the cost function is generically defined as:

\[
Cost(r) = TT(r, q(\tau)) + Other(r, q(\tau), \ldots)
\]  
(3.3)

Thus, the objective function would be:

\[
SODTA : r^* = \arg\min_r Cost(r)
\]  
(3.4)

where \( q(\tau) \in U \) be the demand vector in scenario \( \tau \), \( r \) be the optimal solution, \( TT(r, q(\tau)) \) be the total system travel time of a solution \( r \) in scenario \( \tau \).

There should also have various constraints of SODTA problem based on the DTA system requirements. When solving the real optimal problem, the constraints are more obvious.
To solve the optimization problem, the objective function takes the network state, the total travel time and other parameters into consideration. Thus, the implementation of SODTA is essential for solving the optimization problem. As shown in Figure 3-1, the generic DTA system consists of input, demand, supply and traffic conditions part, the SODTA system would have a new module: system optimizer added to the framework (Figure 3-2).

As shown in Figure 3-2, the System Optimizer (SO) could optimize either supply or demand modules. With the interaction of supply and demand modules, SO would take both module's output as input parameters and solve the objective function with optimum solution $r$. For example, when optimizing the demand module's route choice behavior, SO would use supply module's outputs, including network state, traffic control and etc., as inputs and acquire the solution $r$. Then, the demand module would use optimal $r$ to simulate optimal route choice and users' response.

Thus, the supply, demand and SO interaction process becomes complete. While the SODTA system is running, the optimal traffic conditions, system optimization problem and optimum solution $r$ will be measured and outputted in a timely manner.

### 3.2 Scenario Analyzer

In the SODTA system, the input data availability is important for solving objective functions of system optimizer. For a short-term planning and forecasting, the input data is easily acquired and the computation efficiency is enough to simulate the "future" traffic conditions. While for long-term planning, the available level of input data required for SODTA system may not be enough (Chiu et al. [2011]). So we need to make some reasonable assumptions and prediction for a long-term period in order to construct the SODTA system.

To help solving the long-term planning optimization problem, a data-driven estimator called Scenario Analyzer is introduced. The objective is to predict, at any current rolling horizon, the parameters or important values over some specific time scenarios. The assumption we make for scenario analyzer algorithm is that, traffic
Dynamic Traffic Management System
ATMS, ATIS, APTS

Demand
Origin-Destination flows
Route choice
Response to information

Supply
Network
Traffic control
Incidents and events

System Optimizer
System optimal problem
Objective function

Supply - Demand - SO Interaction
Optimal Traffic assignment
Optimal solution
Flow propagation

Traffic Conditions
Flows, densities,
speeds, travel time
optimal solution

Figure 3-2: SODTA framework
system conditions are time-dependent and time-correlated.

Based on the assumption, to forecast traffic conditions for a future scenario, one could use the base-scenario state to approximate an initial solution. The main method is using traffic state condition(s) from the beginning of the simulation to identify similar scenario(s) in the achieved database. Then, the matched scenario(s) would be used to represent the parameters or values as the system needs. For a long-term guidance, SODTA systems could use scenario analyzer to approximate future scenario data, and thus providing a new optimal solution from system optimizer.

Therefore, there are two essential procedures to build an efficient scenario analyzer for system optimization problem:

1. first, identify traffic features, for example, total travel time, aggregate traffic flows, emission and etc., as similarity comparison criterion

2. second, finding out the potential methods to match current scenario and historical scenario(s) in the achieved database

From the implementation procedure, we introduce two main parts of scenario analyzer design: (1) matching algorithm and (2) historical database. Similarity features is also essential for scenario analyzer, but for different methods we could consider using the same features, such as network information, traffic condition, travel time etc.

As shown in Figure 3-3, current traffic conditions input to the matching algorithm. Thereafter, matching algorithm read the historical database and in the historical database the corresponding future traffic conditions will be selected back to the SODTA system.

In this next section, the matching algorithm and the database development of scenario analyzer would be precisely discussed.

3.2.1 Matching Algorithm

Matching algorithm of scenario analyzer measures the similarity between current scenario(s) and scenarios in the historical database. We select two modeling methods:
Figure 3-3: Interacting with scenario analyzer
one employing forecasting using only the last time-step feature\(^2\) value and the other using data from multiple previous time steps. Basically, how we treat the data differentiates the two modeling methods.

1. **Single Time Step: clustering and classification**

The first method measure the difference between current simulation scenario’s traffic condition with the historical data. Only scalar values of one of the features or a vector of some features is matched with corresponding values for scenario stored in the historical database. Clustering and classification are tasks that are rather well researched as they have extensive applications in many practical and research fields (Antoniou et al. [2013]).

Some of the most popular heuristics used for clustering are approximate estimation methods for probability models (Fraley and Raftery [2002]). For example, k-means clustering is equivalent to maximizing the multivariate normal likelihood.

Clustering algorithms are usually accompanied by classification methods. There are multiple ways to fulfill classification goal, the most popular one is the k-nearest neighbors (k-nn) algorithm. Also, classification algorithm as support vector machine (SVM) could also be used in this content.

2. **Multiple Time Steps: modeling the evolution of traffic states**

In the second method, multiple time steps of the features are compared. Different from the first method, vector time-series data is matched with historical database.

One possible way is to compare the similarity between different scenarios. Parameters, such as L1-norm\(^3\), L2-norm\(^4\) and KL-divergence\(^5\) could be seen as the similarity reference. For example, using \(q(\tau)\) the traffic feature in scenario \(\tau\) from

\[^2\text{DTA systems are simulated based on rolling horizon approach. In the executing cycle C1, the decision variables for the system are for the prediction sub-intervals. Here the last time-step feature refers to the decision variables from executing cycle C1.}\]

\[^3\text{L1-norm (also known as Sum of Absolute Difference (SAD)) } SAD(x_1, x_2) = \|x_1 - x_2\|_1 = \sum |x_{1i} - x_{2i}|\]

\[^4\text{L2-norm (also known as Sum of Squared Difference (SSD)) } SSD(x_1, x_2) = \|x_1 - x_2\|_2^2 = \sum_i (x_{1i} - x_{2i})^2\]

\[^5\text{KL-divergence } D_{KL}(p||q) = \sum_{t_i} p(t_i) \ln \left( \frac{p(t_i)}{q(t_i)} \right)\]
current simulation, \( p(\tau) \) as the traffic feature from historical database, the similarity parameter is given by,

\[
Similarity(\tau) = \sum_{i=1}^{\tau} similarity(q(i)||p(i))
\]

where \( similarity(q(i)||p(i)) \) measures the similarity between \( q(i) \) and \( p(i) \), \( q(i) \) and \( p(i) \) represents the traffic feature in scenario \( \tau \) from current simulation and historical database

As we quantify the similarity between scenarios, the best matching scenario with largest similarity value or smallest difference value could be chosen. If the dataset is not representative enough, it is possible to take an average of the nearest 5 or more cases.

