The Dissertation Committee for Hossein Tavana certifies that this is the approved version of the following dissertation:

INTERNALLY-CONSISTENT ESTIMATION OF DYNAMIC NETWORK ORIGIN-DESTINATION FLOWS FROM INTELLIGENT TRANSPORTATION SYSTEMS DATA USING BI-LEVEL OPTIMIZATION

Committee:

Hani S. Mahmassani, Supervisor

Carl T. Haas

Chandra R. Bhat

Randy B. Machemehl

Patrick Jaillet

INTERNALLY-CONSISTENT ESTIMATION OF DYNAMIC NETWORK ORIGIN-DESTINATION FLOWS FROM INTELLIGENT TRANSPORTATION SYSTEMS DATA USING BI-LEVEL OPTIMIZATION

by

Hossein Tavana, B.S., M.S.

Dissertation

Presented to the Faculty of the Graduate School of

the University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

December 2001

UMI Number: 3055254

ProQuest Information and Learning Company
300 North Zeeb Road
PO Box 1346
Ann Arbor, MI 48106-1346

To my

dearest mom and dad

without whose sacrifices

this achievement

would not have been possible.

Acknowledgments

I am grateful to all my friends, classmates, the faculty and staff at the University of Texas at Austin and the Department of Civil Engineering.

In particular, my sincerest thanks go to my educational and personal advisor, my guru, scholar and professor, Dr. Hani Mahmassani, without whose guidance, broad and deep knowledge and patience, the completion of my studies would have been impossible. He has always been available to advise me throughout the happiest and hardest times.

I would also like to extend my gratitude to Prof. Clyde E. Lee who supported and guided me during the beginning of my doctoral studies, prior to his retirement.

Special thanks go to my committee members, Professors Carl Haas, Randy Machemehl and Chandra Bhat for their support and invaluable insight. And last but not least, I would like to thank Professor Patrick Jaillet who has been a friend and a great teacher.

I would like to acknowledge my colleagues at Continental Airlines, in particular, my supervisors Judy Pastor, Greg Lough and Bill Brunger. They have been supportive and have given me the leeway to fulfill the requirements of my Ph.D. while working as a member of the Operations Research Group.

I feel obliged to name some of my friends and classmates who were valuable resources during my life and studies in Austin. They include but are not limited to Yi-Chang Chiu, Khaled Abdelghany, Ahmed Abdelghany, Manuel Trevino, Karthik Srinavasan, Akmal Abdelfatah, Pam Murray, Michael Hunter, Joshua Peterman, and in particular my close friend and tennis partner Nhan Huynh, who was more helpful than ever during the last few months when I was not in residence in Austin.

In closing, I would like to thank my beloved wife, Traci Petersen, who was very patient and supportive during the final year of my studies. Ultimately, I must acknowledge and thank my mom, dad and brothers, who were supportive of my decision to leave home to further pursue my studies; they bore the distance and endured the emotional difficulties so I could achieve my lifetime dream.

Hossein Tavana
Houston, Texas
October 2001

This research is partially funded through the Southwest University Transportation Center (SWUTC)--Project No. 472840-00070, titled “From Information to Knowledge, Strategies and Techniques for Mining Large Real-Time Traffic Databases”. However, the author is solely responsible for the analysis and views expressed in this dissertation.

INTERNALLY-CONSISTENT ESTIMATION OF

DYNAMIC NETWORK ORIGIN-DESTINATION FLOWS FROM

INTELLIGENT TRANSPORTATION SYSTEMS DATA

USING BI-LEVEL OPTIMIZATION

Publication No. _______________

Hossein Tavana, Ph.D.
The University of Texas at Austin, 2001

Supervisor: Hani S. Mahmassani

Deployment of Intelligent Transportation Systems (ITS) is providing researchers and practitioners with an unprecedented amount of valuable on-line and archived traffic data. To date, ITS data have been used primarily to support real-time operational applications, while other potential uses of these data have been largely ignored.

In this research, the effort to extract knowledge from the on-line or archived data gathered by Advanced Transportation Management Systems (ATMS) is focused on the estimation of dynamic origin-destination (OD) flows using optimization methods. In addition to their use for planning purposes, time-dependent OD flows can

be used as an input to Dynamic Traffic Assignment (DTA) systems. However, gathering OD demand flow information directly by conducting surveys is very costly and time consuming.

To estimate the OD flows, a methodology is developed to minimize an overall measure of the deviation of estimated link-flows from the time-varying link-flow observations, subject to a set of constraints. The set of constraints could include non-negativity constraints, initial condition constraints, cordon line counts and the user's route-choice behavior or traffic assignment rules. The traffic assignment solution, itself, is often obtained by optimizing an objective function. This objective function can explicitly be included in the constraints of the main or upper minimization problem. This formulation results in a bi-level optimization or theoretical game problem.

In this dissertation, the upper-level problem is formulated alternatively as linear and non-linear optimization problems. To solve the lower-level traffic assignment problem, a DTA simulation program, namely DYNASMART-P, is used to find the equilibrium flows. The suggested algorithm iterates between the upper-level and the lower-level optimization problems for a pre-specified number of times or until convergence in terms of the estimated OD flows or the simulated link flows is achieved.

To integrate the *a priori* information on OD demand flows with the information extracted from the link flow observations, adoption of the Bayesian inference method is proposed. If such information on OD flows is available, Bayesian inference treats the old information as the target values to update the estimated OD flows from the sample of the link flow observations.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1. INTRODUCTION

1.1. Research Motivation and Objectives

Deployment of Intelligent Transportation Systems (ITS) is providing researchers and practitioners with unprecedented amount of valuable on-line and archived traffic data. These data can be used in different applications such as traffic simulation, traffic control and Advanced Traveler Information Systems (ATIS). To date, however, ITS data have been used primarily to support real-time operational applications, while other potential uses of these data have been largely ignored. On-line or archived data carry useful information that can be extracted to improve the theoretical and empirical basis of the models and procedures used in the analysis, design and operation of transportation systems.

Real-time or archived data can be used in various transportation engineering applications such as Advanced Commercial Vehicle Systems (ACVS), Advanced Public Transportation Systems (APTS), Advanced Traveler Information Systems (ATIS), Advanced Traffic Control Systems (ATCS), transportation safety studies and traffic simulation.

In this research, the effort in extracting knowledge from the on-line or archived data is focused on the estimation of dynamic origin-destination (OD) flows using optimization methods from the information gathered by Advanced Transportation Management Systems (ATMS). In addition to their use for planning purposes, time-dependent origin-destination (OD) flows may be used as input to Dynamic Traffic Assignment (DTA) models, and to improve the external consistency of DTA systems. The method presented in this research is implemented in DYNASMART-P, which is the planning version of a Dynamic Traffic Assignment

(DTA) simulation program developed at the University of Texas at Austin (Mahmassani *et al.*, 2000). Two versions of this DTA system are now developed, with one targeted at applications in transportation planning, and the other (DYNASMART-X) intended for real-time control applications, ATCS and ATIS in the realm of Intelligent Transportation Systems (ITS).

Time-dependent origin-destination flows constitute an essential input to DTA systems. However, gathering OD demand flow information directly by conducting surveys is very costly and time consuming, and the required detailed information in a DTA system makes its collection and frequent update impractical if conventional methods are used.

The method presented for OD-flow estimation can also be used internally in a DTA system to improve the consistency of the results with real world observations by reducing the overall errors due to assignment assumptions, flow propagation inconsistencies, etc. Doan, Ziliaskopoulos and Mahmassani (1999) have classified the sources of errors in a real-time dynamic traffic assignment system into the following: 1) demand estimation errors, 2) path estimation errors, 3) traffic propagation errors, 4) internal traffic model structure errors, and 5) on-line data observation errors. In this context, the state of a traffic network is specified by the path that every vehicle follows; if we know the exact spatio-temporal path of every vehicle in the network, the network state can be uniquely defined.

Though continuing advances in wireless technologies have made it possible to track each suitably equipped or electronically tagged vehicle, widespread adoption is not likely in the near future. More importantly, concerns over privacy issues make the widespread adoption of this technology, except in emergency cases, improbable. Therefore, in a DTA system, the state of the system is usually estimated by attempting to replicate the time-varying traffic flows in the network. Errors, then, are discrepancies between the observed link-level flows and the corresponding DTA model estimates.

In practice, it is not possible to isolate errors caused by different sources as they are confounded in the observed error. The methods presented in this research to estimate dynamic OD flows rely on the results of the DTA simulation program. In these methods, the system-wide deviation of the simulated flows from the observed traffic flows are minimized, thereby reducing the combined errors arising from all different sources.

The objective of this research is to propose efficient methods to estimate dynamic origin-destination demand flows from time-dependent traffic flows using well-established optimization methods, with the flexibility to incorporate in the formulation information that might be obtained from other sources. In this research, the DTA simulation program is used to assign traffic to the network and to estimate the values of the parameters required for estimation of OD flows. The proposed methods are such that once the estimated OD matrix is assigned to the network, the observed flows would be as close as possible to the measured time-varying traffic volumes. Thus, the estimation process can be used to improve the external consistency of the DTA simulation program. The estimated time-dependent OD matrix will also provide an essential input required to run a DTA simulation program to deliver route-guidance to drivers and to assess ITS-related traffic control and planning strategies.

The problem addressed in this research is as follows: Given the observations of the time-varying traffic flow on the links in a network, we seek to estimate the time-dependent origin-destination flows, which, when assigned to the network, result in traffic volumes that are as close as possible to the measured link volumes.

The general conceptual optimization problem for estimation of dynamic origin-destination flows from dynamic traffic flows can be written as follows:

$$\boldsymbol{D}^* = arg\ min\ \ Z(\boldsymbol{C}, \hat{\boldsymbol{C}}(\boldsymbol{D}, \boldsymbol{R}, \boldsymbol{F}, \boldsymbol{P})) \qquad (1.1)$$

subject to:

set of constraints

where Z is a general function of links' measured traffic volume, $\boldsymbol{C}$, and the estimated link flows, $\hat{\boldsymbol{C}}$, which in turn is an implicit function of OD demand flows, $\boldsymbol{D}$, users' route-choice behavior, $\boldsymbol{R}$, flow propagation rules, $\boldsymbol{F}$, and set of control policies, $\boldsymbol{P}$, among others. Function Z should be of a form to represent the errors in estimation.

The set of constraints depends on the application of the problem as well as the desired level of accuracy, and it can include non-negativity constraints, initial condition constraints, fixed OD demand values (if information on some OD flows is known with certainty), cordon line counts, etc. On the other hand, users' route-choice or traffic assignment rules are often obtained by optimizing an objective function, which can be explicitly included in the set of constraints. This formulation results in a bi-level optimization or theoretical game problem (Bard, 1998).

In this research we seek to estimate OD demand values by minimizing the sum of squared errors in estimation of time-varying traffic volumes. Therefore, the above conceptual objective function can be more explicitly written as:

$$D^* = \arg\min \sum_l \sum_t \left[c_{(l,t)} - \hat{c}_{(l,t)}\right]^2 \qquad (1.2)$$

subject to:

set of constraints

where c and $\hat{c}$ represent the observed and estimated (simulated) traffic volumes respectively, l denotes sequential link numbers that have traffic volume data, and t denotes sequential observation interval numbers. A complete formulation is provided in Chapter 3.

In general, the deviation in observed flows and estimated flows can be due to different sources of errors.

$$\boldsymbol{C} = \hat{\boldsymbol{C}} + \boldsymbol{E_D} + \boldsymbol{E}_R + \boldsymbol{E}_F + \boldsymbol{E}_P + \boldsymbol{E}_U + \text{interaction error terms} \qquad (1.3)$$

where

$\boldsymbol{C}$ is the vector of time-varying link flow observations

$\hat{\boldsymbol{C}}$ is the vector of time-varying traffic volumes on links resulting from simulation

$\boldsymbol{E_D}$ is the vector of errors due to using the 'estimated' OD flows as the input

$\boldsymbol{E_R}$ is the vector of errors due to inconsistencies in traffic assignment assumptions

$\boldsymbol{E_F}$ is the vector of errors due to inconsistencies in flow propagation assumptions

$\boldsymbol{E_P}$ is the vector of errors due to inconsistencies in traffic control assumptions

$\boldsymbol{E}_U$ is the vector of errors due to unknown or other sources including traffic volume measurement errors (sensor errors)

We combine all sources of errors and denote it by *E*, that is

$$\boldsymbol{C} = \hat{\boldsymbol{C}} + E \tag{1.3}$$

Equation (1.3) coupled with the relation among simulated link flows, link-flow proportions and OD demand flows (as stated below) constitute the basic formulation for estimation of dynamic OD flows from traffic data.

$$\hat{\boldsymbol{C}} = \hat{\boldsymbol{P}}.\boldsymbol{D} \tag{1.4}$$

or

$$\boldsymbol{C} = \hat{\boldsymbol{P}}.\boldsymbol{D} + E \tag{1.5}$$

where $\hat{\boldsymbol{P}}$ denotes the link-flow proportion matrix, which consists of $p_{(l,t)(\tau,i,j)}$ elements denoting fraction of vehicular demand flows from *i* to *j*, starting their trips during departure interval τ, that are observed on link *l* during observation interval *t*. In a two dimensional representation of link-flow proportions, (*l,t*) denotes the rows and (τ,*i,j*) represent the columns (the elements of matrix are shown in Chapter 3) .

Minimization of equation (1.2) produces estimates of the optimal time-dependent OD flows that minimize the *overall* sum of squared errors. As mentioned before, the key point is to estimate time-dependent OD flows, which, when assigned to the network, result in time-varying link flows that are as close as possible to the measured link volumes.

1.2. Overview of Proposed Methods

To solve the optimization problem stated in (1.3), researchers have adopted different methods, which are reviewed in more detail in the second chapter of this dissertation.

The methods proposed in this research are based on the generalized least-squares (GLS) estimation technique. In the formulation of the problem in a transportation network, link-flow proportions play a key role. In a dynamic traffic assignment, particularly when the network is congested, the values of these variables are not constant and depend on the (unknown) OD demand flow values, though their dependence on OD flows is often ignored.

To address this problem in the static case, some researchers have adopted a bi-level optimization formulation. By solving a traffic assignment problem in the lower-level optimization, the dependency of link-flow proportions on demand flows is incorporated in the solution. The first approach presented in this research adopts the same method and extends it to the dynamic case. In the upper-level, we treat equation (1.5) as a quasi-linear equation and obtain the conventional generalized least-squares estimate of the time-dependent demand flows. In the lower-level, we use the DTA simulation program, DYNASMART-P, to find the equilibrium flows and link-flow proportions. We iterate between the upper-level and lower-level optimization problems for a pre-specified number of times or until convergence in terms of estimated OD values or simulated link flows is achieved. We name this approach *Bi-Level Generalized Least-Squares Estimation Method* (Bi-GLS). This formulation is akin to a theoretical game with upper-level and lower-level players or groups of players.

In the second proposed method in this research, the problem is formulated as a non-linear optimization problem. To find the optimal OD flows, we explicitly include the derivative of link-flow proportions with respect to demand flows in the derivation of the optimality conditions. As presented in Chapter 3, the solution to this approach results in a fixed-point problem formulation that can be decomposed into a set of simultaneous quadratic equations. Finding the derivatives of link-flow proportions with respect to demand flows is the challenging issue in this approach, particularly in a dynamic traffic assignment environment

Because an analytical relation cannot be established between link-flow proportions and dynamic demand flows in a transportation network, the DTA simulation program is used to estimate the derivatives numerically. As the values of the derivatives might change with the (unknown) OD flow values, we may still need to use the bi-level (non-linear) optimization formulation (Bi-NLP); in this case the upper-level is formulated as a non-linear optimization problem, while the lower-level is an equilibrium dynamic traffic assignment problem, solution to which results in the estimated values of link-flow proportions and their derivatives.

For clarity, the derivation of the non-linear optimization formulation is presented in two stages: first it is assumed that the error terms are independently and identically distributed (i.i.d); due to similarity of this assumption to the assumptions made in ordinary least-squares estimation, we call this formulation as *Ordinary Non-Linear Optimization formulation*. This formulation is then extended to the cases where there are (known) correlations among error terms; we call this approach a *Generalized Non-Linear Optimization Formulation*.

To incorporate *a priori* information on OD demand values with the information extracted from the link flow observations, implementation of Bayesian inference method is proposed. If such information on OD flows exists, Bayesian inference treats the old information as the target value and updates the outdated or recent OD flow values with the information obtained from the link flow measurements. It is shown that using the Bayesian inference method results in OD demand flows as if the target OD demand flows are directly incorporated in the generalized least-squares estimation. The former method is preferable, because the existing past information can be combined with the estimated OD demand flows, irrespective of the estimation method.