Another possible method to model the evolution of traffic states is Markov chains, which takes values in a finite categorical space \( X \) of high but finite order \( p \) (\( p \) represents the number of past states on which a future state depends) (Markov [1971]). Using the Markov chain algorithm, the transition probabilities between each state need to be specified as well as the sequence of transition between states.

### 3.2.2 Database Implementation

The second part of the scenario analyzer is the historical database. The historical database provides information to matching algorithm to help quickly lookup and transfer data to the system optimizer. The running efficiency would be highly improved with an efficient historical database structure.

Before building the historical database, we need to figure out the features considered by the scenario analyzer. For traffic simulation purpose, the traffic condition data, such as OD flows in the network, total travel time, total energy consumption, total network emission and etc., should be considered. Also, the supply module data, including special-event and incident information, could be considered. Moreover, if the weather, holiday, weekdays information is available, these data sources should also be included in the database. For a data-driven algorithm as scenario analyzer,
the historical database should be informative.

There are three main tables in the historical database:

1. Master data table. In the table set 1, the basic information of the network, the environmental data (including weather and holiday) and the OD demand data are included.

2. Daily run. Table set 2 consists of the daily simulation results, such as daily run OD flow, daily run parameters, daily run travel time and etc. The daily run table works for storing the current simulation results and parameters. When the simulation finish, the daily run table would be transferred to table set 3. Most important, the features would be used in matching algorithm need to be included in the table set 2.

3. Historical scenario table. Table set 3 includes the historical scenarios features data. The data is updated on time series basis. Similar as table set 2, the features that used in matching algorithm should be included. In addition, table set 3 must present the forecasting target — future scenarios’ traffic data.

The historical database should be connected to the simulation-DTA system. By connecting the database directly with real-time DTA system, the scenario analyzer would interact with the simulation process as both input and output. From the beginning, when the historical database is empty, we need to build the database with some reasonable initial guess about traffic states and related information. After the historical database includes more and more scenarios, the simulation accuracy and efficiency would be largely improved.

3.3 Evaluation Metric

The objective of scenario analyzer is to obtain the future traffic conditions so that the forecasting results are close to the observed ones. In order to evaluate the perfor-

---

\textsuperscript{6}Daily run refers to the recent simulation process.

\textsuperscript{7}The historical scenario table is stored in the same table, but with different label to recognize the differences between these scenarios.
mances of the developed models, we use the mean absolute percentage error (MAPE) as loss function. The closer the value is to zero, the better the forecasting performance is. The index is given by:

\[
\text{MAPE} = \frac{100}{\tau} \sum_{i} \frac{|A_i - F_i|}{A_i} \% \tag{3.6}
\]

where \( n \) is the number of testing values, \( A_i \) and \( F_i \) denote the actual and forecast value at time step \( i \), respectively.

The MAPE is one of the most popular evaluation metric for prediction and forecasting. In machine learning area, researchers use MAPE to present the error for prediction.

3.4 Summary

In this chapter, the general DTA system framework was first reviewed. Then, the SODTA was introduced and added to the generic DTA system to help with the real-time simulation and optimization purpose. Further, the scenario analyzer design and workflow was discussed in detail. Finally, the control architecture and SODTA system with scenario analyzer were completed after fully understanding each part of the system. In the next chapter, the case study of the integrated system would be presented.
Chapter 4

Case Study: Incentives for Energy Efficiency in the Boston CBD area

In this chapter, a case study is conducted on the Boston Central Business District (CBD) area to demonstrate the utility and performance of scenario analyzer that proposed in Chapter 3. First, the DTA model and SODTA model used in this study are introduced. Then, the Boston CBD network and traffic demand resource is briefly summarized. Finally, the beyond-horizon scenario analyzer is tested on the network, along with results and discussion.

4.1 Model Overview

4.1.1 DTA System: DynaMIT

In this study, DynaMIT (Dynamic Network Assignment for the Management of Information to Travelers) is used to estimate and predict traffic conditions. DynaMIT is a simulation-based DTA system. The DynaMIT work flow is shown in Figure 4-1, which is composed of state estimation (SE) and state prediction (SP):

The state estimation (SE) module makes use of the historical data and real-time observations to calibrate the demand and supply parameters, which gives a more reliable and accurate estimation of current state. Also, online calibration can be
switched on or off depending on the application. From state estimation, the estimated traffic network state and parameters would be used as inputs for state prediction.

The state prediction (SP) module predicts future traffic state based on the estimated network state. The ultimate goal is to provide real-time predictions on how users respond to provided information and incentives. Since the state prediction is restricted to a specific rolling horizon, usually 15 minutes, it is crucial to gain accurate energy estimation for longer rolling horizon, thus forms the topic of the paper.

In real time applications, DynaMIT operates on a rolling horizon where estimation
and prediction intervals are performed successively. The estimation horizon is usually 5 minutes. The prediction horizon can be specified depending on the application. Generally, DynaMIT is capable of modeling large networks with proper simulation efficiency. In real-time traffic management, DynaMIT could work accurately and effectively.

4.1.2 SODTA System: Tripod Overview

In this study, Tripod (sustainable travel incentives with prediction, optimization and personalization) works as the control architecture and integrates with DynaMIT. Tripod is a smartphone-based system to influence individual real-time travel decisions by offering information and incentives to optimize system-wide energy performance (Azevedo et al. [2018]).

![Tripod Framework](image)

Figure 4-2: Tripod Framework (Azevedo et al. [2018])

From users standpoint, travelers are offered incentives in the form of tokens for a variety of energy-reducing travel options, including route choice, travel mode, departure...
ture time, driving style and trip making. From operation viewpoint, the network and
users choices influence the system performance. From both users and system opera-
tion point of view, the network-wide supply and demand interactions as well as users
travel preference need to be considered in the Tripod control architecture. Thus, the
tripod model is decomposed into two main problems: the System Optimization (SO)
and the User Experience (UE) (Figure 4-2).

**System Optimization (SO)** is the SODTA model of Tripod. For the Tripod
SODTA system, the ultimate goal is to save more energy with limited amount of
incentives. To fulfill the ultimate goal, there are five steps: (1) estimates the current
state of the network; (2) predict the network state given different token assignment
strategies; (3) estimate the energy saving given different token assignment strategies;
(4) system optimizer: optimize the token awarding strategy; and (5) providing token
energy efficiency value (i.e. energy savings per token) to the User Experience (UE).

As shown in Figure 4-3, the system optimizer is integrated with demand simulator,
supply simulator, historical database and system optimizer inside DynaMIT. **System
Optimizer** is the extension module of DynaMIT, which works for optimization prob-
lem. The system optimizer achieved the expected energy savings and expected token
usage for the prediction period from **state prediction**. For the energy savings be-
yond the rolling horizon, the historical database of **scenario analyzer** will calculate
that.

Thereafter, **system optimizer** given each possible token energy value and state
conditions to calculate the energy consumption for rolling period from DynaMIT
dependently. Based on the result, the optimal value are selected as the input for
**state prediction**’s demand simulator. Then, the demand simulator simulates user
interaction and provide guidance for user route choice. For each rolling period, the
optimal token energy value is selected and the minimum energy consumption value
is recorded to the output file and historical database.

In this chapter, how scenario analyzer works with system optimizer as well as
Tripod and how to design the beyond-horizon scenario analyzer would be discussed
in detail.
4.2 Overview of Boston CBD

4.2.1 Supply: Boston CBD Network

The case study was conducted on the Boston central business district (CBD) network. The Boston CBD area (shown in orange color in Figure 4-4) is the location of many residents, corporates, city facilities and many Boston's tourist attractions. In recent years, Boston CBD area has undergone a transformation that included the construction of new buildings, renovation of historical buildings and arrival of new residents and businesses.

The lines with gray color represent roads in different categories in CBD area. The CBD network is extracted and modeled in DynaMIT. The simulated network includes contains detailed information of the road links, segments, nodes, lanes and lane connectors. Further, the location, length, curvature, road category, default speed

---

8The data comes from the CTPS model.
limits, default travel time and capacity are specified in the network file.

The CBD network includes 1746 links and 846 nodes. Each link consists of one or more segments and each segments consists of multiple lanes. Nodes connect different links and serve as loaders where vehicles are added to the network. The simulated CBD network is presented in Figure 4-5. The green dots represent the node and the blue lines represent the links.

4.2.2 Demand: Origin-Destination flows

On the demand side, we use the morning peak (6:30 AM to 9:00 AM) OD counts information from Central Transportation Planning Staff (CTPS). In each 5 minute estimation interval, there are 13080 Origin-Destination (OD) pairs. The total number of OD flows in the morning peak is 115448.

As shown in Figure 4-6, the demand stayed flat around 2200 flows per five minutes at the beginning of morning peak. Then, after 7:00 AM, the demand highly increased
to 3800 cars every time interval, and decreased to 3200 after 8:00 AM. The demand distribution shows the same trend as we expected.

4.3 Scenario Analyzer for Boston CBD Network

The objective for Tripod system is minimizing energy consumption, in other words maximizing energy savings for the whole Boston CBD network and for the whole day. From DynaMIT, we could acquire that for each rolling prediction horizon, how much energy is consumed and how people react to Tripod. As the token assigned to the users could be used now or later, when we try to minimizing the total energy consumption, the energy consumption beyond the prediction horizon of DynaMIT until the end of the day is missing. By designing the scenario analyzer for Boston CBD network, we want to design a system to calculate the beyond-horizon energy consumptions accurately and efficiently.

In section 3-3, a generic scenario analyzer design is discussed. In this section,
a specific scenario analyzer which works for estimating beyond-horizon energy consumption is presented. First, the selection of traffic similarity features is discussed. Then, the matching algorithm and database implementation is presented.

4.3.1 Traffic Features Selection

From DynaMIT-Tripod simulation, several traffic features, including travel time, energy consumption, OD counts, token usage etc., are available for each prediction period. Let \( t_0 \) represents the starting time of simulation, \( t \) represents the current time, \( H\delta \) as the rolling period for DynaMIT estimations (usually 5 minutes) and \( H\Delta \) as the rolling period for DynaMIT predictions (usually 15 minutes). Thus, the state estimation horizon is from \([t - H\delta, t]\) and state prediction horizon is from \([t, t + H\Delta]\).

For the design, we use five different features as following:

(a) Cumulative energy consumption: \( E(t) = Ec(t_0, t + H\Delta) = \sum_{t_0}^t Ec(t) + \bar{Ec}(t) \)

(b) Energy consumption in the last period of rolling horizon: \( SE(t) = Ec(t) + (H - \bar{Ec}) \)
1) \( \Delta, t + H\Delta ) = E(t) - E(t - H\delta) \)

where \( Ec(t) \) denotes the estimated energy consumption in the current state estimation horizon \([t - H\delta, t]\) and \( Ec(t) \) denotes the predicted energy consumption in the current state prediction horizon \([t, t + H\Delta]\)

(c) Cumulative OD flows: \( Q(t) = q(t_0, t + H\Delta) = \sum_{t_0}^t q(t) + \tilde{q}(t) \)

(d) OD flows in the last period of the rolling horizon: \( SQ(t) = q(t + (H - 1)\Delta, t + H\Delta) = Q(t) - Q(t - H\delta) \)

where \( q(t) \) represents the number of vehicles on the network in the current state estimation horizon \([t - H\delta, t]\) and \( \tilde{q}(t) \) represents the number of vehicles on the network in the current predicted horizon

(e) Remaining tokens at the end of the rolling horizon: \( W(t, e) = B(t) - T(t, e) \).

\( T(t, e) \) denote the number of tokens consumed in the current state prediction horizon \([t, t + H\Delta]\) and \( B(t) \) denotes the number of tokens available at time \( t \).

In total five features (a) to (e) are selected for scenario analyzer. First, we select energy consumption as a base feature, then using cumulative value and stepwise value to present the traffic feature. In addition, for the number of vehicles, we consider both cumulative values and stepwise values. Finally, the benchmark, i.e. remaining tokens, for the objective function is selected as one of the features. Each features would be tested in the scenario analyzer, the performance and results will be discussed in Section 4.4.

### 4.3.2 Matching Algorithm

After identifying traffic features, the potential matching algorithms for beyond-horizon scenario analyzer are demonstrated. Following the methodology in section 3.5, we would use two approaches to match current traffic scenarios with the scenarios in historical database. The first approach, i.e. k-nearest neighbors (k-NN), measures the difference between current scenario and the historical data. The second approach is
Kullback-Leibler divergence, using multiple time steps of the features to select target value.