1.3. Dissertation Overview

In the next chapter, the background on estimation of OD flows from traffic counts is reviewed. In this review, special attention is paid to methods pertaining to generalized least-squares estimation of demand flows. In Chapter 3, the formulation of the proposed methods is presented, first the existing generalized least-squares estimation is extended to a bi-level optimization problem to estimate the time-dependent OD flows and then, the bi-level non-linear optimization formulation is presented. The formulation of the problem is further extended to rolling-horizon instances. In Chapter 4, Bayesian inference method is reviewed and its applications to the estimation of dynamic origin-destination are discussed. In Chapter 5, issues regarding implementation of the proposed methods as an integral part of DYNASMART-P simulation program are discussed. In Chapter 6, the performance of the proposed methods is examined by testing hypothesis and conducting pertaining experiments. In Chapter 7, future research and possible extensions are discussed. The pseudo-codes of the added algorithm to DYNASMART-P system for each of the proposed methods are described in Appendices A to E.

CHAPTER 2. BACKGROUND ON OD-FLOW ESTIMATION FROM TRAFFIC COUNTS

2.1. Introduction

In general, there are three different categories of methods for OD estimation. The most traditional and the costliest method is to conduct surveys for the *direct sample estimation* of the OD matrix. Many types of surveys such as home or destination interviews, roadside interviews or a combination of those may be used (Cochran, 1963; Yates, 1981).

A second commonly used method can be defined as *model estimation*. The OD matrix is estimated by applying a system of models that give the number of journeys made during a certain period of time. In this method the demand models are used as a relation between the OD matrix (to be estimated) and other variables such as socio-economic, geographic and transportation supply characteristics. Model specification and parameter estimation can be performed based on the results of the surveys carried out in the study area, or other models calibrated in similar areas.

Finally, the third method is *estimation of the OD matrix from traffic flows*. This method is more recent than the other two methods. Approaches for estimation of OD matrices from traffic counts have been motivated primarily by the practical realities of limited data availability, and relative ease of obtaining link traffic counts compared to more elaborate survey procedures. The approximate nature of this approach is offset by its practicality and affordability. Furthermore, with increasing deployment of ITS in recent years, traffic flow data are collected continuously and at no extra cost; therefore, the time-dependent OD can be updated as frequently as desired, which is not practical with conventional household surveys.

Review of the previous works on estimation of OD flows from traffic counts is presented in two sections. The first section provides a general and historical presentation of different approaches for estimation of OD flows from traffic counts. In the second section, some of the works that are closely related to the proposed methods in this dissertation are discussed in more detail.

2.2. General Review of Estimation of OD Flows from Traffic Counts

Up to the late 1980's, most OD estimation methods using traffic counts dealt with "static" estimation problems, which seek to estimate average OD flow rates that are assumed to be constant over a significant period of time, given average link traffic flow measurements over the same period. Robillard's works (1973, 1975) represent the first major effort in this area. He solved a linear regression problem to determine the total originating and terminating trips for each zone. A generalized gravity model was then used to determine the trip table. Willumsen (1978) and Nguyen (1982) provide extensive bibliographies.

In the static case, the problem is generally under-specified, that is the number of links with aggregated traffic flow data in the network is less than the number of unknowns (OD matrix cells). Therefore, prior beliefs about the OD matrix must be incorporated to allow a unique solution.

The estimation process tries to minimize a measure of distance, "entropic" or Euclidian, from a "target" matrix given by a model or by an old direct estimate, taking into account the traffic flow measurements on the links, in such a way that if the estimated OD matrix is assigned to the network, the observed flows would be as close as possible to the measured flows (Gur *et al.*, 1980).

The early works used the closely related principles of minimal information and maximum entropy, as the entropic distance, to formulate the problem as an optimization problem. Willumsen (1981) gives a useful general description of the method. Methods for producing trip matrices in this way have been proposed by Robillard (1975), Willumsen (1982), Van Zuylen and Willumsen (1980), using the assumption of proportional assignment, and by Nguyen (1977) and LeBlanc and Farhangian (1982) using equilibrium assignment.

The problem of estimating turning flows at an intersection from traffic counts on the inflows and outflows are of the same form and work by Jeffreys and Norman (1977), Mekky (1979), Van Zuylen (1979) and Cremer and Keller (1981) has been directed toward solving this problem.

Later some statistical aspects of the estimation were considered. Bell (1983) expressed the variance-covariance matrix of the maximum entropy estimator and Maher (1983) proposed a Bayesian estimator for the OD table in which a multivariate normal distribution was hypothesized for both the trip matrix prior distribution and the observed flows.

Cascetta (1984) used generalized least-squares (GLS) estimators to combine direct estimates, as the “target” table, with the traffic counts on some network links by means of an assignment model. The presence of measurement errors and temporal variability in the observed flows were explicitly considered. Bell (1984) showed that the GLS approach approximates the entropy approach originally propounded by Van Zuylen and Willumsen (1980) when the link flows are known to a high level of accuracy. Bell (1991) incorporated inequality (non-negativity) constraints in estimation of the static OD matrices using the generalized least squares method. Like many other authors, he assumed that link choice proportions are proportional to the demand level and are known with certainty.

The earliest reported works to estimate "time dependent" OD matrices are by Cremer and Keller (1981, 1984, 1987) and Cremer (1983) who developed four methods for identification of dynamic origin-destination flows in interchanges. They proposed the following methods: an ordinary least-squares estimator involving cross-correlation matrices, a constrained optimization method, a simple recursive estimation formula and estimation by Kalman filtering. At interchanges and intersections, traffic counts provide the total generation and attractions at the entry and exit points, and the estimator should estimate the distribution (the split ratios) of demand among different entrance and exit points. It is mentioned that in an intersection, enough information can be obtained from the traffic counts to make the problem over-specified in order to obtain a unique and biased-free estimates for the unknown OD flows without further *a priori* information. In these implementations, it was assumed that travel times from all origins to all destinations were known *a priori*.

It should be noted that the estimation of dynamic OD flows for an interchange is different in nature from estimation of demand flows in a network because users do not have the option to take different paths from origin to their destinations. Therefore, there is no need to deal with any traffic assignment assumptions in estimation of OD flows in an interchange.

Willumsen (1984) addressed the estimation of time-dependent trip matrices. Keller and Ploss (1987) used entropy maximization method to estimate OD's at intersections to provide better strategies for on-line signal control. Okutani (1987) used the Kalman filtering approach to estimate dynamic OD matrices.

Since the early 1990's, more attention has been accorded to the problem of "dynamic" OD estimation. The European Community program 'DRIVE I' founded the research project ODIN to assess the role of OD information in traffic control problems (Inaudi *et al.*, 1991).

A study by Cascetta, Inaudi and Marquis (1993) can be considered a milestone in estimation of dynamic OD flows from traffic counts in a *network* using the generalized least-squares method. This study is described in more detail in the second section of this chapter.

Other approaches for estimation of dynamic OD flows include the works by Chang and Tao (1995), and Xu and Chan (1993). In those works, OD flows are estimated (without *a priori* information) by introducing numerous dynamic screen-lines and assuming that travel times between origin and destinations and screen-lines are known.

Ding, Mirchandani and Nobe (1999) have revisited the static OD estimation problem. They have presented one non-iterative (open-loop) method and one iterative (closed-loop) algorithm. In both methods, it is assumed that link-flow proportions are known with certainty. The non-iterative method is very similar to the Bayesian inference implementation proposed by Maher (1983). In the closed-loop method, algorithm iterates based on the computed OD matrix and finds the incremental difference in OD demand mean values and their variances. It is claimed that by using the incremental difference, one can forecast the traffic volume in near future and it can be used in real time for estimation of traffic volumes in short time intervals, but still most of the formulation and specifically link-flow proportions are specified for the static case. Furthermore, in the implementation, in the first iteration the sum of squares of differences in observed flows and computed flows are minimized, while in the subsequent iterations the sum of the absolute differences are minimized.

Dixon and Rilett (2000) have incorporated Automatic Vehicle Identification (AVI) data to estimate the OD demand flows. They compare four different estimation methods, two of them are Generalized Least Squares estimation methods and the other two are based on Kalman filtering techniques. The methods were used to estimate OD flows on twenty kilometers (12.5 miles) of a freeway stretch in Houston. AVI data were used to provide link volumes, link-flow proportions as well as observed OD flows.

Ashok (1996) and Kang (1999), in their dissertations used the Kalman filtering techniques for estimation of OD flows. Ashok used Kalman filtering to estimate OD demand values directly, and Kang introduced the concept of polynomial (third degree) variation of OD demand values within each estimation period to reduce the degrees of under-specification of the problem. He used Kalman filtering to estimate the coefficients of the polynomial functions.

2.3. Detailed Review of More Related Research Works

Cascetta, Inaudi and Marquis (1993) generalized the statistical framework proposed for the "static" problem and extended it to the dynamic OD estimation case. Another contribution was related to the formulation of the estimation problem for a general *network* making use of the notation and modeling results in the field of Dynamic Traffic Assignment. They used the notation of "flow proportions" as the fraction of OD flow that contributes to the flow on a link in a time interval. As their work is very similar to the estimation method proposed in this dissertation, it is explored in more detail.

Two types of estimators were proposed: simultaneous estimators which produce joint estimates of the whole set of OD matrices, and sequential estimators which produce a sequence of OD estimates for successive time slices. They used the Generalized Least- Squares (GLS) method, which combined traffic counts with other available information on OD flows such as earlier matrices and surveys.

The objective function in the optimization process consisted of the sum of two functions: 1) a function of deviation of time dependent OD from an old or assumed OD matrix, and 2) a function of deviation of traffic counts and link flows obtained by assigning the *old* demand matrix. Specifically,

$$\boldsymbol{\delta}_h = \left(\sum_{t=1}^{h} \hat{\boldsymbol{P}}_{ht} \boldsymbol{s}_t \right) - \hat{\boldsymbol{v}}_h \qquad (2.1)$$

$$(d_1^* \ldots d_{n_h}^*) = \underset{s_1 \geq 0 .. s_{n_h} \geq 0}{\arg\min} \sum_{h=1}^{n_h} \left[(\boldsymbol{s}_h - \hat{\boldsymbol{d}}_h)^T \boldsymbol{V}_h^{-1} (\boldsymbol{s}_h - \hat{\boldsymbol{d}}_h) + \boldsymbol{\delta}_h^T \boldsymbol{W}_h^{-1} \boldsymbol{\delta}_h \right] \qquad (2.2)$$

where

$\mathbf{s}_h$ is the current value of the demand vector departing at time interval h,

$\hat{\mathbf{d}}_h$ is the estimated or target value of demand departing at time interval h. It is mentioned that the number of stochastic equations in observed flows ($n_l.n_h$) is usually smaller than the number of unknown OD demand flows ($n_r.n_h$), therefore existence of a target matrix is needed to make the problem over-specified. (n_h is the number of observation or departure intervals, n_l is the number of links with observed flows, n_r is the number of OD pairs.)

$\hat{\mathbf{v}}_{h}$ is the *counted* traffic flows on links at time interval h. It is the estimate of the flow due to the errors in *measuring* the actual flows.

$\hat{\mathbf{P}}_{ht}$ is the matrix of elements $\hat{p}_{lh}^{rt}$, the fraction of OD flow d_{rt} contributing to the flow on link l in interval h. This matrix is calculated based on the current demand matrix, $\mathbf{s}_t$. It is mentioned that these values can be obtained through path choice and Dynamic Network Loading (DNL) models.

$\boldsymbol{V}_h$ is the variance-covariance matrix of the vector of sampling errors affecting the estimate of $\hat{\boldsymbol{d}}_{h}$.

$\boldsymbol{W}_h$ is the variance-covariance matrix of the combined assignment and measurements errors.

The authors also proposed a sequential estimator, in which a demand vector for a single interval h is estimated at each time interval:

> The main idea in this approach is that of expressing the counts of a period h as a linear (stochastic) function of the unknown demand of the same period only. This is achieved by equating the demand relative to previous periods to the already-computed estimates $\boldsymbol{d}_t^*$.

$$\hat{\boldsymbol{v}}_h = \sum_{t=1}^{h-1} \hat{\boldsymbol{P}}_{ht}\boldsymbol{d}_t^* + \hat{\boldsymbol{P}}_{hh}\boldsymbol{s}_h + \mu_h \qquad (2.3)$$

The only difference in the GLS formulation (the first case shown above) is that in this case,

$$\delta_h = \sum_{t=1}^{h-1} \hat{\boldsymbol{P}}_{ht}\boldsymbol{d}_t^* + \hat{\boldsymbol{P}}_{\boldsymbol{hh}}\boldsymbol{s}_h - \hat{\boldsymbol{v}}_h \qquad (2.4)$$

It is suggested that if Bayesian inference is used, the estimated OD flows in the previous period be used as an initial value for the current period ($\boldsymbol{d}^*_{h-1} = \hat{\boldsymbol{d}}_h$).

The authors further modified the formulation to compute the average demand over an aggregated time interval based on aggregated traffic counts. In this case, the actual demand between OD pair *r* leaving in period t, d_{rt}, is broken into two components, that is

$$d_{rt} = \bar{d}_r + \varepsilon_{rt}$$

or in matrix form,

$$\boldsymbol{d}_t = \bar{\boldsymbol{d}} + \varepsilon_{\boldsymbol{t}}$$

where

$\bar{d}_r$ is the average value of demand (of OD pair *r*) over the entire observation period H, i.e. $\bar{d}_r = (1/n_h)\sum_t d_{rt}$

ε_{rt} is the residual (deviation from the average value of OD pair *r*) at time interval *t*.

Consequently,

$$\hat{\boldsymbol{v}}_h = \sum_{t=1}^{h} \hat{\boldsymbol{P}}_{\boldsymbol{ht}} \bar{\boldsymbol{d}} + \sum_{\boldsymbol{t=1}}^{\boldsymbol{h}} \hat{\boldsymbol{P}}_{\boldsymbol{ht}} \varepsilon_t + \mu_h \qquad (2.5)$$

Therefore, an overall GLS or multivariate normal (MVN) Bayesian estimator is obtained:

$$\boldsymbol{\delta_h} = \sum_{t=1}^{h} \hat{\boldsymbol{P}}_{ht}\bar{\boldsymbol{s}} - \hat{\boldsymbol{v}}_h \quad (2.6)$$

$$\bar{\boldsymbol{d}}^* = \arg\min_{\bar{s}\geq 0}\left[(\bar{\boldsymbol{s}} - \hat{\boldsymbol{d}})^T \boldsymbol{V}^{-1}(\bar{\boldsymbol{s}} - \hat{\boldsymbol{d}}) + \sum_{h=1}^{n_h}(\boldsymbol{\delta}_h^T \boldsymbol{W}_h^{-1}\boldsymbol{\delta}_h)\right] \quad (2.7)$$

A particular case of the estimator is obtained when the number of independent equations from traffic counts ($n_h.n_l$) is larger than the number of unknown OD demand cells (n_r), in which no sampling or *a priori* information is needed (infinite variance of the prior distribution of $\bar{d}_r$). In this case the average OD demand estimator is:

$$\bar{\boldsymbol{d}}^* = arg\,min_{\bar{s}\geq 0} \sum_{h=1}^{n_h}(\delta_h^T \boldsymbol{W}_h^{-1}\delta_h) \quad (2.8)$$

Despite the authors' significant contributions to the estimation of dynamic OD matrix, their assumptions and implementations have several shortcomings:

- In the sequential estimation of demand (consecutive estimation periods), OD demand values for only time interval *h* is computed. Though this method might work well for a very small network (freeway links) or for an intersection, it is not suitable in a large network where long trips that are initiated in time interval *h* might not have reached their destination by the end of this time interval. If the time interval *h* is assumed to be long enough, the problem will tend toward the static case. In addition, vehicles that start their trip toward the end of the time interval *h*, mostly will not reach their destination by the end of the estimation interval. Therefore, in both cases, there may be a bias or high variance in the estimation of OD flows. It is not clear how the interval should be set to circumvent this problem.

- In the formulation, it is assumed that traffic volume on a link is linearly proportional to the demand (proportional assignment). In other words, in that formulation, like many other published works, the dependence of link-flow proportions on the demand is not explicitly included in the solution procedure. This dependency and non-linearity in link-flow proportions can be significant in a dynamic traffic assignment and particularly in congested networks.

To validate the models empirically, the authors implemented the method to estimate OD flows in a section of a freeway 140 km (87 miles) long, with 19 origins and 19 destinations, 171 one way OD pairs, and with 54 links. The authors had access to information on origin, destination, entrance, and exit time of each reported vehicle with the precision to the minute. The following points in those experiments are noteworthy:

- The authors knew the desired OD flows. Traffic counts were not measured, but they were computed “numerically” from the known OD flows. In addition, the link-flow proportions were computed coherently such that they would reproduce exactly the computed flows, given the exact demand. (The authors have acknowledged that this will overestimate the statistical performance of the tested methods.)
- When demand for an OD pair and a departure interval was low, the estimate had relatively large errors, therefore in the measure of performance, demands less than ten vehicles were not included.