1. **K-nearest neighbors (k-NN)**

K-nearest neighbors (k-NN) is the most useful supervised learning algorithm, which is often used for classification and regression problems. In this study, k-NN method would use the scalar values of one of the features (a) to (e) to match with the values for days stored in the historical database.

For a general k-NN algorithm, the principle is to classify the class or estimate the target value for inputs based on the distance function. In k-NN classification, an object is classified by a majority vote of its neighbors, with the object being assigned to the class most common among its k nearest neighbors (Figure 4-7). In k-NN regression, the value of an object is the average of the values of its k nearest neighbors.

![K-Nearest Neighbors Classification](image)

**Figure 4-7: Example of k-NN classification (Bronshtein [2017])**

---

9The test sample (green circle) should be classified either to the first class of blue squares or to the second class of red triangles. If k = 3, it is assigned to the second class. If k = 5, it is assigned to the first class.

48
As for scenario analyzer for beyond-horizon energy consumption, we consider matching the features (a) to (e) in historical database with the current simulation within a specified range of error rate. In other words, only the neighbors within the allowable range is considered as "k-nearest" neighbors. The k-value is not specified as a constant value in the scenario analyzer. Instead, the k-value differs from scenarios but the "distance" remains in the range.

In our case, the historical days that have the corresponding feature values within ±5% of the current value are chosen as similar days. If scenario analyzer with some features could not find match within ±5% error range, we will enlarge the range until finding at least one match. Same as for k-NN regression, an average of the beyond-horizon energy consumption in these chosen historical days is reported as the estimate for the current day.

2. Kullback-Leibler divergence (KL divergence or KLD)

The second approach we use is Kullback-Leibler divergence (also called relative entropy). KL divergence is a measure of how one probability distribution diverges from a second, expected probability distribution. Different from k-NN algorithm, KL divergence method would use the complete time series of the features to match the days in the archived database.

From definition, Kullback-Leibler divergence is calculated by:

\[
D_{KL}(p||q) = \sum_{t_0}^{t_i} p(t_i) \ln \left( \frac{p(t_i)}{q(t_i)} \right) 
\]

where \( p(t) \) is the feature values of current simulation, \( q(t) \) is the corresponding features of scenarios in the historical database.

As the Kullback-Leibler divergence function considered \( p(t) \) and \( q(t) \) as two probability distribution, we need to normalize the time series values. For example, we select energy consumption in the last period of rolling period as the traffic feature. The rolling period is 5 minutes and prediction horizon is 15 minutes. For Boston CBD morning peak (6:30 AM to 9:00 AM) test, the time-series for stepwise en-
ergy consumption is given by \( \{SE(6:35), SE(6:40), SE(6:45), \cdots \} \), also known as \( \{Ec(6:45 \text{ to } 6:50), Ec(6:50 \text{ to } 6:55), Ec(6:55 \text{ to } 7:00), \cdots \} \). As shown in Figure 4-8, the green line is the current simulation's stepwise energy consumption and the blue line represents one scenario's stepwise energy consumption in the historical database. The blue line and the green line shows similar fluctuation trend. After normalizing the stepwise energy values, the KL divergence value could be calculated.

For each simulation, the \( D_{KL} \) would be calculated in a timely manner and differentiate between different historical days. The time series in the historical database that has the minimum \( D_{KL} \) with the current day is chosen. However, at the beginning of the simulation, the current simulation only has one feature value, after normalization, all the \( D_{KL} \) values are the same, i.e. equal to 0. In order to solve that, an average of all historical beyond-horizon energy consumptions is selected as the estimate for the first step. For the first time step, the prediction for beyond-horizon energy consumption may be wrong. However, after the number of data points increase, the KLD method would generate the time-series observations and become more accurate.

### 4.3.3 Database Design

The scenario analyzer for beyond-horizon energy estimation is tested on Boston CBD network with the CTPS morning peak demand (6:30 AM to 9:00 AM). The network and demand file is described in Section 4.2. In this section, the database design and the method for collecting data is briefly summarized.

#### 1. Database design

The database of scenario analyzer works for storing daily run and historical values. Also, the database could provide possible functionalities for further data analysis, i.e. similarity matching. Here, we would show how we construct the database for beyond-horizon energy scenario analyzer. The design document is appended in Appendix A.
Figure 4-8: Energy consumption for two similar days (existing KL-divergence sample)

(1) Master data table

Table 4.1 - 4.3 includes the basic information of the simulation. There are three tables in the Table set 4. Table 4.1 denotes the simulation time interval and the corresponding interval number. For the morning peak simulation, the time number ranges from 1 to 27, where 1 represents 06:35AM and 27 represents 08:45AM. For example, Table 4.1: Master Data is given by,

Table 4.1: Master Data: basic information of simulation

<table>
<thead>
<tr>
<th>Interval Start Time</th>
<th>Interval End Time</th>
<th>Time Number</th>
</tr>
</thead>
</table>

Table 4.2 denotes the OD pair ID and its corresponding origin and destination nodes. For example:

In Table 4.3, the network for the current simulation is pointed out. To store basic information about the network, we includes number of nodes, links, segment, lanes and etc in the table. For example, for the Boston CBD network, the table 4.3:Master
Table 4.2: Master Data: OD pair ID and its corresponding origin and destination

<table>
<thead>
<tr>
<th>OD Pair ID</th>
<th>Origin Node</th>
<th>Dest Node</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>59295</td>
<td>45696</td>
</tr>
<tr>
<td>2</td>
<td>58882</td>
<td>59136</td>
</tr>
</tbody>
</table>

Data looks like:

Table 4.3: Master Data: network information

<table>
<thead>
<tr>
<th>Network Name</th>
<th>Nodes</th>
<th>Links</th>
<th>Segments</th>
<th>Lanes</th>
<th>Connectors</th>
<th>Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boston CBD</td>
<td>846</td>
<td>1746</td>
<td>3085</td>
<td>5057</td>
<td>5232</td>
<td>0</td>
</tr>
</tbody>
</table>

(2) Daily run table

Table 4.4 includes the intermediate data from daily simulation. We separate the daily run data into basic info, state estimation and state prediction tables. In Table 4.4, the primary key is the simulation date, simulation start time, end time. Other information that describe characteristics of the current day (For instance, weather, incidents, special events, etc) are included.