- The problem was solved for the case that disturbances had constant variance and were not autocorrelated, that is $\boldsymbol{V}_h^{-1} = \boldsymbol{I}$ and $\boldsymbol{W}_h^{-1} = \boldsymbol{I}$.
- The simultaneous estimation was done on a two-hour time period. Three different departure intervals were considered: two hours, half an hour and 15 minutes. As an example, the estimation reduced the RMSE of the initial guess from 19 to 15 vehicles per departure interval when departure interval was two hours and from 7.6 to 5.7 vehicles per departure interval when departure intervals were 15 minutes. As mentioned, RMSEs were computed only for the demands greater than 10 vehicles per departure interval.
- The authors also tried different weights for the two functions in the objective functions (2.2). Interestingly, the best result was obtained when the weight for the first term (sum of squares of deviation from *a priori* demand) was the smallest, i.e. 0.2, showing that the link flow observations carried the most information. This fact could be expected, especially in the designed experiments where link-flow proportions and traffic counts would exactly replicate the demand flows.
- The sequential estimation method was also empirically validated. The 18-hour period was divided into 72 intervals of 15 minutes. The estimation was conducted for each 15-minute interval, but the required information for the estimation of OD in each 15-minute interval was contained in a rolling horizon of the past two hours. In other words, it was assumed that all journeys would reach their destinations in two hours. Therefore, in each two-hour rolling horizon, the estimated demands of the first seven 15-minute departure intervals were assumed to contribute to the flows in the current estimation interval.
- Two different assumptions were made as *a priori* information. First, the estimation for the previous time interval was used as the *a priori*,

and second the average daily demand flows, corrected for the flow volume in the current time interval, was used as the *a priori*.

- Again, different weights were considered for the two functions in the objective function. This time when the weight for the *a priori* information was higher, the estimation had a better performance.
- The problem with sequential estimation is that the vehicles that have just started their trips in the current estimation interval most likely have not reached their destination by the end of the estimation time interval. That is why when the weight for the *a prior* information was higher, the estimation performed better, because the traffic flow in 15-minute time intervals cannot produce enough information for estimation of OD flows in a large network.

Similar to the above work, most of the research work using optimization or generalized least squares methods has assumed proportional traffic assignment, that is it is assumed that the traffic assigned to a path or a link is linearly proportional to the demand values. This implies that link-flow proportions are treated as constant values. This assumption can be valid only in static and uncongested situations.

Nguyen (1977) was among the first researchers to incorporate traffic assignment rule into the OD flow estimation process. He presented an approach to estimate an OD matrix based on the assumption that observed link flows represent network equilibrium in the sense of satisfying Wardrop's first principle. He suggested solving the following nonlinear programming problem:

$$\min \boldsymbol{F} = \sum_{a}\left[\int_{0}^{f_a} \boldsymbol{t}_a(\boldsymbol{x})\, \boldsymbol{dx}\right] - \sum_{j} \hat{\boldsymbol{u}}_j \boldsymbol{T}_j$$

subject to:

$$\boldsymbol{T}_j - \sum_{k} \boldsymbol{h}_j^k = 0 \qquad \text{for each OD pair j}$$

$$\sum_{j}\sum_{r} \boldsymbol{d}_{ja}^k \boldsymbol{h}_j^k = 0 \qquad \text{for each link a}$$

$$\boldsymbol{f}_a, \boldsymbol{T}_j, \boldsymbol{h}_j^k \geq 0$$

where

$\boldsymbol{f}_a$ = observed flow on link *a*

$\boldsymbol{t}_a(\boldsymbol{x})$ = impedance function for link *a*

$\hat{\boldsymbol{u}}_j$ = observed OD impedance for interchange *j*

$\boldsymbol{T}_j$ = trips for interchange *j*

$\boldsymbol{h}_j^k$ = number of trips from interchange j using path *k*, and

$\boldsymbol{d}_{ja}^k$ = 1 if link *a* is in path *k* for interchange *j*, and

0 otherwise.

The decision variables (or unknowns) are the $\boldsymbol{T}_j$'s and $\boldsymbol{h}_j^k$'s. The above optimization problem is very similar to an equilibrium traffic assignment problem and a few iterative solution algorithms, very similar to the user equilibrium traffic assignment problem, were introduced by Nguyen (1977) and Turnquist and Gur (1979).

Yang *et al.* (1992) and Yang, Iida and Sasaki (1994) addressed the shortcoming of the assumptions that users route choice are independent of the OD demand in the static case. Oh (1992) examined the simultaneous estimation of OD matrices and proposed three different solution methods: penalty function method, extrapolation method, and perturbation method. Florian and Chen (1994) presented a bi-level programming formulation for the OD matrix estimation problem in congested networks and developed a coordinate descent solution method. Yang (1994) extended this approach, still in the static case, and developed more general methods and heuristic algorithms to solve the problem in situations where link flow interactions cannot be ignored.

In the static bi-level OD matrix estimation problem, the generalized least-squares estimation model has been coupled with an equilibrium traffic assignment in the form of two simultaneous optimization problems. The upper-level problem seeks to minimize the sum of squared error of traffic volumes plus the sum of squared errors of a target OD matrix, whereas the lower-level problem represents a network equilibrium assignment that guarantees that the estimated OD matrix and corresponding link flows satisfy the user-equilibrium conditions. The problem is presented in the form of a bi-level programming problem with variational inequality constraints.

The proposed bi-level optimization model has the following form:

$$\min F(t) = (\bar{\boldsymbol{t}} - \boldsymbol{t})^T \boldsymbol{U}^{-1} (\bar{\boldsymbol{t}} - \boldsymbol{t}) + (\bar{\boldsymbol{v}} - \boldsymbol{v})^T \boldsymbol{V}^{-1} (\bar{\boldsymbol{v}} - \boldsymbol{v})$$

subject to

$$\boldsymbol{t} \geq 0,$$

where v(t) solves

$$\boldsymbol{c(v)}^T .(\boldsymbol{e} - \boldsymbol{v}) \geq 0 \qquad \text{for all } \boldsymbol{e} \in \boldsymbol{\Omega}(\boldsymbol{t})$$

The variables $\boldsymbol{U}$ and $\boldsymbol{V}$ are weighting factors (or they could be variance-covariance matrices), $\bar{\boldsymbol{t}}$ is a vector representing a target OD matrix, $\bar{\boldsymbol{v}}$ is a vector representing observed link flows and $\boldsymbol{v}$ denotes the link-flow estimates obtained by loading an estimate of OD matrix $\boldsymbol{t}$, onto the network.

Yang (1994) proposed two heuristic methods to solve the problem. In both methods, the lower-level network equilibrium problem is solved first based on an initial demand matrix. Then some influence factors, $\boldsymbol{Z}$'s, are calculated as follows.

In one of the methods, called iterative estimation-assignment (IEA) algorithm, the influence factors are defined as the link usage proportions, $\boldsymbol{Z}=[p_{aw}]$ (which are the same as link-flow proportion terms used in this research). In the other method, called sensitivity-analysis based algorithm (SAB), the influence factors are defined as the derivatives of link flows with respect to OD demands, $\boldsymbol{Z}=[q_{aw}]$, where:

$$\boldsymbol{q}_{aw} = \frac{\partial \boldsymbol{v}_a}{\partial \boldsymbol{t}_w} \;, a \in \boldsymbol{A},\, w \in \boldsymbol{W}$$

where $\boldsymbol{A}$ is the set of links, and $\boldsymbol{W}$ the set of OD pairs.

The derivatives are obtained by performing a sensitivity analysis for a given solution of the network equilibrium problem. The sensitivity analysis method for equilibrium network flows has been developed by Tobin and Friesz (1988).

Based on the calculated influence factors and the nonlinear reaction function, link flows ($\boldsymbol{v}$) are linearly approximated as:

$$\boldsymbol{v}(\boldsymbol{t}) \approx \boldsymbol{v}(\boldsymbol{t}^*) + \boldsymbol{Z}(\boldsymbol{t} - \boldsymbol{t}^*)$$

where ($\boldsymbol{t}^*$, $\boldsymbol{v}(\boldsymbol{t}^*)$) is the current solution and $\boldsymbol{t}$ is the OD matrix which is estimated in the next iteration ($\boldsymbol{t}^*$ is shown by $\boldsymbol{t}^k$ and $\boldsymbol{t}$ by $\boldsymbol{t}^{k+1}$ in the following formulation). The upper level problem, then, is approximated as a quadratic programming problem.

In the IEA implementation, where link proportions are taken as the influence factors, $\boldsymbol{v}(\boldsymbol{t})$=$\boldsymbol{Zt}$, where $\boldsymbol{Z}$ is the link proportions obtained in the last main iteration. If the non-negativity constraint is omitted, the least squares estimate of demand will be the typical GLS estimate:

$$\boldsymbol{t}^{(\boldsymbol{k}+1)} = (\boldsymbol{U}^{-1} + \boldsymbol{Z}^{(\boldsymbol{k})\boldsymbol{T}}\boldsymbol{V}^{-1}\boldsymbol{Z}^{(\boldsymbol{k})})^{-1}(\boldsymbol{U}^{-1}\bar{\boldsymbol{t}} + \boldsymbol{Z}^{(\boldsymbol{k})\boldsymbol{T}}\boldsymbol{V}^{-1}\bar{\boldsymbol{v}})$$

The above equation is indeed the same as Bayesian inference with MVN distribution for the target demand matrix.

In the SAB implementation, $\boldsymbol{v}(\boldsymbol{t})$ is linearized using Taylor's expansion and the GLS solution to the upper-level optimization is slightly different:

$$\boldsymbol{t}^{(\boldsymbol{k}+1)} = (\boldsymbol{U}^{-1} + \boldsymbol{Z}^{(\boldsymbol{k})\boldsymbol{T}}\boldsymbol{V}^{-1}\boldsymbol{Z}^{(\boldsymbol{k})})^{-1}(\boldsymbol{U}^{-1}\bar{\boldsymbol{t}} + \boldsymbol{Z}^{(\boldsymbol{k})\boldsymbol{T}}\boldsymbol{V}^{-1}(\bar{\boldsymbol{v}} - \boldsymbol{v}^{(\boldsymbol{k})} + \boldsymbol{Z}^{(\boldsymbol{k})}\boldsymbol{t}^{(\boldsymbol{k})}))$$

In these iterative bi-level estimation algorithms, the reaction of the follower (the lower-level optimization) to the leader's decision (upper-level problem) is explicitly taken into account. Therefore, the author claims that, both algorithms are a close representation of the actual decision-making in terms of a Stackelberg game.

Practically, the difference between the two proposed methods is as follows. In the IEA algorithm, the lower-level optimization problem is first solved and, then, based on the estimated demand matrix, a new set of link flow proportions, $\boldsymbol{Z}$, is calculated. The calculated $\boldsymbol{Z}$ values are replaced in the upper-level equation to find a new set of OD flows. In contrast, in the SAB algorithm, the anticipated change of the traffic flows due to the change in demand flows is explicitly included in the upper-level solution. The drawback of the latter method is that the derivatives of link flows with respect to OD flows should be computed in every iteration of the algorithm.

Yang (1995) claimed that the two heuristic algorithms provide a close representation of solutions to a Stackelberg game, because for each estimated OD flow in the upper-level problem, $\boldsymbol{t}$, the lower-level decision variables, $\boldsymbol{v}$, are not assumed to be constant, but are updated based on the new estimated values of the OD flows using the relation $\boldsymbol{v(t)=v(t^{*})+Z(t-t^{*})}$. However, an iterative optimization-assignment algorithm is an exact and efficient algorithm for solving Cournot-Nash games.

In the next chapter of this dissertation, it will be shown that the first heuristic algorithm proposed by Yang, i.e. IEA, is still a solution to the Cournot-Nash game, because in the upper-level optimization, the dependence of link-flow proportions on the OD flows and their derivatives are ignored. In the bi-level nonlinear optimization formulation of dynamic networks, which is presented in this dissertation, the derivatives of link-flow proportions with respect to the dynamic demand flows are explicitly included in the estimation procedure. Therefore, it is expected that the solution will be closer to a Stackelberg game solution.

The experiments by Yang (1995) on a few sample small networks indicate that, in general, both algorithms have similar performance. In terms of RMSE or value of the objective function, both algorithms have comparable results after almost the same number of iterations. But if convergence is defined in terms of maximum difference between OD values, the SAB algorithm reaches convergence in fewer iterations, which indicates that this approach is more stable. The experiment results also confirm the need for scaling the target demand matrix based on the total observed OD flows, introduced originally by Ortuzar and Willumsen (1990).

2.4. Summary

In this chapter, different methods for estimating OD flows from traffic counts in the static and dynamic cases were reviewed. Special attention was paid to several essential research works by Yang, Cascetta, Inaudi and Marquis whose approaches are similar to the methods presented in this dissertation. A detailed and critical review of each method is presented and the distinctions of the methods adopted in this dissertation are pointed out. In this dissertation, by adopting an iterative bi-level optimization method in the dynamic case, the simplifying assumption of the proportional assignment is dropped. By explicitly including the derivatives of link-flow proportions with respect to demand, a Stackelberg solution to the theoretical game problem is sought. To exploit the *a priori* information on the OD flows, Bayes' theorem is used. Furthermore, in the case of the sequential (or rolling-horizon) estimation of OD flows, a new formulation is presented. The formulation is different from the previous works due to the assumptions made on the initial state of the system at the beginning of each estimation stage.

CHAPTER 3. PROBLEM FORMULATION

3.1. Introduction

In this chapter, the general formulation of the problem, the underlying assumptions and the methods to solve the problem are presented. As explained before, the objective of this research is to estimate time-dependent OD flows in a dynamic transportation network, which when assigned to the network produce link volumes that are as close as possible to the observed values. DYNASMART-P, which is a simulation-based Dynamic Traffic Assignment (DTA) program developed at the University of Texas at Austin, is used for the simulation of flow and assignment of the estimated OD-flows to the network. The parameters employed in the optimization problem, namely the link-flow proportions, are themselves a function of the unknown OD demand. Therefore, given an OD flow matrix, the DTA program is used to estimate the link-flow proportions. The assignment problem can be formulated as an optimization problem, which may be included in the set of constraints to the main optimization problem, resulting in a bi-level optimization formulation.

First, the notation used throughout the chapter is introduced and the relevant time intervals are defined. In Section 3.3, the common unconstrained problem, i.e. the upper-level problem, is formulated and the associated error terms are described. The assumptions used to make the problem over-specified are also explained. In the same section, the ordinary and generalized least-squares solutions to the upper-level optimization problem are discussed. In Section 3.4, the algorithm for the iterative bi-level generalized least-square (Bi-GLS) optimization for a quasi-linear formulation is described.

In Section 3.5, the link-flow derivatives are introduced in the optimization problem solution. The problem is first solved by using the ordinary non-linear optimization formulation, and then extended to the generalized non-linear cases. In section 3.7, the formulation of the bi-level non-linear problem (Bi-NLP) is introduced. In section 3.8, the characteristics of the bi-level optimization problem in the context of theoretical games are reviewed.

The constrained optimization problems are discussed separately. Section 3.9 deals with constrained optimization in the single-horizon case. The problem is further extended to the context of dynamic traffic assignment where OD flows should be estimated over rolling-horizon windows. Two scenarios are considered. 1) fixed initial point where the initial condition in each estimation period is dictated by the results of the previous estimation periods, and 2) free-initial point where the initial condition is not fixed and is governed by the instantaneous state of the system in the real network at the start of each estimation period. In closing, a summary of the chapter is presented.

3.2. Definitions

In the following list of variables, the capital bold letters represent matrices and the lowercase letters denote the elements of the matrices or scalar variables. This convention is followed throughout the text unless explicitly mentioned otherwise.

t	index for the observation or reporting intervals, during which the traffic volume is accumulated and reported, $t = 1,\ldots,T$.
s	index for short departure intervals (the same size as observation intervals).
l	index number for links with traffic flow measurements, $l=1,\ldots,L$.

τ	(*tau*) index for aggregation intervals, τ=1,…,Γ. Each aggregation interval encompasses one or several departure intervals.
n_τ	number of observation intervals in aggregation interval τ (assumed equal for all aggregation intervals).
i	denotes the origin zone number, i=1,…,I.
j	denotes the destination zone number, j=1,…,J.
T	number of observation intervals in the estimation period.
L	number of links in the network that have flow measurements.
Γ	(*Gamma*) number of aggregation intervals in the estimation period.
I	number of origin zones in the network.
J	number of destination zones in the network.
$\boldsymbol{v}_{(l,t)}$	actual traffic volume on link l, during observation interval t.
$\boldsymbol{c}_{(l,t)}$	measured or "sensed" traffic volume on link l, during observation interval t.
$\boldsymbol{d}_{(s,i,j)}$	unknown demand volume in number of vehicles originating their trip at zone i during departure interval s with destination zone j.
$\boldsymbol{d}_{(\tau,i,j)}$	aggregated demand volume in number of vehicles with destination in zone j, originating their trip at zone i during aggregation interval τ.
$\boldsymbol{d}^*_{(\tau,i,j)}$	optimal estimate of the aggregated demand volume.
$\boldsymbol{p}_{(l,t),(s,i,j)}$	link-flow proportions, that is the proportion of demand $\boldsymbol{d}_{(s,i,j)}$ that flows onto link l during observation interval t; (l,t) denotes the index for the rows and (s,i,j) is the index for the columns of the matrix.
$\boldsymbol{p}_{(l,t),(\tau,i,j)}$	aggregated link-flow proportions, that is the proportion of aggregated demand flow $\boldsymbol{d}_{(\tau,i,j)}$ that flows onto link l during observation interval t,.
$\hat{\boldsymbol{p}}_{(l,t),(\tau,i,j)}$	estimate of aggregated link-flow proportions resulting from the simulator.