Basically, the Table 4.4 for state estimation or prediction for state prediction includes all the features that the scenario analyzer would use. The feature OD flows could be achieved from the number of records in the table. For example:

Table 4.4: Daily run: state estimation (or prediction) table

<table>
<thead>
<tr>
<th>Trip ID</th>
<th>Current Time</th>
<th>Departure Time</th>
<th>Arrival Time</th>
<th>Arrival</th>
<th>Energy Consumption</th>
<th>Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>1</td>
<td>6:30:00</td>
<td>6:35:00</td>
<td>0</td>
<td>9102</td>
<td>0</td>
</tr>
<tr>
<td>2</td>
<td>1</td>
<td>6:30:31</td>
<td>6:32:19</td>
<td>1</td>
<td>584</td>
<td>3</td>
</tr>
</tbody>
</table>

First, all the information of current simulation is stored as an intermediate table. As it is faster to calculate the feature values. After the whole simulation finish, the results would be uploaded to the historical database.
(3) **Historical scenario table**

The last part of database is the historical table, the data of historical days is stored in the aggregate level. Different from daily run table, the historical table has a primary key points to simulation time.

In Table 4.5, for each time interval, the demand, energy consumption, token usage and the other features are recorded. For example:

<table>
<thead>
<tr>
<th>Scenario</th>
<th>Current T</th>
<th>Prediction T</th>
<th>Label</th>
<th>cumDemand</th>
<th>stepDemand</th>
<th>cumEnergy</th>
<th>stepEnergy</th>
<th>Token</th>
<th>BHEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>6:35</td>
<td>6:50</td>
<td>1000U125</td>
<td>10138</td>
<td>2536</td>
<td>57785500</td>
<td>15630417</td>
<td>26245</td>
<td>761313177</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>12659</td>
<td>...</td>
<td>...</td>
<td>73383177</td>
<td>15597677</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>27</td>
<td>8:45</td>
<td>9:00</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>6:35</td>
<td>6:50</td>
<td>1500U125</td>
<td>10140</td>
<td>2536</td>
<td>57660322</td>
<td>15595291</td>
<td>40516</td>
<td>756890630</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

The scenario is the ID for current time, which would help to identify similar scenario number when using KL divergence matching algorithm. Also, the label represents the demand scale and the 5 minutes token budget. Other forms of label is also acceptable.

The Table 4.6 is the validation sheet, which includes the predicted value and actual beyond-horizon energy consumption. For the testing purpose, we also includes the predicted feature value and actual feature value in Table 4.5.

2. **Data set**

For testing the performance of scenario analyzer, a historical data set and test data set are specified. For each test data set, the same historical database is utilized.

<table>
<thead>
<tr>
<th>Current Time</th>
<th>Predicted Feature</th>
<th>Predicted BHEC</th>
<th>Actual Feature</th>
<th>Actual BHEC</th>
</tr>
</thead>
<tbody>
<tr>
<td>6:35</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>6:40</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>End</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
Based on the difference between test data set, we could examine the performance and summarize the findings from the results.

(1) **Historical data set**

The historical database contains various different traffic states, which simulates the traffic from different types of days. The base demand was taken to be CTPS demand for the morning peak, i.e. 6:30 AM to 9:00 AM. For artificially different types of historical days, the base demand was modified by scaling the demand on all OD pairs using a scale factor, which adds the randomness to the Origin-Destination flows.

The scale factor is given from the following uniform and normal distribution:

1) Uniform distribution with scale factor of 1.00 (base case)
2) Uniform distribution with scale factor of 1.25 (scaled up)
3) Uniform distribution with scale factor of 0.75 (scaled down)
4) Normal distribution with mean of 1.00 and standard deviation of 0.25 (scaled up and down to different extend for different OD pairs)
5) Normal distribution with mean of 1.00 and standard deviation of 0.50 (scaled up and down to different extend for different OD pairs)

The five different scale factor generators provide five different demand files for the database. Historical data is generated by running Tripod-DynaMIT with each new demand file and fixed token budgets of: 1) 1000, 2) 1500 3) 2000 tokens per time period (5 minutes). Thus, with the five demand files, three token budget thresholds and two more tests with different random seed values, the historical database includes 17 scenarios for using.

(2) **Train data set**

Similar as cross-validation, we launch another run of one of the demand profiles already in the historical data, match it with the other scenarios in the database and validate the matching algorithm.
This is a verification step to ensure that whether the estimator is able to find exact or most similar match as we proposed. The train data set is only a statistical variation on the existing historical data, i.e. the difference is bounded by the statistical variation of Tripod prediction.

After iterating all the scenarios in the demand profiles, the average performance is summarized according to the equation 3.6. The train data set could give an insight that when we have a small statistical variation, how well the scenario analyzer would perform.

(2) Test data set

After ensuring the scenario analyzer. To test the performance of the beyond-horizon scenario analyzer, three test scenarios are generated using the same base demand but with different scale factors:

1) Normal distribution with mean 1.0 and standard deviation of 0.1

2) Uniform distribution with scale factor of 0.5 (scaled down)

3) Uniform distribution with scale factor of 1.5 (scaled up)

Test set 1) uses a relative small standard deviation. Although there is no exact match to any of the historical days, the demand profile still has a representative match in the archived historical database since the mean demand is the same. Test set 2) results in a much lower demand, which has no representative match in the historical dataset. Test set 3) results in a much higher demand, which has no representative match in the historical database either. Thereafter, we expect no match results would exist for the test set 2) and test set 3).

The results for train set and test set are summarized and concluded in section 4.4. By comparing the results for two set of data, we could find the properties, such as advantages and disadvantages, of scenario analyzer.
4.3.4 Scenario Analyzer’s Interaction with Tripod-DynaMIT

As Figure 3-3 shown, the generic scenario analyzer interacts with the system optimizer and Supply-Demand-SO interaction module. In Figure 4-9, the scenario analyzer for beyond-horizon with Tripod-DynaMIT project is shown.

Inside Tripod-DynaMIT, the state estimation and state prediction provides the traffic features for scenario analyzer. Then, the scenario analyzer would find matching scenarios in the historical database by the matching algorithm. Afterwards, the target value, beyond-horizon energy consumption is estimated by beyond-horizon scenario analyzer. Finally, the system optimizer minimize the total energy consumption and select the optimal token efficiency value $\tau^*$ back to state prediction. State prediction would reanalyze the traveler behavior, route choice and predict the future traffic conditions.