$\omega_{(l,t)}$ — (*omega*) is the error in traffic flow measurements on link l during observation interval t due to equipment, environment or other sources.

$\eta_{(l,t)(s,i,j)}$ — (*eta*) errors due to substitution of estimated link-flow proportion, $\hat{p}_{(l,t),(s,i,j)}$, obtained from simulation for actual values of $p_{(l,t),(s,i,j)}$.

$\zeta_{(s,i,j)}$ — (*zeta*) is the deviation of demand flow $d_{(s,i,j)}$, from the average demand flow departing during aggregation interval τ that encompasses s.

$\varepsilon_{(l,t)}$ — (*epsilon*) is the combined error terms in estimation of traffic volume on link l during observation interval t (this is the overall effect of ω, η, and ζ and their interactions).

$e_{(l,t)}$ — observed error (residual) which is an estimate of $\varepsilon_{(l,t)}$.

$\boldsymbol{E}$ — (Epsilon) the vector of combined error terms

$\boldsymbol{E}$ — the vector of combined residuals (estimates of errors)

$\boldsymbol{V}$ — vector of actual flows on the links.

$\boldsymbol{C}$ — vector of measured flows on the links.

$\boldsymbol{D}$ — vector of aggregated OD demand flows consisting of elements $d_{(\tau,i,j)}$.

$\boldsymbol{D}^*$ — optimal estimate of vector of aggregated OD flows.

$\hat{\boldsymbol{P}}$ — matrix of estimated link-flow proportions (obtained from simulation) with L.T number of rows and Γ.I.J number of columns.

$\boldsymbol{W}$ — variance-covariance matrix of the combined error terms $\varepsilon_{(l,t)}$.

Figure 3.1 depicts the definition of the time intervals used in the formulations. The estimation period (stage) consists of T observation intervals. Several observation intervals t collectively make up an aggregate departure interval τ. The objective is to estimate the OD flows departing during aggregate departure intervals τ, given the traffic flow values during observation intervals t.

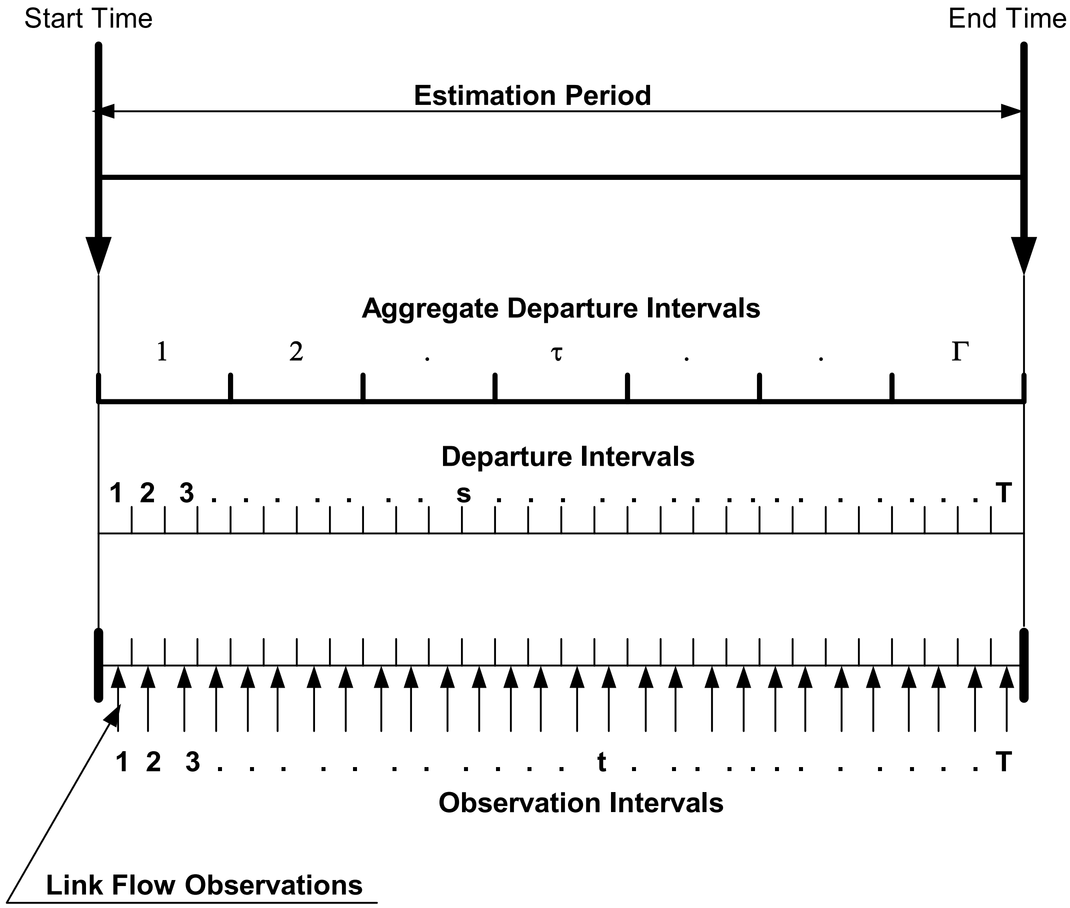


Figure 3.1. Definition of time intervals and OD-flow estimation stage

Link-flow proportions are presented in a two-dimensional matrix as shown below. For clarity here $p_{(l,t)(\tau,i,j)}$ is denoted by $p_{(\tau,k)}^{(l,t)}$, where OD pair k is substituted for OD pair (i,j).

$$
\boldsymbol{P} = \begin{bmatrix}
p_{1,1}^{1,1} & p_{1,2}^{1,1} & \cdots & p_{1,K}^{1,1} & \cdots & \cdots & p_{\Gamma,1}^{1,1} & p_{\Gamma,2}^{1,1} & \cdots & p_{\Gamma,K}^{1,1} \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\
p_{1,1}^{L,1} & p_{1,2}^{L,1} & \cdots & p_{1,K}^{L,1} & \cdots & \cdots & p_{\Gamma,1}^{L,1} & p_{\Gamma,2}^{L,1} & \cdots & p_{\Gamma,K}^{L,1} \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & \vdots & & p_{(\tau,k)}^{(l,t)} & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\
p_{1,1}^{1,T} & p_{1,2}^{1,T} & \cdots & p_{1,K}^{1,T} & \cdots & \cdots & p_{\Gamma,1}^{1,T} & p_{\Gamma,1}^{1,T} & \cdots & p_{\Gamma,1}^{1,T} \\
\vdots & \vdots & \cdots & \vdots & & & \vdots & \vdots & & \vdots \\
p_{1,1}^{L,T} & p_{1,2}^{L,T} & \cdots & p_{1,K}^{L,T} & \cdots & \cdots & p_{\Gamma,1}^{L,T} & p_{\Gamma,2}^{L,T} & \cdots & p_{\Gamma,K}^{L,T}
\end{bmatrix}
\begin{matrix}
\textit{link } 1 & \\
\vdots & t=1 \\
\textit{link } L & \\
\vdots & \vdots \\
\vdots & \vdots \\
\vdots & \vdots \\
\textit{link } 1 & \\
\vdots & t=T \\
\textit{link } L &
\end{matrix}
$$

$$
\begin{array}{l|ccccc|c|ccc|c|ccccc|}
 & \vdots & \vdots & & \vdots & & & & \vdots & & & \vdots & \vdots & & \vdots & \\
\textit{OD pair} & 1 & 2 & \cdots & K & & & \cdots & k & \cdots & & 1 & 2 & \cdots & K & \\
\textit{dep. int.} & & \cdots & 1 & \cdots & & & \cdots & \tau & \cdots & & & \cdots & \Gamma & \cdots &
\end{array}
$$

The OD flow matrix is a column matrix of the form:

$$\begin{bmatrix} d_{1,1} \\ \vdots \\ d_{\Gamma,1} \\ \vdots \\ d_{\tau,k} \\ \vdots \\ d_{1,K} \\ \vdots \\ d_{\Gamma,K} \end{bmatrix}$$

Link flow observations are represented in a column matrix of the form:

$$\begin{bmatrix} c_{1,1} \\ \vdots \\ c_{1,T} \\ \vdots \\ \vdots \\ c_{l,t} \\ \vdots \\ \vdots \\ \vdots \\ c_{L,1} \\ \vdots \\ c_{L,T} \end{bmatrix}$$

With the above definitions, equation (1.5) can be rewritten in the following form.

$$
\begin{bmatrix}
p_{1,1}^{1,1} & p_{1,2}^{1,1} & \cdots & p_{1,K}^{1,1} & \cdots & \cdots & p_{\Gamma,1}^{1,1} & p_{\Gamma,2}^{1,1} & \cdots & p_{\Gamma,K}^{1,1} \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\
p_{1,1}^{L,1} & p_{1,2}^{L,1} & \cdots & p_{1,K}^{L,1} & \cdots & \cdots & p_{\Gamma,1}^{L,1} & p_{\Gamma,2}^{L,1} & \cdots & p_{\Gamma,K}^{L,1} \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\
\vdots & \vdots & \vdots & \vdots & & p_{(\tau,k)}^{(l,t)} & \vdots & \vdots & \vdots & \vdots \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & & \vdots \\
p_{1,1}^{1,T} & p_{1,2}^{1,T} & \cdots & p_{1,K}^{1,T} & \cdots & \cdots & p_{\Gamma,1}^{1,T} & p_{\Gamma,1}^{1,T} & \cdots & p_{\Gamma,1}^{1,T} \\
\vdots & \vdots & \cdots & \vdots & & & \vdots & \vdots & & \vdots \\
p_{1,1}^{L,T} & p_{1,2}^{L,T} & \cdots & p_{1,K}^{L,T} & \cdots & \cdots & p_{\Gamma,1}^{L,T} & p_{\Gamma,2}^{L,T} & \cdots & p_{\Gamma,K}^{L,T}
\end{bmatrix}
\times
\begin{bmatrix}
d_{1,1} \\ \vdots \\ d_{\Gamma,1} \\ \vdots \\ d_{\tau,k} \\ \vdots \\ d_{1,K} \\ \vdots \\ d_{\Gamma,K}
\end{bmatrix}
=
\begin{bmatrix}
c_{1,1} \\ \vdots \\ c_{1,T} \\ \vdots \\ \vdots \\ c_{l,t} \\ \vdots \\ \vdots \\ \vdots \\ c_{L,1} \\ \vdots \\ c_{L,T}
\end{bmatrix}
+
\begin{bmatrix}
\varepsilon_{1,1} \\ \vdots \\ \varepsilon_{1,T} \\ \vdots \\ \vdots \\ \varepsilon_{l,t} \\ \vdots \\ \vdots \\ \vdots \\ \varepsilon_{L,1} \\ \vdots \\ \varepsilon_{L,T}
\end{bmatrix}
$$

It will be shown that in the upper-level optimization problem, the demand flows can be estimated by finding the least-squares estimate to the above over-specified set of simultaneous equations.

Next, we formulate the upper-level optimization problem with no constraints.

3.3. Formulation of Unconstrained Problem

First, the actual traffic volume observed on link l during time interval t is related to the OD volumes of interest using the link-flow proportions, resulting in the following definitional equation:

$$\boldsymbol{v}_{(l,t)} = \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J} \boldsymbol{p}_{(l,t)(s,i,j)} . \boldsymbol{d}_{(s,i,j)} \tag{3.1}$$

Since true values of link-flow proportions are difficult to find, we substitute them with their estimate resulting from a dynamic assignment simulation program. Hence, some errors are introduced, that is:

$$\boldsymbol{p}_{(l,t)(s,i,j)} = \hat{\boldsymbol{p}}_{(l,t)(s,i,j)} + \boldsymbol{\eta}_{(l,t)(s,i,j)} \tag{3.2}$$

Substituting equation (3.2) in (3.1):

$$\boldsymbol{v}_{(l,t)} = \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J} \hat{\boldsymbol{p}}_{(l,t)(s,i,j)} . \boldsymbol{d}_{(s,i,j)} + \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J} \boldsymbol{\eta}_{(l,t)(s,i,j)} . \boldsymbol{d}_{(s,i,j)} \tag{3.3}$$

There are also some errors in traffic volume measurements, i.e.

$$\boldsymbol{c}_{(l,t)} = \boldsymbol{v}_{(l,t)} + \boldsymbol{\omega}_{(l,t)} \tag{3.4}$$

From (3.3) and (3.4), one obtains

$$\boldsymbol{c}_{(l,t)} = \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J} \hat{\boldsymbol{p}}_{(l,t)(s,i,j)} . \boldsymbol{d}_{(s,i,j)} + \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J} \boldsymbol{\eta}_{(l,t)(s,i,j)} . \boldsymbol{d}_{(s,i,j)} + \boldsymbol{\omega}_{(l,t)} \tag{3.5}$$

To make the problem over-specified, we will estimate the aggregated demand flows over longer departure intervals than observation intervals. In general, there are Γ.I.J unknowns, while there are L.T equations. If there are not enough links with flow measurements in the network, we increase the length of departure intervals such that

$$\frac{\Gamma}{\boldsymbol{T}} \leq \frac{\boldsymbol{L}}{\boldsymbol{I} \times \boldsymbol{J}}$$

In other words, the ratio of the *lengths* of departure intervals to observation intervals should be *greater* than (I×J)/L, that is:

$$\frac{|\tau|}{|\boldsymbol{t}|} \geq \frac{\boldsymbol{I} \times \boldsymbol{J}}{\boldsymbol{L}} \qquad (3.6)$$

where $|\tau|$ denotes the length of departure intervals, say, in minutes, and $|t|$ represents the length of observation intervals in the same time unit.

Indeed, deployment of Intelligent Transportation Systems (ITS) can alleviate the under-specification of the problem. For example, if there are 2500 origin-destination pairs in a network, flow measurements are reported in 30-second intervals, and data are collected on 250 links, the shortest departure interval that makes the problem over-specified is 5 minutes. In general, it is recommended to choose longer departure intervals to increase the over-specification of the problem and to reduce errors due to short-term variation and inherent randomness in the system.

It should be noted that in considering the links with measurements, we ignore the measurements from detector sites between which there are no points of entry or exit of traffic. That is, if two detector sites are located on a link with no intermediate intersections or entry/exit ramps, traffic measurements at the two sites would be highly correlated and would not provide any extra information or equation.

Aggregating demand over longer time intervals also introduces some errors. Demand flow in each observation interval is equal to the average demand flow during departure interval τ, plus a random perturbation, that is

$$\begin{aligned} d_{(s,i,j)} &= \frac{\sum_{s\in\tau} d_{(s,i,j)}}{n_\tau} + \zeta_{(s,i,j)} \\ &= \frac{d_{(\tau,i,j)}}{n_\tau} + \zeta_{(s,i,j)} \end{aligned} \tag{3.7}$$

where $s \in \tau$ denotes the departure intervals contained in the aggregate departure interval τ.

In equation (3.7), $\sum_{s\in\tau} d_{(s,i,j)}$ is replaced by $d_{(\tau,i,j)}$, and $\zeta_{(s,i,j)}$ denotes the deviation of demand flow generated during departure interval s from its mean value during the aggregate interval τ as shown in Figure 3.2.

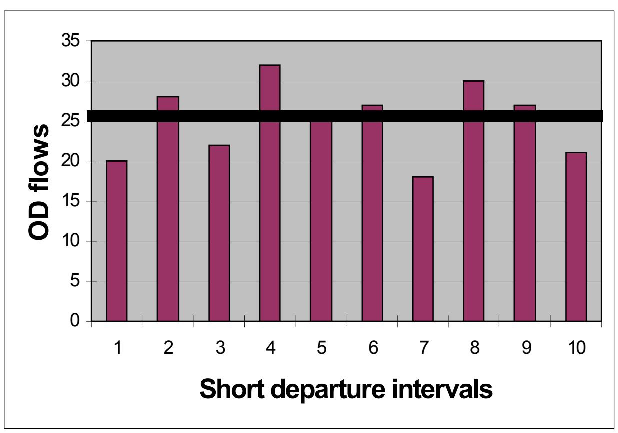


Figure 3.2. Induced errors due to aggregation of departure intervals

By substituting equation (3.7) in (3.5), one obtains

$$c_{(l,t)} = \sum_{\tau=1}^{\tau_t}\sum_{i=1}^{I}\sum_{j=1}^{J}\hat{p}_{(l,t)(\tau,i,j)}.d_{(\tau,i,j)} + \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J}\eta_{(l,t)(s,i,j)}.d_{(s,i,j)} + \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J}\hat{p}_{(l,t)(s,i,j)}.\zeta_{(s,i,j)} + \omega_{(l,t)} \tag{3.8}$$

where $\hat{p}_{(l,t)(\tau,i,j)} = \frac{\sum_{s\in\tau}\hat{p}_{(l,t)(s,i,j)}}{n_\tau}$, and τ_t is the index of the aggregate departure interval which encompasses the observation interval t.