For each rolling horizon, the interaction between Tripod-DynaMIT and scenario analyzer follows the process until the full simulation ends. For different network and different simulation purpose, the historical database need to store corresponding scenarios for selecting.

4.4 Results

In this section, the performance of scenario analyzer on train and test data sets are reported and compared with Tripod-DynaMIT simulation. The analysis is conducted based on prediction error rate (MAPE evaluation metric).

4.4.1 Performance of Beyond-Horizon Energy Estimator

The performance of two matching algorithms (k-NN and KLD) on train and test data sets are discussed in this section. The train data set includes 17 historical scenarios, the test data sets includes 3 profiles.
1. k-NN performance

The k-NN method is configured with 5% error rate range. As stated in Section 4-4, the simulation time is from 06:30 AM to 09:00 AM. Then the k-NN matching algorithm is used to estimate for the beyond-horizon energy consumption.

The results are shown in Table 4.7. The table reports error rate as MAPE (mean absolute percentage error) in the beyond-horizon energy prediction. From Table 4.7, several observations can be made.

First, both demand and energy consumption appear to be features well suited for the scenario analyzer design, no matter cumulative values or stepwise ones. The smallest error rate is 0.59%, even smaller than 1%. The cumulative features (energy
consumption or demand) serve as better indicators than the instantaneous values. As expected, the cumulative features have information of the past that the instantaneous values do not. In addition, the energy consumption is more informative than demand and remaining tokens, as it indicates various traffic conditions, for instance, traffic flows, travel time and etc. The remaining tokens is not a good indicator from the results. One possible reason is that the system users are not enough sensitive to the tokens assigned.

Second, as expected, the train set that could find an exact or the most similar match performs very well (Figure 4-10(a)). Although test set 1 does not have an exactly similar scenario, scenario analyzer also performs well with minimum error less than 2% and average error rate around 6% among all features. It indicates that as long as the main demand profile is the same as in achieved data, a small perturbation on demand would not affect the performance of scenario analyzer. For example, the test set using scale factors from normal distribution with mean 1.0 and standard deviation of 0.1 is most similar to the uniform distribution of scale factor 1.0 in the historical database. However, on the other hand, with unexpected profile, such as test set 2 and test set 3, beyond-horizon scenario analyzer reports an extremely large error. One notable thing is that with the configured 5% error range, test set 2 and 3 couldn’t find a possible match. Thus, we could find out the limitation of k-NN approach.

Table 4.7: Prediction errors for beyond-horizon energy consumption using the k-NN matching algorithm

<table>
<thead>
<tr>
<th>Features</th>
<th>CumEC</th>
<th>StepEC</th>
<th>CumDemand</th>
<th>StepDemand</th>
<th>Remaining Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train set</td>
<td>0.59%</td>
<td>1.35%</td>
<td>1.86%</td>
<td>1.57%</td>
<td>25.99%</td>
</tr>
<tr>
<td>Test set 1</td>
<td>1.85%</td>
<td>2.66%</td>
<td>4.47%</td>
<td>4.11%</td>
<td>17.06%</td>
</tr>
<tr>
<td>Test set 2</td>
<td>43.90%</td>
<td>43.33%</td>
<td>43.90%</td>
<td>43.90%</td>
<td>109.97%</td>
</tr>
<tr>
<td>Test set 3</td>
<td>48.53%</td>
<td>33.48%</td>
<td>50.27%</td>
<td>33.48%</td>
<td>33.48%</td>
</tr>
</tbody>
</table>
2. KL divergence performance

Similar as k-NN method, the KL divergence matching approach is used to predict the beyond-horizon energy consumption for 06:30 AM to 09:00 AM simulation. The first time step, the KLD (KL divergence) is configured with average value of all historical days. The results are shown in Table 4.8. MAPEs in the beyond-horizon scenario analyzer are reported in the table. Several observations can be made from these results.

Table 4.8: Prediction errors for beyond-horizon energy consumption using the KL divergence matching algorithm

<table>
<thead>
<tr>
<th>Features</th>
<th>Cum EC</th>
<th>Step EC</th>
<th>Cum Demand</th>
<th>Step Demand</th>
<th>Remaining Token</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train set</td>
<td>1.47%</td>
<td>2.11%</td>
<td>1.42%</td>
<td>1.59%</td>
<td>27.19%</td>
</tr>
<tr>
<td>Test set 1</td>
<td>4.41%</td>
<td>29.68%</td>
<td>8.21%</td>
<td>57.48%</td>
<td>26.89%</td>
</tr>
<tr>
<td>Test set 2</td>
<td>202.87%</td>
<td>228.77%</td>
<td>63.06%</td>
<td>126.86%</td>
<td>98.04%</td>
</tr>
<tr>
<td>Test set 3</td>
<td>37.39%</td>
<td>36.53%</td>
<td>34.98%</td>
<td>35.48%</td>
<td>46.08%</td>
</tr>
</tbody>
</table>

First, only cumulative demand and cumulative energy consumption are good features for beyond-horizon scenario analyzer. Different from k-NN method, the stepwise indicators report a relative high error rate, which exceed the acceptable range.

Second, the train set and test set 1 still have a good performance with minimum less than 1.5% prediction error and maximum less than 10% error rate. When an exact match or similar match is not available, KL divergence algorithm becomes unreliable.

Third, among the two methods, k-NN algorithm performs marginally better. One possible reason is that KL divergence method work as time-series solution, which need more than one data points to form. In the early part of the simulation, where the current time series is not well-developed, the KL divergence method could not prove its advantages. As shown in Figure 4-10(b), when an exact match is available, the time series need only two time steps to formulate. However, when exact match is not available, the prediction beyond-horizon energy consumption is approaching the actual value after step 5 (Figure 4-10(d)).
4.4.2 Comparison with DynaMIT

1. Advantages

First of all, the scenario analyzer helps to extend horizon for SODTA’s optimization problem. In the optimization function, not only current situation is considered, but also the future horizon is included. Thus, the optimization problem is improved, we would achieve a more accurate and reasonable optimal solution.

Second, using scenario analyzer would solve the long-term planning problem. The long running time for 24 hour traffic simulation in DTA system is a bottleneck for DTA system development. Thus, scenario analyzer is quite helpful to avoid full simulation and get the future scenario. In addition, when the input data is not available for long-term planning, scenario analyzer with further implementation could help.
2. Disadvantages

The scenario analyzer highly depends on the historical database. With limited number of scenarios in the historical database, the prediction accuracy is still limited on the unseen days. Although the method and results are promising, further tests are needed for more conclusive outcomes.