Equation (3.8) represents the general non-linear relation between traffic volume counts and the unknown demand flow including the confounded error terms. This relation shows that due to non-linearity, the combined error is not white noise. We assume that the sum of error terms and the interaction terms has a normal distribution with zero mean and unknown variance, but we will keep in mind that the existing interactions in the error terms are ignored, introducing a possible limitation on the performance of the estimation methods. Moreover, since we are ignoring the existence of the interaction terms, the estimator may not be efficient (by definition an estimator is efficient if it has the lowest variance among all unbiased estimators).

We denote the combined error term by $\varepsilon_{(l,t)}$,

$$\varepsilon_{(l,t)} = \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J}\eta_{(l,t)(s,i,j)}.d_{(s,i,j)} + \sum_{s=1}^{t}\sum_{i=1}^{I}\sum_{j=1}^{J}\hat{p}_{(l,t)(s,i,j)}.\zeta_{(s,i,j)} + \omega_{(l,t)}$$

Therefore, the combined error term, is a non-linear combination of the errors due to the following sources:

- Substituting the estimated link-flow proportions. This error includes the errors due to: 1) inconsistencies in traffic assignment assumptions, 2) inconsistencies in flow-propagation assumptions, and 3) inconsistencies in traffic control assumptions.
- Substituting the *average* aggregated demand flows for the varying demand flows within each departure interval.
- Measurement of traffic volumes (sensor errors).

Therefore, in the unconstrained formulation of the problem, we seek to find the demand flows that minimize the sum of squared errors in equation (3.9), or conventionally, the least-squares estimator of the set of equations:

$$\boldsymbol{c}_{(l,t)} = \sum_{\tau=1}^{\tau_t} \sum_{i=1}^{I} \sum_{j=1}^{J} \hat{\boldsymbol{p}}_{(l,t)(\tau,i,j)} . \boldsymbol{d}_{(\tau,i,j)} + \varepsilon_{(l,t)} \qquad \forall (\boldsymbol{l}, \boldsymbol{t}) \tag{3.9}$$

Equation (3.9) can be represented in matrix form as

$$\boldsymbol{C} = \hat{\boldsymbol{P}}.\boldsymbol{D} + E \tag{3.10}$$

3.3.1. Ordinary Least-Squares Estimation

We consider the following assumptions for the error terms E in equation (3.10)

$$\mathrm{E}(\boldsymbol{E})=0$$
$$\mathrm{E}(\boldsymbol{E}\,\boldsymbol{E}^{\mathrm{T}})=\sigma^2 \boldsymbol{I} \tag{3.11}$$

where E(.) denotes the expected value of any variable, $\boldsymbol{I}$ is the identity matrix of size L.T×L.T, and σ^2 is the variance which is a known scalar value.

Under the assumptions on error terms in (3.11), the least-squares estimates of the demand flows (computed below) will be unbiased with the smallest variance (known as an efficient estimator).

To solve the unconstrained problem, we minimize the sum of squared residuals; i.e. we seek the optimum demand value $\boldsymbol{D}^*$ that solves the following optimization problem. To avoid notational confusion, because of using the *estimate* of the link-flow proportion obtained from simulation ($\hat{\boldsymbol{P}}$), we will use $\boldsymbol{D}$ to denote the estimate of demand as well as its true value.

$$\begin{aligned} \boldsymbol{Min} \quad \boldsymbol{Z}(\boldsymbol{D}) &= \sum_{\boldsymbol{k}=1}^{\boldsymbol{L.T}} \boldsymbol{e}_{\boldsymbol{k}}^2 = \boldsymbol{E}^T \boldsymbol{E} \\ &= (\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D})^T (\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D}) \\ &= \boldsymbol{C}^T \boldsymbol{C} - 2\boldsymbol{D}^T \hat{\boldsymbol{P}}^T \boldsymbol{C} + \boldsymbol{D}^T \hat{\boldsymbol{P}}^T \hat{\boldsymbol{P}} \boldsymbol{D} \end{aligned} \qquad (3.12)$$

which follows from noting that $\boldsymbol{D}^T \hat{\boldsymbol{P}}^T \boldsymbol{C}$ is a scalar and thus equal to its transpose $\boldsymbol{C}^T \hat{\boldsymbol{P}} \boldsymbol{D}$.

To find a closed-form solution to the above unconstrained optimization problem, we assume that the link-flow proportions are constant. Therefore, to find the value of $\boldsymbol{D}$ that minimizes the sum of squared residuals we differentiate (3.12) with respect to $\boldsymbol{D}$ and equate it to zero (Johnston, 1972). This will yield the conventional least-squares estimate of OD flows in equation (3.10).

$$\frac{\partial}{\partial \boldsymbol{D}}(\boldsymbol{E}^T \boldsymbol{E}) = -2\hat{\boldsymbol{P}}^T \boldsymbol{C} + 2\hat{\boldsymbol{P}}^T \hat{\boldsymbol{P}} \boldsymbol{D} = 0$$

If the rank of $\hat{\boldsymbol{P}}$ is greater than the number of unknown demand flows, then

$$\boldsymbol{D}^* = \left(\hat{\boldsymbol{P}}^T \hat{\boldsymbol{P}}\right)^{-1} \left(\hat{\boldsymbol{P}}^T \boldsymbol{C}\right) \quad (3.13)$$

3.3.2. <u>Generalized Least-Squares Estimation</u>

Now we relax one of the constraints assumed for the error terms $\boldsymbol{E}$, that is

$$\mathrm{E}(\boldsymbol{E})=0$$
$$\mathrm{E}(\boldsymbol{E}\,\boldsymbol{E}^{\mathrm{T}})=\sigma^2 \boldsymbol{W} \quad (3.14)$$

where $\boldsymbol{W}$ is a known symmetric, positive-definite matrix of size L.T×L.T.

Any positive definite matrix can be expressed in the form of $\boldsymbol{F}.\boldsymbol{F}^{\mathrm{T}}$, where $\boldsymbol{F}$ is nonsingular. So we can write (Johnston, 1972):

$$\boldsymbol{W} = \boldsymbol{F}.\boldsymbol{F}^T$$

so that

$$\boldsymbol{F}^{-1}.\boldsymbol{W}.\boldsymbol{F}^{-1T} = \boldsymbol{I} \quad (3.15)$$

Pre-multiplying the model $\boldsymbol{C} = \hat{\boldsymbol{P}}.\boldsymbol{D} + \boldsymbol{E}$ by $\boldsymbol{F}^{1}$ gives

$$\boldsymbol{C}' = \hat{\boldsymbol{P}}'.\boldsymbol{D} + \boldsymbol{E}' \quad (3.16)$$

where

$$\boldsymbol{C}' = \boldsymbol{F}^{-1}.\boldsymbol{C}, \qquad \hat{\boldsymbol{P}}' = \boldsymbol{F}^{-1}.\hat{\boldsymbol{P}}, \quad \text{and} \quad \boldsymbol{E}' = \boldsymbol{F}^{-1}.\boldsymbol{E} \quad (3.17)$$

Using (3.14), (3.15) and (3.17), it is easily seen that

$$\mathrm{E}(E'.E'^{T}) = \sigma^2 \boldsymbol{I}$$

Therefore, equation (3.16) satisfies all the assumptions required for the ordinary least-squares estimation mentioned in equation (3.11). By substituting the transformations (3.17) in equation (3.13), we get:

$$\boldsymbol{D}^* = \left(\hat{\boldsymbol{P}}^T \boldsymbol{W}^{-1} \hat{\boldsymbol{P}}\right)^{-1} \left(\hat{\boldsymbol{P}}^T \boldsymbol{W}^{-1} \boldsymbol{C}\right) \tag{3.18}$$

3.4. Iterative Bi-level Generalized Least-Squares Estimation (Bi-GLS)

As mentioned in the previous sections, to find the least-squares estimates of the demand flows, link-flow proportions should be calculated, but in a dynamic assignment and particularly in congested networks, the link-flow proportions are not constant, and are themselves a function of unknown time-dependent demand flows.

To address this problem, link-flow proportions should be estimated analytically, or numerically by use of a dynamic simulation program, for a given demand flow. Then, given the link-flow proportions, the generalized least-squares estimates of the demand flows are found. The process is repeated for a pre-specified number of times or until a convergence criterion is met.

To find the link-flow proportions, in each iteration, the presumed demand should be assigned onto the network. To assign the traffic to the network, users' path choices should be replicated as closely as possible. There are several assignment rules (Sheffi, 1985) but there is no evidence that users' actual path choice conforms to any of them. Furthermore, in the context of providing real-time information and route guidance to drivers, there likely exist multiple user classes (MUC) that have varying degrees of information availability (Peeta and Mahmassani, 1995a).

Peeta and Mahmassani (1995b) have formulated the user-equilibrium (UE) and system-optimal (SO) time-dependent traffic assignment problems. UE and SO procedures are integral components of DYNASMART-P, which is used as a tool in this research to solve the lower-level assignment problem resulting in time-varying link flows and link-flow proportions, given the demand flows.

The resulting bi-level formulation is similar to a game with two (groups of) players, each trying to optimize its own objective function (Fisk, 1984). The equilibrium solution is the point at which optimality conditions for both lower and upper level problems are satisfied simultaneously. In user-equilibrium assignment, the lower player itself consists of a group of players each trying to minimize his own travel time; while in system optimal the supplier/manager of the transportation network assigns the traffic in a way to minimize the total travel time in the network.

The schematic flow chart of the proposed process is shown in Figure 2. If a historical OD table is available, it could be incorporated in the formulation (Cascetta *et al.*, 1993), or used as *a priori* information in a Bayesian inference scheme (Maher, 1983). We prefer the separate Bayesian inference implementation, because it can be implemented independently from the OD-flow estimation method.

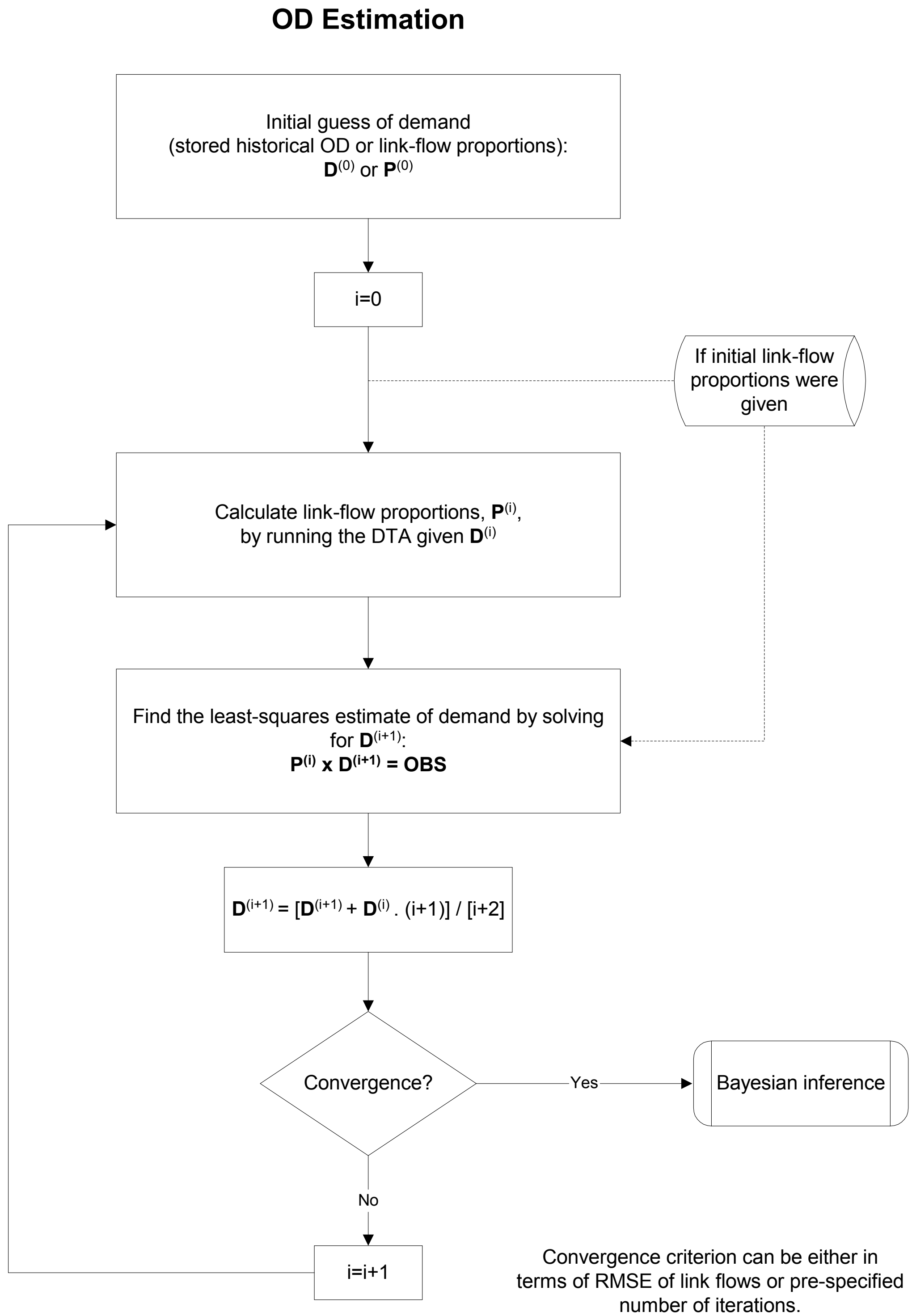


Figure 3.3. Flow-chart of the proposed bi-level optimization OD-flow estimation

3.5. Ordinary Non-Linear Optimization Formulation

This formulation is used in the upper-level problem and is an extension to the least-squares estimation method. However, in this formulation, the dependency of link-flow proportions on the demand is explicitly considered in taking derivatives to solve the upper-level optimization problem. Since we cannot find the derivative of link-flow proportions analytically, we use the simulation program to approximate the derivatives numerically.

On the other hand, the value of the derivative of link-flow proportions with respect to the OD demand is dependent on demand values. Moreover, link-flow proportions are not a continuous function of demand. Thus, we should still solve the problem in two levels: minimizing the deviation of flows in the upper level, and finding link-flow proportions and its derivatives with respect to demand in the lower level.

We repeat here equation (3.10):

$$\boldsymbol{C} = \hat{\boldsymbol{P}}.\boldsymbol{D} + E$$

and assume the following for the error terms E:

$$\mathrm{E}(\boldsymbol{E})=0$$
$$\mathrm{E}(\boldsymbol{E}\,\mathrm{E}^{\mathrm{T}})=\sigma^2\boldsymbol{I}_{\mathrm{L.T}}$$

Since these assumptions are similar to the ordinary least-squares assumptions, we call the formulation that will be introduced as *ordinary* bi-level non-linear optimization formulation.

We need to minimize the sum of squared residuals, that is to find ***D*** in (3.10) that minimizes the sum of squares of errors:

$$\begin{aligned} \boldsymbol{Min} \quad \boldsymbol{Z}(\boldsymbol{D}) &= \sum_{\boldsymbol{k}=1}^{\boldsymbol{L.T}} \boldsymbol{e}_{\boldsymbol{k}}^2 = \boldsymbol{E}^{\boldsymbol{T}} \boldsymbol{E} \\ &= (\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D})^{\boldsymbol{T}} (\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D}) \end{aligned}$$

We differentiate the above equation with respect to ***D***, this time taking into account that link-flow proportions are a function of demand flows. Setting the derivative equal to zero:

$$\frac{\partial \boldsymbol{Z}(\boldsymbol{D})}{\partial \boldsymbol{d}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}} = -2\left[\frac{\partial \boldsymbol{D}^{\boldsymbol{T}}}{\partial \boldsymbol{d}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}} \hat{\boldsymbol{P}}^{\boldsymbol{T}} + \boldsymbol{D}^{\boldsymbol{T}} \frac{\partial \hat{\boldsymbol{P}}^{\boldsymbol{T}}}{\partial \boldsymbol{d}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}}\right]\left[\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D}\ \right] = 0 \qquad \forall(\boldsymbol{\nu}, \mathrm{o}, \mathrm{d}) \tag{3.19}$$

It should be noted that for each (ν,o,d) equation (3.19) results in a scalar value which is quadratic in terms of time dependent demand flows.

In equation (3.19), $\frac{\partial \boldsymbol{D}^{\boldsymbol{T}}}{\partial \boldsymbol{d}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}}$ is a row vector of zero elements, except the entry pertaining to the demand flow from *o* to *d* departing during departure interval *ν,* which is equal to one. Therefore $\frac{\partial \boldsymbol{D}^T}{\partial \boldsymbol{d}_{(v,o,d)}} \hat{\boldsymbol{P}}^T$ is a row vector (1×L.T) denoting the proportions of demand flow $\boldsymbol{d}_{(v,o,d)}$ on all links during each observation interval.

A term ignored in most of the previous studies is $\boldsymbol{D}^T \frac{\partial \hat{\boldsymbol{P}}^T}{\partial \boldsymbol{d}_{(v,o,d)}}$. As mentioned before, the link-flow proportions are not constant and are themselves a function of the (unknown) demand flows and users' route-choice behavior. In congested dynamic networks, a change in any demand flow, $\boldsymbol{d}_{(v,o,d)}$, can change the route choice of other users, who have started their trips before or even after departure interval v. Since finding a closed-form relation between link-flows, and consequently link-flow proportions, and demand flows in a dynamic network is not feasible, we will use a simulation program to estimate the partial derivatives of the link-flow proportions with respect to demand. Estimation of derivatives in a large network is very time consuming since they should be recomputed at different demand levels and for each OD pair and departure interval.

If the above term is included in the equations, it will result in a set of L.T simultaneous quadratic equations. Numerical methods can be used to solve this set of equations. By using the following notational definition

$$\hat{\boldsymbol{P}}^T_{(v,o,d)} = \frac{\partial \boldsymbol{D}^T}{\partial \boldsymbol{d}_{(v,o,d)}} \hat{\boldsymbol{P}}^T \qquad \text{(a row vector of size 1×L.T)}$$

$$\nabla \hat{\boldsymbol{P}}^T{}_{(v,o,d)} = \frac{\partial \hat{\boldsymbol{P}}^T}{\partial \boldsymbol{d}_{(v,o,d)}} \qquad \text{(a matrix of size Γ.I.J×L.T)}$$

one can rewrite the equation (3.19) as

$$\left[\hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})} + \boldsymbol{D}^{\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})}\right]\left[\hat{\boldsymbol{P}}\boldsymbol{D} \ - \boldsymbol{C}\right] = 0$$

or

$$\boldsymbol{D}^{\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})}\hat{\boldsymbol{P}}\boldsymbol{D} \ = \left[\boldsymbol{C}^{\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}_{(\nu,\boldsymbol{o},\boldsymbol{d})} - \hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})}\hat{\boldsymbol{P}}\right]\boldsymbol{D} \ + \hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})}\boldsymbol{C} \qquad \forall\ (\nu,o,d) \tag{3.20}$$

Note that the above equation results in a scalar quadratic relation in terms of unknown OD values (as shown below). If the following notation is used:

$$\boldsymbol{A}^{(\nu,\boldsymbol{o},\boldsymbol{d})} = \nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})}\hat{\boldsymbol{P}} \qquad \text{(a } \Gamma\text{.I.J} \times \Gamma\text{.I.J matrix)}$$

$$\boldsymbol{B}^{(\nu,\boldsymbol{o},\boldsymbol{d})} = \left[\boldsymbol{C}^{\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}_{(\nu,\boldsymbol{o},\boldsymbol{d})} - \boldsymbol{P}^{\boldsymbol{T}}_{(\nu,\boldsymbol{od})}\hat{\boldsymbol{P}}\right] \qquad \text{(a } 1 \times \Gamma\text{.I.J matrix)}$$

$$\boldsymbol{g}^{(\nu,\boldsymbol{o},\boldsymbol{d})} = \hat{\boldsymbol{P}}^{\boldsymbol{T}}_{(\nu,\boldsymbol{o},\boldsymbol{d})}\boldsymbol{C} \qquad \text{(a scalar value)}$$

we can rewrite equation (3.20) as:

$$\boldsymbol{D}^{\boldsymbol{T}}\boldsymbol{A}^{(\nu,\boldsymbol{o},\boldsymbol{d})}\boldsymbol{D} \ = \boldsymbol{B}^{(\nu,\boldsymbol{o},\boldsymbol{d})}\boldsymbol{D} \ + \boldsymbol{g}^{(\nu,\boldsymbol{o},\boldsymbol{d})} \qquad \forall\ (\nu,o,d) \tag{3.21}$$

For simplicity of notation, we use a sequential number for each element of the time dependent OD pair, that is, if:

$$\boldsymbol{m} = (\tau, \boldsymbol{i}, \boldsymbol{j})$$

and

$$\boldsymbol{n} = (\tau, \boldsymbol{i}, \boldsymbol{j})'$$

then, equation (3.21) can be written in a scalar form:

$$\sum_{m=1}^{\Pi J}\sum_{n=1}^{\Pi J} \boldsymbol{a}_{mn}^{(\nu,o,d)} \boldsymbol{d}_m \boldsymbol{d}_n = \sum_{m=1}^{\Pi J} \boldsymbol{b}_m^{(\nu,o,d)} \boldsymbol{d}_m + \boldsymbol{g}^{(\nu,o,d)} \qquad \forall(\nu,o,d) \qquad (3.22)$$

where $\boldsymbol{a}_{mn}^{(\nu,o,d)}$ and $\boldsymbol{b}_m^{(\nu,o,d)}$ are the elements of matrices $\boldsymbol{A}^{(\nu,o,d)}$ and $\boldsymbol{B}^{(\nu,o,d)}$ respectively.

The set of equations in (3.20) can be written in the general form of a fixed-point problem, $\boldsymbol{D} = \boldsymbol{f}(\boldsymbol{D})$.

If one ignores that the link-flow proportions are a function of the demand, say in a static case and in uncongested conditions, the conventional least-squares estimator of demand will be obtained. The resulting equation is the same as the conventional ordinary least-squares estimation of demand flow from traffic counts shown in (3.13). That is, if one assumes $\nabla \hat{\boldsymbol{P}}_{(\nu,o,d)}^{\boldsymbol{T}} = 0$, equation (3.18) is simplified to:

$$\boldsymbol{D}^* = \left(\hat{\boldsymbol{P}}^{\boldsymbol{T}}.\hat{\boldsymbol{P}}\right)^{-1}\left(\hat{\boldsymbol{P}}^{\boldsymbol{T}}.\boldsymbol{C}\right)$$

3.6. Generalized Non-Linear Optimization Formulation

In the general case, if there is some correlation in the observed data, we can relax some of the assumptions on the error terms and assume the following relations:

$$\mathrm{E}(\boldsymbol{E})=0$$
$$\mathrm{E}(\boldsymbol{E}\,\boldsymbol{E}^{\mathrm{T}})=\sigma^2\boldsymbol{W} \tag{3.23}$$

where $\boldsymbol{W}$ is a known symmetric, positive-definite matrix.

It should be pointed out again that $\boldsymbol{E}$ denotes the combined effect of residuals due to errors in measurement, errors in estimation of link-flow proportions, and residuals due to aggregation of demand flows over several short departure intervals.

Any positive definite matrix can be expressed in the form of $\boldsymbol{F}.\boldsymbol{F}^{\mathrm{T}}$, where $\boldsymbol{F}$ is nonsingular. So we can write

$$\boldsymbol{W} = \boldsymbol{F}.\boldsymbol{F}^{\boldsymbol{T}} \tag{3.24}$$

so that

$$\boldsymbol{F}^{-1}.\boldsymbol{W}.\boldsymbol{F}^{-1\boldsymbol{T}} = \boldsymbol{I} \tag{3.25}$$

Pre-multiplying the model $\boldsymbol{C} = \hat{\boldsymbol{P}}.\boldsymbol{D} + E$ by $\boldsymbol{F}^{-1}$ gives

$$\boldsymbol{C}' = \hat{\boldsymbol{P}}'.\boldsymbol{D} + E' \tag{3.26}$$

where

$$\boldsymbol{C}' = \boldsymbol{F}^{-1}.\boldsymbol{C}, \qquad \hat{\boldsymbol{P}}' = \boldsymbol{F}^{-1}.\hat{\boldsymbol{P}}, \quad \text{and} \quad E' = \boldsymbol{F}^{-1}.E \tag{3.27}$$

Using (3.25) and (3.23), it is easily seen that

$$\mathrm{E}(E'.E'^{\boldsymbol{T}}) = \sigma^2\boldsymbol{I} \tag{3.28}$$

Therefore, equation (3.26) satisfies all the assumptions required for the ordinary non-linear optimization formulation mentioned in the previous section. By substituting transformations (3.27) in equation (3.19), we get

$$\frac{\partial \boldsymbol{Z}(\boldsymbol{D})}{\partial \boldsymbol{D}_{(\nu,o,d)}} = -2\left[\frac{\partial \boldsymbol{D}^{T}}{\partial \boldsymbol{D}_{(\nu,o,d)}}\hat{\boldsymbol{P}}^{T}\boldsymbol{F}^{-1T} + \boldsymbol{D}^{T}\frac{\partial \hat{\boldsymbol{P}}^{T}}{\partial \boldsymbol{D}_{(\nu,o,d)}}\boldsymbol{F}^{-1T}\right]\left[\boldsymbol{F}^{-1}\boldsymbol{C} - \boldsymbol{F}^{-1}\hat{\boldsymbol{P}}.\boldsymbol{D}\ \right] = 0 \qquad \forall(\nu,o,d) \qquad (3.29)$$

and equation (3.20) becomes

$$\boldsymbol{D}^{T}\nabla\hat{\boldsymbol{P}}^{T}_{(\nu,o,d)}\boldsymbol{F}^{-1T}\boldsymbol{F}^{-1}\hat{\boldsymbol{P}}\boldsymbol{D} \ = \left[\boldsymbol{C}^{T}\boldsymbol{F}^{-1T}\boldsymbol{F}^{-1}\nabla\hat{\boldsymbol{P}}^{T}_{(\nu,o,d)} - \hat{\boldsymbol{P}}^{T}_{(\nu,o,d)}\boldsymbol{F}^{-1T}\boldsymbol{F}^{-1}\hat{\boldsymbol{P}}\right]\boldsymbol{D} \ + \hat{\boldsymbol{P}}^{T}_{(\nu,o,d)}\boldsymbol{F}^{-1T}\boldsymbol{F}^{-1}\boldsymbol{C} \qquad \forall(\nu,o,d) \qquad (3.30)$$

From equation (3.23)

$$\boldsymbol{W}^{-1} = \boldsymbol{F}^{-1T}.\boldsymbol{F}^{-1} \qquad (3.31)$$

Therefore, using the following notation:

$$\begin{aligned}
\boldsymbol{A}^{(\nu,o,d)} &= \nabla\hat{\boldsymbol{P}}^{T}_{(\nu,o,d)}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}} && \text{(a } \Gamma.\text{I.J} \times \Gamma.\text{I.J matrix)} \\
\boldsymbol{B}^{(\nu,o,d)} &= \left[\boldsymbol{C}^{T}\boldsymbol{W}^{-1}\nabla\hat{\boldsymbol{P}}_{(\nu,o,d)} - \boldsymbol{P}^{T}_{(\nu,od)}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}\right] && \text{(a 1} \times \Gamma.\text{I.J matrix)} \qquad (3.32) \\
\boldsymbol{g}^{(\nu,o,d)} &= \hat{\boldsymbol{P}}^{T}_{(\nu,o,d)}\boldsymbol{W}^{-1}\boldsymbol{C} && \text{(a scalar value)}
\end{aligned}$$

we can rewrite equation (3.30) as:

$$\boldsymbol{D}^{T}\boldsymbol{A}^{(\nu,o,d)}\boldsymbol{D} \ = \boldsymbol{B}^{(\nu,o,d)}\boldsymbol{D} \ + \boldsymbol{g}^{(\nu,o,d)} \qquad \forall\ (\nu,o,d) \qquad (3.33)$$

If we use sequential numbering for elements of time dependent OD pairs, that is:

$$\boldsymbol{m} = (\tau, \boldsymbol{i}, \boldsymbol{j})$$

and

$$\boldsymbol{n} = (\tau, \boldsymbol{i}, \boldsymbol{j})'$$

then equation (3.19) can be written in scalar form:

$$\sum_{\boldsymbol{m}=1}^{\boldsymbol{\Gamma IJ}} \sum_{\boldsymbol{n}=1}^{\boldsymbol{\Gamma IJ}} \boldsymbol{a}_{\boldsymbol{mn}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} \boldsymbol{d}_{\boldsymbol{m}} \boldsymbol{d}_{\boldsymbol{n}} = \sum_{\boldsymbol{m}=1}^{\boldsymbol{\Gamma IJ}} \boldsymbol{b}_{\boldsymbol{m}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} \boldsymbol{d}_{\boldsymbol{m}} + \boldsymbol{g}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} \qquad \forall(\nu, o, d) \tag{3.34}$$

where $\boldsymbol{a}_{\boldsymbol{mn}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}$ and $\boldsymbol{b}_{\boldsymbol{m}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}$ are the elements of matrices $\boldsymbol{A}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}$ and $\boldsymbol{B}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}$ respectively.

The set of equations in (3.33) or (3.34) can be written in the form of a fixed-point problem, $\boldsymbol{D} = \boldsymbol{f}(\boldsymbol{D})$.

If we ignore the partial derivative of link-flow proportions with respect to demand flows, that is if we assume that $\nabla \hat{\boldsymbol{P}}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}^{\boldsymbol{T}} = 0$, the conventional generalized least-squares estimate is obtained:

$$\boldsymbol{D}^* = \left(\hat{\boldsymbol{P}}^{\boldsymbol{T}} \boldsymbol{W}^{-1} \hat{\boldsymbol{P}}\right)^{-1} \left(\hat{\boldsymbol{P}}^{\boldsymbol{T}} \boldsymbol{W}^{-1} \boldsymbol{C}\right) \tag{3.35}$$

3.7. Iterative Bi-Level Non-Linear Optimization (Bi-NLP)

As mentioned in the previous sections, to find the least-squares estimates of the demand flows, we need to calculate the values of link-flow proportions and their partial derivatives with respect to the demand terms. Unfortunately, in general, and especially in dynamic networks, there is no closed-form equation for the partial derivative terms. Particularly, in a congested network, in both static and dynamic cases, this problem becomes more critical as drivers are more likely to switch to new paths. Path switching might cause some kinks in the value of link-flow proportions that will cause the derivatives to become discontinuous.

To overcome this problem, both link-flow proportions and their derivatives should be estimated numerically by the use of a dynamic simulation program. Since the value of link-flow proportions and their partial derivatives are dependent on demand values, themselves unknown, an iterative procedure is needed to find link-flow proportions and their partial derivatives, given a set of demand flows. Given the link-flow proportions and their partial derivatives we can find a generalized linear or non-linear least-squares estimate of demand flows and can then repeat the process a pre-specified number of times or until a convergence criterion is met.

On the other hand, to find the link-flow proportions we should assign the presumed demand in each iteration to the network. For traffic assignment, we should try to mimic users behavior as closely as possible. Researchers have proposed several assignment rules, of which not any one in particular is followed by the users. Furthermore, in the context of providing real-time information and route guidance to drivers, we might encounter multiple user classes (MUC) in terms of access to the information and adhering to the provided information (Peeta and Mahmassani, 1995a). In the context of this research, notwithstanding lack of generality, we will assume user equilibrium assignment rules, consistently with most of the transportation science literature.

In user equilibrium assignment, or any other method, the users or the supplier agency tries to optimize an objective function. In user equilibrium, each user tries to minimize his own travel time. According to Wardrop's first principle, at equilibrium, no user can reduce his travel time by unilaterally changing his path. In the dynamic context, the mathematical formulation for this optimization is suggested as follows.

$$\textit{Min} \quad G(V \mid D) = \sum_{l} \int_{t} \int_{0}^{v_{(l,t)}} f_{(l,t)}(w,t)\,dw\,dt \tag{3.36}$$

subject to:

$$\begin{aligned} &\sum_{k \in K^{od}} q^{k}_{(\tau,i,j)} = d_{(\tau,i,j)} && \forall i,j \\ &q^{k}_{(\tau,i,j)} \geq 0 && \forall \tau,i,j,k \\ &v_{(l,t)} = \sum_{i \in I} \sum_{j \in J} \sum_{k \in K^{ij}} q^{k}_{(\tau,i,j)} \delta^{(k,l,t)}_{(\tau,i,j)} && \forall l,t \end{aligned} \tag{3.37}$$

or alternatively, with the notation used in the formulation of OD-flow estimation, the above constraints can be rewritten as:

$$\begin{aligned} v_{(l,t)} &= \sum_{\tau \leq t} \sum_{i \in I} \sum_{j \in J} p_{(l,t)(\tau,i,j)} . d_{(\tau,i,j)} \\ p_{(l,t)(\tau,i,j)} &= \frac{\sum_{k \in K^{ij}} q^{k}_{(\tau,i,j)} . \delta^{(k,l,t)}_{(\tau,i,j)}}{d_{(\tau,i,j)}} \end{aligned} \tag{3.38}$$

where

$f_{(l,t)}(.)$ is an (implicit) cost function of travel along link l at time t. If time is discretized the integration over time can be approximated by summation over observation intervals.

K^{ij} is the set of paths between origin i and destination j.

$\boldsymbol{q}^{\boldsymbol{k}}_{(\tau,\boldsymbol{i},\boldsymbol{j})}$ is the demand flow between origin i and destination j, departing at τ which takes the k^{th} path between i and j.

$\delta^{(\boldsymbol{k},\boldsymbol{l},\boldsymbol{t})}_{(\tau,\boldsymbol{i},\boldsymbol{j})}$ is the proportion of demand flow along path k, $\boldsymbol{q}^{\boldsymbol{k}}_{(\tau,\boldsymbol{i},\boldsymbol{j})}$,which traverses on link l in observation interval t. Unlike its static counterpart, path-link incident variable which could only take values of zero or one, this variable has fractional values between zero and one.