4.5 Summary

In this chapter, a case study based on Boston CBD network was presented. The main DTA system, SODTA control architecture and research area is introduced. Then, the scenario analyzer for beyond-horizon energy consumption prediction were demonstrated in detail. It is followed by the performance of k-NN method and KL divergence method. It was concluded that k-NN algorithm performs marginally better than the KLD algorithm. In addition, the cumulative energy consumption was the best traffic features for scenario analyzer. The results for the beyond-horizon scenario analyzer were discussed and some tables and figures were given to support this.
Chapter 5

Conclusion and Discussion

5.1 Summary

Inefficiency in transportation systems causes waste of energy, pollution, congestions and delays. Traffic planning and management aims to optimize the transportation facilities usage, thus achieve system-wide efficient and energy saving. In order to fulfill transportation management goal, the real-time dynamic traffic assignment (DTA) systems are integrated to the Traffic Management Center (TMC) and widely used for traffic simulation. While achieving the global or local optima, the DTA models are extended to system optimal dynamic traffic assignment (SODTA) models.

However, the SODTA models highly depends on the DTA models, whose information is restricted to a specific prediction horizon. To be specific, DTA systems receive the inputs and parameters from real world or virtual network, then the demand module interact with the supply module and provide the output as traffic conditions and routing decisions. Usually, the prediction for traffic conditions is accurate in 15 to 30 minutes horizon. Thus, it is critical to achieve information outside the prediction horizon for long-term planning and management. In this thesis, the goal is to develop a scenario analyzer works for extend the prediction horizon and help achieve traffic information for a longer period.

In general, the proposed scenario analyzer is integrated with SODTA models and helps with the optimization problem. Inspired by the innovated data-driven methods
for transportation research, the scenario analyzer is equipped with an archived historical database and matching algorithms. Due to the time-dependent characteristic of traffic system, the future scenario highly depends on the current prediction horizon. Thus, matching algorithms which help to find the similarity situations in the historical database is an essential part of scenario analyzer. The matching algorithm could be separated to two modeling thoughts: 1) single time step modeling and 2) time-series modeling. Under each category, various methods could be used. In addition, the selection of traffic features is important to scenario analyzer.

Finally, a case study with DynaMIT and Tripod for Boston control business district (CBD) network during morning peak is conducted. The scenario analyzer and Tripod system optimizer integration framework is proposed, where scenario analyzer predict for the beyond-horizon energy consumption and the system optimizer deal with finding the optimal solution for Tripod. In order to make the test persuasive, a real network and demand from Central Transportation Planning Staff (CTPS) is used. There are 1746 links and 846 nodes in the network, 13080 OD pairs in the demand. The simulation period is 6:30 AM to 9:00 AM, and the traffic is estimated every 5 minutes. The prediction horizon is 15 minutes, and ideally the scenario analyzer predict the energy consumption after 15 minutes until the end of the simulation period. The matching algorithms for testing is k-NN and KL divergence. In addition, the database is split to train data set and test data set.

Results from case study show that scenario analyzer performs extremely well with small error rate. As for the matching algorithms, k-NN algorithm performs marginally better than the KLD method as the KLD method need more time steps for training. As for the traffic features selection, different algorithms apparently have different preference. Still, in general, cumulative demand and cumulative energy consumption work for both algorithms. As for the historical data sets, scenario analyzer shows its limitation as high error rate for new traffic demand. All in all, the results indicate that scenario analyzer is practical in SODTA framework and thus could be improved and used for transportation planning with optimization purpose.
5.2 Future Research Directions

1. More matching algorithm testing

In the case study, we proposed k-NN and KL divergence algorithms. Due to the empirical results, the error rate is relatively small. However, the two algorithms perform badly for the unexpected demand. There are more matching algorithms, for example, neural network, SVR to test. It is interesting to see if we could find a more accurate matching algorithm that works for all the test sets.

2. More scenarios testing

From the results, k-NN performs marginally better than the KLD method. As KLD method need more time steps to train, the results may change if we use a longer simulation time, for example, 24 hour simulation. Also, the case study only shows the results for morning peak. It is also beneficial to perform more case studies accounting for more scenarios, preferably with longer simulation period, larger network and more traffic demand. In addition, with more scenarios being tested, the historical database would be more informative and improve the performance of scenario analyzer.

3. Computation performance

Currently, the computation performance of scenario analyzer is roughly examined. If we want to launch test on a larger network with longer simulation time, the profiling of computation process is essential for real-time traffic management. Also, the historical database is still small. It is interesting to construct a larger historical database and test the computation performance.

4. Sensitivity testing

In the k-NN matching algorithm, only ±5% similarity range is used for testing on CBD network. We expect that the similarity range would influence the performance of scenario analyzer significantly. Also, the results show that some features need larger similarity range than others. It is useful to figure out the relation between
features and similarity range. In addition, it is worthwhile to launch the sensitivity analysis on the similarity range and prediction’s precise level.

5. Other applications

Now, the scenario analyzer is working in energy optimization for beyond-horizon. Theoretically, scenario analyzer could help extend the prediction horizon for general DTA systems and work for long-term traffic forecasting. If the forecasting is quite well and the scenario analyzer works well with large network, the simulation time of DTA systems would be largely reduced.
Appendix A

Database Design Document

Objectives

1. Construct a database that stores daily run of state prediction and system optimizer for Tripod;

2. Also store and update the historical values based on state prediction and system optimizer results;

3. Provide some functionalities for further data analysis;

Assumptions

1. Only one latest prediction result for each day is stored in the daily run database;

2. Assuming 5 minutes estimation interval length and 15 minutes prediction interval length;

Table A.1: Master Data: basic information of simulation

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>Interval Start Time</td>
<td>Start time of DynaMIT interval</td>
<td>time</td>
</tr>
<tr>
<td>Interval End Time</td>
<td>End time of DynaMIT interval</td>
<td>time</td>
</tr>
<tr>
<td>Time Number</td>
<td>Corresponding number ranging from 1 (06:35) to 27 (08:45), if in 5 minutes</td>
<td>integer</td>
</tr>
</tbody>
</table>
Table A.2: Master Data: OD pair ID and its corresponding origin and destination

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>OD Pair ID</td>
<td>Number ranging from 1 to number of ODs</td>
<td>integer</td>
</tr>
<tr>
<td>Origin Node</td>
<td>Origin node ID in DynaMIT network file for a specific</td>
<td>integer</td>
</tr>
<tr>
<td>Dest Node</td>
<td>OD pair</td>
<td>integer</td>
</tr>
</tbody>
</table>