Alternatively, Dafermos (1980) has shown that the nonlinear optimization formulation for user equilibrium assignment in the static case can be represented by a variational inequality. If one extends the notation of his formulation to the dynamic case, one may obtain

$$\text{Find } V^* \in X \text{ such that } \boldsymbol{f}_{(l,t)}(\boldsymbol{V}^*)(\boldsymbol{V}-\boldsymbol{V}^*) \geq 0 \qquad \forall\ \boldsymbol{V} \in \boldsymbol{X} \qquad (3.39)$$

where X is the feasible region that satisfies the flow conservation constraints and the non-negativity restrictions expressed in the first and second constraints in the set of equations (3.37).

We will adopt the variational inequality notation to denote the user equilibrium assignment, as it is more compact. Therefore, the bi-level optimization program can be formulated as follows:

$$\text{Min} \quad \boldsymbol{Z}(\boldsymbol{D}) = (\boldsymbol{V} - \hat{\boldsymbol{P}}.\boldsymbol{D}^*)^{\boldsymbol{T}}(\boldsymbol{V} - \hat{\boldsymbol{P}}.\boldsymbol{D}^*) \qquad (3.40)$$

where V(D) solves

$$\boldsymbol{f}_{(l,t)}(\boldsymbol{V}^*)(\boldsymbol{V}-\boldsymbol{V}^*) \geq 0 \qquad \forall\ \boldsymbol{V} \in \boldsymbol{X} \qquad (3.41)$$

As mentioned in the literature review, some authors have tackled the problem when route choice proportions are themselves the output of a congested assignment model (Oh, 1992; Florian and Chen, 1994; Yang *et al*., 1992). Yang (1995), as mentioned in more detail in Section 2.3, has proposed two heuristics to solve this bi-level optimization problem. He claimed that his proposed algorithms are a close representation of the actual decision-making in terms of the Stackelberg game. In the solution of the upper level optimization, Yang ignored that link-flow proportions are dependent on demand flows and has found the traditional generalized least-squares estimate of demand, similar to equation (3.34). In the following section, by referring to a paper by Fisk (1984), we will explore why the Nash, and not the Stackelberg, solution is achieved if the partial derivative terms in the upper level optimization are not included.

3.8. Bi-level Optimization Based on Game Theory

Bard (1988) provides a valuable theoretical and practical reference source for bi-level optimization problems and formulations. Fisk (1984) gives a good discussion of the basic assumptions and application of game theory in transportation systems modeling. In this discussion, we refer extensively to her work and adapt the examples presented in her paper to our case.

Equations (3.40) and (3.41) can be viewed as a game with two players, each one trying to optimize his own objective function, upper and lower level objective functions. We denote the upper level performance function for estimating the optimum demand by $Z_1(\mathbf{D}, \mathbf{P})$, and the lower level for assigning the demand to network by $Z_2(\mathbf{D}, \mathbf{P})$. We have used link-flow proportions as a surrogate variable for link volumes.

In a Nash non-cooperative game, the equilibrium state is characterized by the property that neither player can improve his objective by unilaterally changing his decision. For a given strategy of the other player, *i*'s optimal strategy is found by solving

$$\min_{i} \quad \boldsymbol{Z}_i(\boldsymbol{D}, \boldsymbol{P}) \tag{3.42}$$

where ***D*** and ***P*** are demand and link-flow proportion matrices.

The equilibrium solution is the point at which the optimality conditions for (3.42) are satisfied simultaneously. Suppose we have the following simple equations for Z_1 and Z_2:

$$\boldsymbol{Z}_1 = \boldsymbol{d}_1^2 - \boldsymbol{d}_1\boldsymbol{p}_1 + 2\boldsymbol{p}_1^2 + \boldsymbol{d}_1 \tag{3.43}$$

$$\boldsymbol{Z}_2 = 2\boldsymbol{d}_1^2 - \boldsymbol{d}_1\boldsymbol{p}_1 + \boldsymbol{p}_1^2 \tag{3.44}$$

then

$$\frac{\partial \boldsymbol{Z}_1}{\partial \boldsymbol{d}_1} = 2\boldsymbol{d}_1 - \boldsymbol{p}_1 + 1 = 0 \tag{3.45}$$

$$\frac{\partial \boldsymbol{Z}_2}{\partial \boldsymbol{p}_1} = -\boldsymbol{d}_1 + 2\boldsymbol{p}_1 = 0 \tag{3.46}$$

Note that in the above derivatives we have *ignored* that $\boldsymbol{p}_1$ itself is a function of $\boldsymbol{d}_1$. Solving equations (3.45) and (3.46) *simultaneously* produces

$$\boldsymbol{d}_1^* = -2/3 \qquad \boldsymbol{Z}_1 = -2/9 = -0.222 \tag{3.47}$$

$$\boldsymbol{p}_1^* = -1/3 \qquad \boldsymbol{Z}_2 = 7/9 = 0.777 \tag{3.48}$$

This is the result of the Nash non-cooperative game.

In the Stackelberg game, one player (the leader) knows how the other player (the follower) will respond to any decision he may make. If the player 1 (demand flow optimizer) is the leader and $\boldsymbol{P} = \boldsymbol{h}(\boldsymbol{D})$ is the response of player 2 (traffic assignment optimizer) to decision ***D***, then for any strategy ***D***, the optimal reaction for the follower *h(**D**)* is obtained by solving

$$\min_{\boldsymbol{P}} \quad \boldsymbol{Z}_2(\boldsymbol{D},\boldsymbol{P}) \tag{3.49}$$

The leader's optimal strategy is found by solving

$$\min_{\boldsymbol{D}} \quad \boldsymbol{Z}_1(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D})) \tag{3.50}$$

or equivalently

$$\begin{aligned} &\min_{\boldsymbol{D},\boldsymbol{P}} \quad \boldsymbol{Z}_1(\boldsymbol{D},\boldsymbol{P}) \\ &\boldsymbol{s.t.} \quad \boldsymbol{Z}_2(\boldsymbol{D},\boldsymbol{P}) = \min_{\boldsymbol{P}} \boldsymbol{Z}_2(\boldsymbol{D},\boldsymbol{P}) \end{aligned} \tag{3.51}$$

In the above example, from (3.46) $\boldsymbol{p}_1 = \boldsymbol{h}(\boldsymbol{d}_1) = \boldsymbol{d}_1/2$ so that

$$\boldsymbol{Z}_1(\boldsymbol{d}_1,\boldsymbol{h}(\boldsymbol{p}_1)) = \boldsymbol{d}_1^2 + \boldsymbol{d}_1$$

and

$$\frac{\partial \boldsymbol{Z}_1}{\partial \boldsymbol{d}_1} = 2\boldsymbol{d}_1 + 1 = 0$$

Therefore

$$\boldsymbol{d}_1^* = -1/2, \qquad \boldsymbol{Z}_1 = -1/4 = -0.25 \tag{3.52}$$

$$\boldsymbol{p}_1^* = -1/4, \qquad \boldsymbol{Z}_2 = 7/16 = 0.44 \tag{3.53}$$

Comparing the results of the above with those of (3.47) and (3.48), one can conclude that in the Stackelberg game, where the leader is aware of the response of the follower, the former can play in such a way as to improve the value of his objective function. This improvement is not necessarily compensated by the deterioration of the follower's objective function. In the above example, for instance, the objective function of the follower has improved too.

In mathematical notation, the difference between solving the two theoretical games lies in the manner in which the derivative of the upper level optimization is obtained. In the Nash non-cooperative game, the fact that $\boldsymbol{P}$ is a function of $\boldsymbol{D}$ is ignored, while in Stackelberg game, this relation is taken into account. In other words, in minimizing (3.50) one will have

$$\frac{d\boldsymbol{Z}_1(\boldsymbol{D},\boldsymbol{P})}{d\boldsymbol{D}} = \frac{\partial \boldsymbol{Z}_1}{\partial \boldsymbol{D}} + \frac{\partial \boldsymbol{Z}_1}{\partial \boldsymbol{P}} . \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{D}} \qquad (3.54)$$

Equation (3.54) is equivalent to substituting $h(\boldsymbol{D})$ for $\boldsymbol{P}$ and computing $\frac{\partial \boldsymbol{Z}_1(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}))}{\partial \boldsymbol{D}}$.

Since the relation stated in equation (3.54) is used in the derivation of equation (3.29) the result of the non-linear bi-level optimization to estimate demand is the solution to the Stackelberg game. Whereas, if the conventional least-squares estimate mentioned in (3.18) is used, the solution to the Nash game is achieved. It is worth noting that so far, in all the proposed methods, equations similar to (3.18) are used. Therefore, contrary to previous claims, the solution to the bi-level conventional least-squares optimization is the Nash solution, unless the partial derivative of link-flow proportions with respect to demand is included in the formulations.

Fisk (1984) continues by referring to an iterative multi-period Nash non-cooperative game where each player tries to minimize his performance function without prior knowledge of the other player's function. In this case, in a given period, each player's strategy is based on the other player's strategy in the previous period. Thus in period k the solution $(\boldsymbol{D}^k, \boldsymbol{P}^k)$ is found by solving

$$\begin{aligned} &\min_{\boldsymbol{x}_1} \quad \boldsymbol{Z}_1(\boldsymbol{D}, \boldsymbol{P}^{k-1}) \\ &\min_{\boldsymbol{x}_2} \quad \boldsymbol{Z}_2(\boldsymbol{D}^{k-1}, \boldsymbol{P}) \end{aligned} \tag{3.55}$$

This is equivalent to the iterative bi-level least-squares estimation procedure mentioned before. The principal difference between the Nash non-cooperative solution and the Stackelberg game is that in the former the upper level objective function is optimized only with respect to $\boldsymbol{D}$.

In dynamic transportation systems, finding a closed form equation for $\boldsymbol{P} = \boldsymbol{h}(\boldsymbol{D})$ is not possible and it is suggested that an equivalent Stackelberg solution be sought by substituting $\boldsymbol{P}^{k+1} = \boldsymbol{h}(\boldsymbol{D}^{k+1})$ in the upper level objective function and optimizing it with respect to both variables, i.e. taking the partial derivatives with respect to both variables, as is done in derivation of equation (3.29).

3.9. Constrained Optimization for Single-Horizon Estimation

So far in solving the upper level optimization problem, we did not include any explicit constraints except the lower-level optimization problem. We first introduce non-negativity constraints in a single-horizon estimation. From now on, the objective function is denoted by $Z(\boldsymbol{D}, h(\boldsymbol{D}))$ to indicate that in the optimization of the upper-level, link flows, and subsequently link-flow proportions, are a function of the demand.

In single-horizon estimation, we findtime-dependent demand flows over a long continuous period of time. We include non-negativity constraints in the formulation. We can use archived traffic observations to estimate the demand matrices. The results can be used in short-term planning, such as work zone management, for on-line control applications, or can be archived for long-term planning studies. Definition of the time intervals and the estimation period are the same as depicted in Figure 3.1.

The mathematical formulation of the upper-level optimization is as follows.

$$\begin{aligned} &\boldsymbol{Min} \quad \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D})) = \sum_{\boldsymbol{k}=1}^{\boldsymbol{L.T}} \boldsymbol{e}_{\boldsymbol{k}}^2 = \boldsymbol{e}^{\boldsymbol{T}}\boldsymbol{e} = (\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D}^*)^{\boldsymbol{T}}(\boldsymbol{C} - \hat{\boldsymbol{P}}\boldsymbol{D}^*) \\ &\boldsymbol{S.T.} \quad \boldsymbol{D}_{(\tau,\boldsymbol{i},\boldsymbol{j})} \geq 0 \qquad \forall \tau,\boldsymbol{i},\boldsymbol{j} \end{aligned}$$

The first order conditions for the above problem can be stated as (Sheffi, 1985):

$$\boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})} \frac{\partial \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}))}{\partial \boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})}} = 0 \quad \text{and} \quad \frac{\partial \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}))}{\partial \boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})}} \geq 0 \quad \forall (\nu,\boldsymbol{o},\boldsymbol{d}) \in (\varGamma,\boldsymbol{I},\boldsymbol{J}) \tag{3.56}$$

As in equation (3.29), two conditions can be written as

$$\boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})} \left[\frac{\partial \boldsymbol{D}^{\boldsymbol{T}}}{\partial \boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})}} \hat{\boldsymbol{P}}^{\boldsymbol{T}} + \boldsymbol{D}^{\boldsymbol{T}} \frac{\partial \hat{\boldsymbol{P}}^{\boldsymbol{T}}}{\partial \boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})}} \right] \boldsymbol{W}^{-1} \left[\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D} \right] = 0 \quad \forall \nu,\boldsymbol{o},\boldsymbol{d} \tag{3.57}$$

and

$$\left[\frac{\partial \boldsymbol{D}^{T}}{\partial \boldsymbol{D}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}}\hat{\boldsymbol{P}}^{T}+\boldsymbol{D}^{T}\frac{\partial \hat{\boldsymbol{P}}^{T}}{\partial \boldsymbol{D}_{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}}\right]\boldsymbol{W}^{-1}\left[\boldsymbol{C}-\hat{\boldsymbol{P}}.\boldsymbol{D}\ \right]\geq 0 \quad \forall \boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d} \qquad (3.58)$$

As equations (3.18) and (3.33) show, the solution to

$$\frac{\partial \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}))}{\partial \boldsymbol{D}_{(\nu,\boldsymbol{o},\boldsymbol{d})}}=0 \qquad \forall \nu,\boldsymbol{o},\boldsymbol{d}$$

provides a set of linear or quadratic simultaneous equations in terms of demand flows. One heuristic solution to the set of equations (3.57) and (3.58) is to solve the set of simultaneous equations (3.33), and if $\boldsymbol{d}_{(\nu,o,d)}$ is negative, set it equal to zero.

The following example shows why the above heuristic might result in non-optimal solutions.

Say, we want to solve the following constrained optimization problem

$$\begin{aligned} &min \quad \boldsymbol{Z}(\boldsymbol{D})=\boldsymbol{d}_1^2+2\boldsymbol{d}_2^2+2\boldsymbol{d}_1\boldsymbol{d}_2-\boldsymbol{d}_1-4\boldsymbol{d}_2 \\ &\boldsymbol{S}.\boldsymbol{T}. \quad \boldsymbol{d}_1,\boldsymbol{d}_2\geq 0 \end{aligned}$$

The first order conditions (3.56) can be written as

$$\begin{aligned} &\boldsymbol{d}_1(2\boldsymbol{d}_1+2\boldsymbol{d}_2-1)=0 \quad \text{and} \quad 2\boldsymbol{d}_1+2\boldsymbol{d}_2-1\geq 0 \\ &\boldsymbol{d}_2(2\boldsymbol{d}_1+4\boldsymbol{d}_2-4)=0 \quad \text{and} \quad 2\boldsymbol{d}_1+4\boldsymbol{d}_2-4\geq 0 \end{aligned} \qquad (3.59)$$

The Hessian matrix is

$$\begin{bmatrix} 2 & 1 \\ 2 & 4 \end{bmatrix}$$

which is positive definite, because both its first leading minor, 2, and its second leading minor (i.e. its determinant), 6, are positive. Therefore $\boldsymbol{Z}(\boldsymbol{d}_1, \boldsymbol{d}_2)$ is strictly convex. Therefore, the solution to (3.59) will minimize the objective function.

To solve the above set of equations, we should consider all combinations of possible solutions, i.e.

$$\begin{cases} \boldsymbol{d}_1 = 0 \text{ or } 2\boldsymbol{d}_1 + 2\boldsymbol{d}_2 - 1 = 0 \quad \text{, and} \\ \boldsymbol{d}_2 = 0 \text{ or } 2\boldsymbol{d}_1 + 4\boldsymbol{d}_2 - 4 = 0 \end{cases}$$

We should then check to see if the solutions will meet the non-negativity conditions of the variables and the partial derivatives. That is, to check if the inequality conditions in (3.59) are satisfied. It is clear that finding the optimal solution in this way for a large network can become combinatorial and non-efficient.

Let us now consider the heuristic-solution case where we solve the non-constrained problem, and set the negative results equal to zero, that is

$$\begin{aligned} 2\boldsymbol{d}_1 + 2\boldsymbol{d}_2 - 1 = 0 \\ 2\boldsymbol{d}_1 + 4\boldsymbol{d}_2 - 4 = 0 \end{aligned} \qquad (3.60)$$

We will get

$$\boldsymbol{d}_1 = -1, \quad \boldsymbol{d}_2 = 1.5, \text{ and } Z(\boldsymbol{D}) = -2.5 \qquad (3.61)$$

Since $\boldsymbol{d}_1$ is negative, we set it equal to zero. If we do not re-solve the set of equations (3.59) with $\boldsymbol{d}_1 = 0$, we will have

$$\boldsymbol{d}_1 = 0, \quad \boldsymbol{d}_2 = 1.5, \quad \boldsymbol{Z}(\boldsymbol{D}) = -1.5 \qquad (3.62)$$

Though the above solution satisfies the inequality conditions in equation (3.59), it does not satisfy the condition $\boldsymbol{d}_2(2\boldsymbol{d}_1 + 4\boldsymbol{d}_2 - 4) = 0$. Letting $\boldsymbol{d}_1$=0 in this equation, we find:

$$\boldsymbol{d}_1 = 0, \quad \boldsymbol{d}_2 = 0, \quad \boldsymbol{Z}(\boldsymbol{D}) = 0 \tag{3.63}$$

or

$$\boldsymbol{d}_1 = 0, \quad \boldsymbol{d}_2 = 1, \quad \boldsymbol{Z}(\boldsymbol{D}) = -2 \tag{3.64}$$

The possible solutions to the set of equations in (3.59) are summarized in the following table.