Table A.3: Master Data: network information

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK: Network</td>
<td>Name of the network</td>
<td>character varying(100)</td>
</tr>
<tr>
<td>Nodes</td>
<td>Number of nodes</td>
<td>integer</td>
</tr>
<tr>
<td>Links</td>
<td>Number of links</td>
<td>integer</td>
</tr>
<tr>
<td>Segments</td>
<td>Number of segments</td>
<td>integer</td>
</tr>
<tr>
<td>Lanes</td>
<td>Number of lanes</td>
<td>integer</td>
</tr>
<tr>
<td>Connectors</td>
<td>Number of connectors</td>
<td>integer</td>
</tr>
<tr>
<td>Sensors</td>
<td>Number of sensors</td>
<td>integer</td>
</tr>
</tbody>
</table>

Table A.4: Daily run: metadata for daily run of DynaMIT

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK: Simulation Date</td>
<td>Date of the traffic scenario</td>
<td>date</td>
</tr>
<tr>
<td>PK: Simulation Start Time</td>
<td>Start time of this DynaMIT run</td>
<td>time</td>
</tr>
<tr>
<td>PK: Simulation End Time</td>
<td>End time of this DynaMIT run</td>
<td>time</td>
</tr>
<tr>
<td>Estimation Interval</td>
<td>The length of each estimation interval</td>
<td>integer</td>
</tr>
<tr>
<td>Prediction Interval</td>
<td>The length of each prediction interval</td>
<td>integer</td>
</tr>
<tr>
<td>Weather</td>
<td>Weather conditions of the day in calibration</td>
<td>character varying(100)</td>
</tr>
<tr>
<td>Incidents</td>
<td>Incident records of the day in calibration</td>
<td>character varying(100)</td>
</tr>
<tr>
<td>isholiday</td>
<td>Whether the current simulation day is holiday</td>
<td>boolean</td>
</tr>
<tr>
<td>season</td>
<td>Spring, summer, fall or winter</td>
<td>character varying(100)</td>
</tr>
<tr>
<td>temperature</td>
<td>Weather information</td>
<td>real</td>
</tr>
<tr>
<td>humidity</td>
<td>Weather information</td>
<td>real</td>
</tr>
<tr>
<td>rainfall</td>
<td>Rain volume (suggested)</td>
<td>real</td>
</tr>
<tr>
<td>wind</td>
<td>Wind speed (suggested)</td>
<td>real</td>
</tr>
<tr>
<td>specialevent</td>
<td>The name of special event</td>
<td>character varying(100)</td>
</tr>
</tbody>
</table>

Table A.5: Daily run: state estimation table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK: Trip ID</td>
<td>Unique ID for each trip</td>
<td>integer</td>
</tr>
<tr>
<td>PK: Current Time</td>
<td>Current simulation time (1-27)</td>
<td>integer</td>
</tr>
<tr>
<td>PK: Departure Time</td>
<td>Departure time</td>
<td>integer</td>
</tr>
<tr>
<td>PK: Arrival Time</td>
<td>Arrival time</td>
<td>integer</td>
</tr>
<tr>
<td>Arrival</td>
<td>Arrival status (0 or 1)</td>
<td>boolean</td>
</tr>
<tr>
<td>Energy Consumption</td>
<td>Estimated energy consumption</td>
<td>real</td>
</tr>
<tr>
<td>Token Usage</td>
<td>Token assigned for each trip</td>
<td>real</td>
</tr>
</tbody>
</table>
Table A.6: Daily run: state prediction table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK: Trip ID</td>
<td>Unique ID for each trip</td>
<td>integer</td>
</tr>
<tr>
<td>PK: Current Time</td>
<td>Current simulation time (1-27)</td>
<td>integer</td>
</tr>
<tr>
<td>PK: Departure Time</td>
<td>Departure time</td>
<td>time</td>
</tr>
<tr>
<td>PK: Arrival Time</td>
<td>Arrival time</td>
<td>time</td>
</tr>
<tr>
<td>Arrival</td>
<td>Arrival status (0 or 1)</td>
<td>boolean</td>
</tr>
<tr>
<td>Energy Consumption</td>
<td>Estimated energy consumption</td>
<td>real</td>
</tr>
<tr>
<td>Token Usage</td>
<td>Token assigned for each trip</td>
<td>real</td>
</tr>
</tbody>
</table>

Table A.7: Historical scenario: information table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK: Scenario</td>
<td>Unique ID for each trip (1-27)</td>
<td>integer</td>
</tr>
<tr>
<td>PK: Current Time</td>
<td>Current time</td>
<td>time</td>
</tr>
<tr>
<td>PK: Prediction Time</td>
<td>Prediction time</td>
<td>time</td>
</tr>
<tr>
<td>PK: Label</td>
<td>Label for the day</td>
<td>string</td>
</tr>
<tr>
<td>cumDemand</td>
<td>Cumulative demand (from t to t+15)</td>
<td>real</td>
</tr>
<tr>
<td>stepDemand</td>
<td>Demand (from t+10 - t+15)</td>
<td>real</td>
</tr>
<tr>
<td>cumEnergy</td>
<td>Cumulative energy (from t to t+15)</td>
<td>real</td>
</tr>
<tr>
<td>stepEnergy</td>
<td>Step energy (from t+10 - t+15)</td>
<td>real</td>
</tr>
<tr>
<td>Token</td>
<td>Token assigned (from 6:30 - t+15)</td>
<td>real</td>
</tr>
<tr>
<td>BHEC</td>
<td>Beyond horizon energy consumption</td>
<td>real</td>
</tr>
</tbody>
</table>

Table A.8: Historical scenario: result table

<table>
<thead>
<tr>
<th>Column</th>
<th>Description</th>
<th>Field Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>PK: Current Time</td>
<td>Current time</td>
<td>time</td>
</tr>
<tr>
<td>Predicted Feature</td>
<td>Traffic feature</td>
<td>real</td>
</tr>
<tr>
<td>Predicted BHEC</td>
<td>Predicted BHEC</td>
<td>real</td>
</tr>
<tr>
<td>Actual Feature</td>
<td>Actual traffic feature</td>
<td>real</td>
</tr>
<tr>
<td>Actual BHEC</td>
<td>Actual beyond horizon energy consumption</td>
<td>real</td>
</tr>
</tbody>
</table>
Bibliography


Andrey Markov. Extension of the limit theorems of probability theory to a sum of variables connected in a chain. 1971.