Equation 1	**Equation 2**	$\boldsymbol{d}_1$	$\boldsymbol{d}_2$	***Z(D)***	**Feasibility**
$2\boldsymbol{d}_1+2\boldsymbol{d}_2-1=0$	$2\boldsymbol{d}_1+4\boldsymbol{d}_2-4=0$	-1	1.5	-2.5	N
Setting $\boldsymbol{d}_1$ in the above to zero		0	1.5	-1.5	Y
$\boldsymbol{d}_1=0$	$\boldsymbol{d}_2=0$	0	0	0	Y
$2\boldsymbol{d}_1+2\boldsymbol{d}_2-1=0$	$\boldsymbol{d}_2=0$	0.5	0	-.25	Y
$\boldsymbol{d}_1=0$	$2\boldsymbol{d}_1+4\boldsymbol{d}_2-4=0$	0	1	-2	**OPTIMAL**

As the results in the above table show, to obtain the optimal solution, we cannot simply substitute the resulting negative demand flows with zero, without re-solving the equations.

Therefore, one brute force approach is to set the negative demand flows equal to zero, one by one, and solve the set of generalized least-squares equations again, repeating the process for each negative demand flow. Another approach, especially justified in real-time applications, is to ignore the likely minor decrease in the objective function and not re-solve the set of equations.

One of the advantages of estimating OD-flows using a bi-level iterative method is that by setting the negative demand equal to zero (or any small number), we have to re-solve the lower-level problem (finding the link-flow proportions), and then return to the upper level to find the new set of demand flows based on the resulting link-flow proportions. Therefore, re-solving the problem with the constrained set of variables would be practically accomplished.

3.9.1. Examining the Heuristic Proposed by Bell

Bell (1991) has discussed the addition of non-negativity constraints and has proposed a heuristic algorithm for the problem, particularly for the conventional GLS formulation in static cases. He has also assumed that link-flow proportions are known with certainty and do not change, i.e. "the congestion effects are neglected."

As explained hereafter, implementation of the algorithm to the non-linear optimization case is not feasible. We also discuss why the implementation of the heuristic might not be as efficient as it seems. Thus, assessment of the efficiency of the algorithm will require further investigation.

The problem can be formulated and solved as follows

$$\begin{aligned} &\boldsymbol{Min} \quad \boldsymbol{Z}(\boldsymbol{D}, \boldsymbol{h}(\boldsymbol{D})) = \sum_{\boldsymbol{k}=1}^{\boldsymbol{L}.\boldsymbol{T}} \boldsymbol{e}_{\boldsymbol{k}}^2 = \boldsymbol{e}^{\boldsymbol{T}} \boldsymbol{e} = (\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D}^*)^{\boldsymbol{T}} (\boldsymbol{C} - \hat{\boldsymbol{P}}\boldsymbol{D}^*) \\ &\boldsymbol{S}.\boldsymbol{T}. \quad \boldsymbol{D} \geq \boldsymbol{B} \end{aligned} \tag{3.65}$$

where $\boldsymbol{B}$ is the matrix of lower bounds for demand flows. In many cases the lower bound is zero. The algorithm can be modified to accommodate cases where the constraints impose upper bounds on feasible values.

The problem is solved by forming the Lagrangian equation

$$\boldsymbol{L}(\boldsymbol{D},\Lambda) = \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D})) + \Lambda^{\boldsymbol{T}}(\boldsymbol{B}-\boldsymbol{D}) \quad (3.66)$$

where Λ (Lambda) is a vector of Lagrange multipliers, that is in dynamic case $\lambda_{(\tau,i,j)}$. Since the objective function is convex and the constraints are concave, the necessary and sufficient conditions for a solution are given by

$$\nabla_{\boldsymbol{D}}\boldsymbol{L}(\boldsymbol{D},\Lambda) = 0 \text{ , and the complementary slackness conditions} \quad (3.67)$$

$$\Lambda\nabla_{\Lambda}\boldsymbol{L}(\boldsymbol{D},\Lambda) = 0, \quad \nabla_{\Lambda}\boldsymbol{L}(\boldsymbol{D},\Lambda) \le 0 \quad \text{and} \quad \Lambda \ge 0 \quad (3.68)$$

That is, we must minimize the Lagrangian function (3.66) with respect to $\boldsymbol{D}$, and maximize it with respect to Λ, subject to $\Lambda \geq 0$. This is like finding the local minimum in a saddle shape surface.

Condition (3.67) results in an equation similar to (3.28), but including the Lagrange multipliers:

$$\left[\hat{\boldsymbol{P}}^{\boldsymbol{T}} + \boldsymbol{D}^{*\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}\right]\boldsymbol{W}^{-1}\left[\boldsymbol{C} - \hat{\boldsymbol{P}}.\boldsymbol{D}^{*}\right] - \Lambda^{\boldsymbol{T}} = 0 \quad (3.69)$$

The fixed-point formulation of the above equation, similar to equation (3.31), is:

$$\left[\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}-\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}\right]^{-1}\left[\boldsymbol{D}^{*\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}\boldsymbol{D}^{*}-\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\boldsymbol{C}+\Lambda\right]=\boldsymbol{D}^{*} \quad (3.70)$$

If we ignore the terms denoting the partial derivatives of link-flow proportions with respect to demand, $\nabla\hat{\boldsymbol{P}}^{\boldsymbol{T}}$, or in other words, if we accept the Nash solution to the bi-level optimization, we obtain the following equation

$$\boldsymbol{D}^{*}=\left(\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}\right)^{-1}\left(\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\boldsymbol{C}+\Lambda\right) \quad (3.71)$$

This is similar to the derivation by Bell (1991), except he included a target or historical matrix in the least-squares formulation. We will incorporate the historical demand data using Bayesian inference, as presented in Chapter 4.

Application of the heuristic algorithm introduced by Bell (1991) to the non-linear optimization case is not possible. Therefore, we will continue with the traditional GLS estimate presented in equation (3.71).

Now the problem is to find Λ such that the second set of conditions stated in equations (3.68) is satisfied. For the sake of notational convenience, we define $\boldsymbol{F}$ as follows

$$\boldsymbol{F}=\left(\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}\right) \quad (3.72)$$

Therefore, from equation (3.71)

$$\nabla_{\Lambda}\boldsymbol{D}=\frac{\partial\boldsymbol{D}}{\partial\Lambda}=\boldsymbol{F}^{-1} \quad (3.73)$$

Since $\boldsymbol{F}$, and therefore $\boldsymbol{F}^{-1}$, are positive semi-definite

$$\frac{\partial \boldsymbol{d}_{(\tau,i,j)}}{\partial \lambda_{(\tau,i,j)}} = \boldsymbol{f}_{(\tau,i,j)} \geq 0 \tag{3.74}$$

where $\boldsymbol{f}_{(\tau,i,j)}$ is a term on the principal diagonal of $\boldsymbol{F}^{-1}$.

Also from equation (3.66), at the solution

$$\frac{\partial \boldsymbol{L}}{\partial \lambda_{(\tau,i,j)}} = \boldsymbol{b}_{(\tau,i,j)} - \boldsymbol{d}_{(\tau,i,j)} \leq 0 \tag{3.75}$$

By extending the procedure proposed by Bell (1991), we can use the following algorithm for time-dependent OD estimation using traditional GLS method:

Conventional GLS estimate with non-negativity constraints

Step 1 (Initialization)

set $\Lambda = 0$ (unconstrained estimation)

Step 2 (Iteration)

repeat

for $\tau = 1, \Gamma$; i=1, I and j=1, J

calculate $\boldsymbol{d}_{(\tau,i,j)}$ from (3.71)

if $\boldsymbol{d}_{(\tau,i,j)} < \boldsymbol{b}_{(\tau,i,j)}$ then

set $\lambda_{(\tau,i,j)} = \lambda_{(\tau,i,j)} + (\boldsymbol{b}_{(\tau,i,j)} - \boldsymbol{d}_{(\tau,i,j)})/\boldsymbol{f}_{(\tau,i,j)}$

if $\boldsymbol{d}_{(\tau,i,j)} > \boldsymbol{b}_{(\tau,i,j)}$ then

set $\lambda_{(\tau,i,j)} = \max(0, \lambda_{(\tau,i,j)} + (\boldsymbol{b}_{(\tau,i,j)} - \boldsymbol{d}_{(\tau,i,j)})/\boldsymbol{f}_{(\tau,i,j)})$

until convergence.

End of Algorithm

Bell (1991) further discusses the convergence of the above algorithm (in the static case). Extending his discussion to the dynamic case follows. In accord with the Saddle Point theorem (Sheffi, 1985), we seek to maximize $L^*(\Lambda)$ with respect to Λ, where $\boldsymbol{L}^*(\Lambda)$ is the minimum value of $\boldsymbol{L}$ with respect to $\boldsymbol{D}$ for given Λ. In step 2 of the algorithm, if after solving (3.71) we find that $\boldsymbol{d}_{(\tau,i,j)} < \boldsymbol{b}_{(\tau,i,j)}$, we know from (3.75) that $\boldsymbol{L}^*(\Lambda)$ can be increased by increasing $\lambda_{(\tau,i,j)}$. From (3.74) we see that increasing $\lambda_{(\tau,i,j)}$ also increases $\boldsymbol{d}_{(\tau,i,j)}$, so we should continue increasing $\lambda_{(\tau,i,j)}$ until $\boldsymbol{d}_{(\tau,i,j)} = \boldsymbol{b}_{(\tau,i,j)}$. Hence in accord with (3.74), $\lambda_{(\tau,i,j)}$ should be increased by $(\boldsymbol{b}_{(\tau,i,j)} - \boldsymbol{d}_{(\tau,i,j)})/\boldsymbol{f}_{(\tau,i,j)}$.

Conversely, if after solving (3.71) we find that $\boldsymbol{d}_{(\tau,i,j)} > \boldsymbol{b}_{(\tau,i,j)}$, we know from (3.75) that $\boldsymbol{L}^*(\Lambda)$ can be increased by reducing $\lambda_{(\tau,i,j)}$ until $\boldsymbol{d}_{(\tau,i,j)} = \boldsymbol{b}_{(\tau,i,j)}$ or $\lambda_{(\tau,i,j)} = 0$. This can be achieved by reducing by $(\boldsymbol{b}_{(\tau,i,j)} - \boldsymbol{d}_{(\tau,i,j)})/\boldsymbol{f}_{(\tau,i,j)}$, unless this would result in a number less than 0, in which case $\lambda_{(\tau,i,j)}$ should be set to 0. Bell, further mentions that, hence, each time $\lambda_{(\tau,i,j)}$ is modified in step 2 of the algorithm, $\boldsymbol{L}^*(\boldsymbol{\Lambda})$ increases. When no further modifications are possible, the algorithm terminates.

A key point that is overlooked in the above discussion is the possibility of the existence of covariance terms. That is, the matrix $\boldsymbol{F} = \left(\hat{\boldsymbol{P}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}\right)$ is guaranteed to be diagonal only if $\boldsymbol{W}^{-1} = \boldsymbol{I}$. Therefore in general, we will have

$$\frac{\partial \boldsymbol{d}_{(\nu,o,d)}}{\partial \lambda_{(\tau,i,j)}} \neq 0 \qquad \exists(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d}) \neq (\boldsymbol{\tau},\boldsymbol{i},\boldsymbol{j}) \tag{3.76}$$

It should be noted that because the link-flow proportions, $\hat{\boldsymbol{p}}_{(\tau,i,j)}$ are all positive, the above term will have a positive value, unless some of the terms in $\boldsymbol{W}^{-1}$ are negative.

The inequality (3.76) means that, in the general case, the increase or decrease in $\lambda_{(\tau,i,j)}$, has unknown effects on other demand terms aside from $\boldsymbol{d}_{(\tau,i,j)}$. Therefore, we are not necessarily increasing $\boldsymbol{L}^*(\boldsymbol{\Lambda})$ every time we update the values of $\lambda_{(\tau,i,j)}$. So the convergence of the algorithm, in the general case, both in static and time-dependent OD estimation, is not guaranteed. However, as mentioned earlier, the practical implementation of the method is plausible and its practical convergence needs to be further investigated.

3.10. Rolling Horizon OD-Flow Estimation—Implementation in the Realm of Dynamic Traffic Assignment

We extend the bi-level optimization method presented earlier so that it can be implemented in a rolling horizon framework in conjunction with a dynamic traffic assignment model. In a rolling horizon application, the demand flow is estimated for shorter time intervals, say 15 to 30 minutes, which in this context define an estimation stage. Consecutive estimation stages may either overlap or be disjoint. We suggest that two consecutive stages have some overlap, as usually the estimated demand flow at the end of each stage is less accurate and has high variance. The lower accuracy is because at the end of each stage, the vehicles that have just started their trips may not have reached their destinations. Consequently, the estimated demand toward the end of each stage has high variance. Hence, the size of the estimation stage and the proportion of reliable estimates are determined in accordance with the size of the network and its congestion levels. For instance, consider a case where it takes, say, 15 minutes to travel between the farthest away origin-destination zones under the prevailing traffic conditions. If the estimated stage is 20 minutes, only demand flow during the first five minutes can be efficiently estimated, but if the stage length is 30 minutes, demand flows during the first 15 minutes can be estimated more reliably. It is noteworthy that this issue has not been addressed in any of the previous studies.

The uncertainty in estimating the destination of incomplete trips is aggravated in real-time OD-flow estimation where methods like Kalman filtering may be used. In Kalman filtering, the state of the system (vector of time-dependent demand flows) is updated after one or several observation intervals. The errors due to uncertainty in demand flows in each estimation period (stage) could propagate to the next time intervals, which could increase the error and variance of the estimation significantly.

The advantage of breaking down the estimation period into smaller estimation stages is that we can deal with smaller matrices and make computation faster. The downside to this breaking down is the discontinuities introduced at the beginning or end of the estimation intervals.

We will introduce two different formulations for rolling horizon OD-flow estimation. The difference between the two formulations is in the assumptions made regarding the initial boundary conditions in rolling from one stage to the other. We treat the initial conditions in two ways:

1) The terminating condition at the end of the previous estimation stage will be recognized as the initial condition of the next stage, and
2) The initial condition in each stage is re-estimated along with the demand flow during that estimation stage.

We will call the former fixed-initial-point estimation and the latter free-initial-point estimation.

3.10.1. Fixed Initial-Point Estimation

In this formulation, the demand flow at the end of the previous estimation stage is taken as the starting point in the next estimation stage. Since we are using a dynamic traffic assignment simulation program to estimate the link-flow proportions, the above assumption means that, at the end of each estimation stage, we take a snapshot of the *simulation* results and continue to simulate and assign vehicles onto the paths based on their last estimated destination in the previous estimation stage.

To set up the formulation for rolling horizon estimation, we should modify some of the equations presented earlier. We modify the basic equation (3.9), which was obtained for single-horizon estimation as follows

$$\boldsymbol{c}_{(l,t)} = \sum_{\tau=-\Gamma_P}^{0} \sum_{i=1}^{I} \sum_{j=1}^{J} \hat{\boldsymbol{p}}_{(l,t)(\tau,i,j)} . \boldsymbol{d}^{*}_{(\tau,i,j)} + \sum_{\tau=1}^{\tau_t} \sum_{i=1}^{I} \sum_{j=1}^{J} \hat{\boldsymbol{p}}_{(l,t)(\tau,i,j)} . \boldsymbol{d}_{(\tau,i,j)} + \boldsymbol{\varepsilon}_{(l,t)}$$

l=1,…L ; t=1,…,T_K (3.77)

where

τ_t is the index of the aggregate departure interval which encompasses the observation interval *t*.

Γ_P is the maximum number of aggregate departure intervals (before the start of the current stage) whose demand flows contribute to traffic flows in the network during the current stage. In other words, we assume that vehicles departing before these departure intervals have already exited the network. Γ_P can be estimated by dividing the maximum estimated travel time in the network between any OD pairs by the length of the departure interval. Interval zero denotes the last departure interval before the start of the current stage.

T_K is the number of observation intervals in the current estimation stage K.

Γ_K is the number of aggregate departure intervals in the current estimation stage K.

$\boldsymbol{d}^{*}_{(\tau,i,j)}$ is the optimal estimate of demand flow going from origin *i* to destination *j* that have started their trip in departure interval $\tau \leq 0$, i.e. intervals before the start of the current estimation stage.

The rest of the terms are as defined previously. Figure 3.4 depicts the definition of these time intervals and the schematic transfer of constraints from one estimation stage to another.

The first term on the right hand side (RHS) of equation (3.77), represents the link flows due to the demand flows that originated before the current estimation stage. We denote this term by $\hat{c}^{\tau\leq 0}_{(l,t)}$. We use the '^' sign to show that this partial flow is an estimate, since it is obtained by loading the estimated demand flow $d^{*}_{(\tau,i,j)}$ in the previous estimation stages.

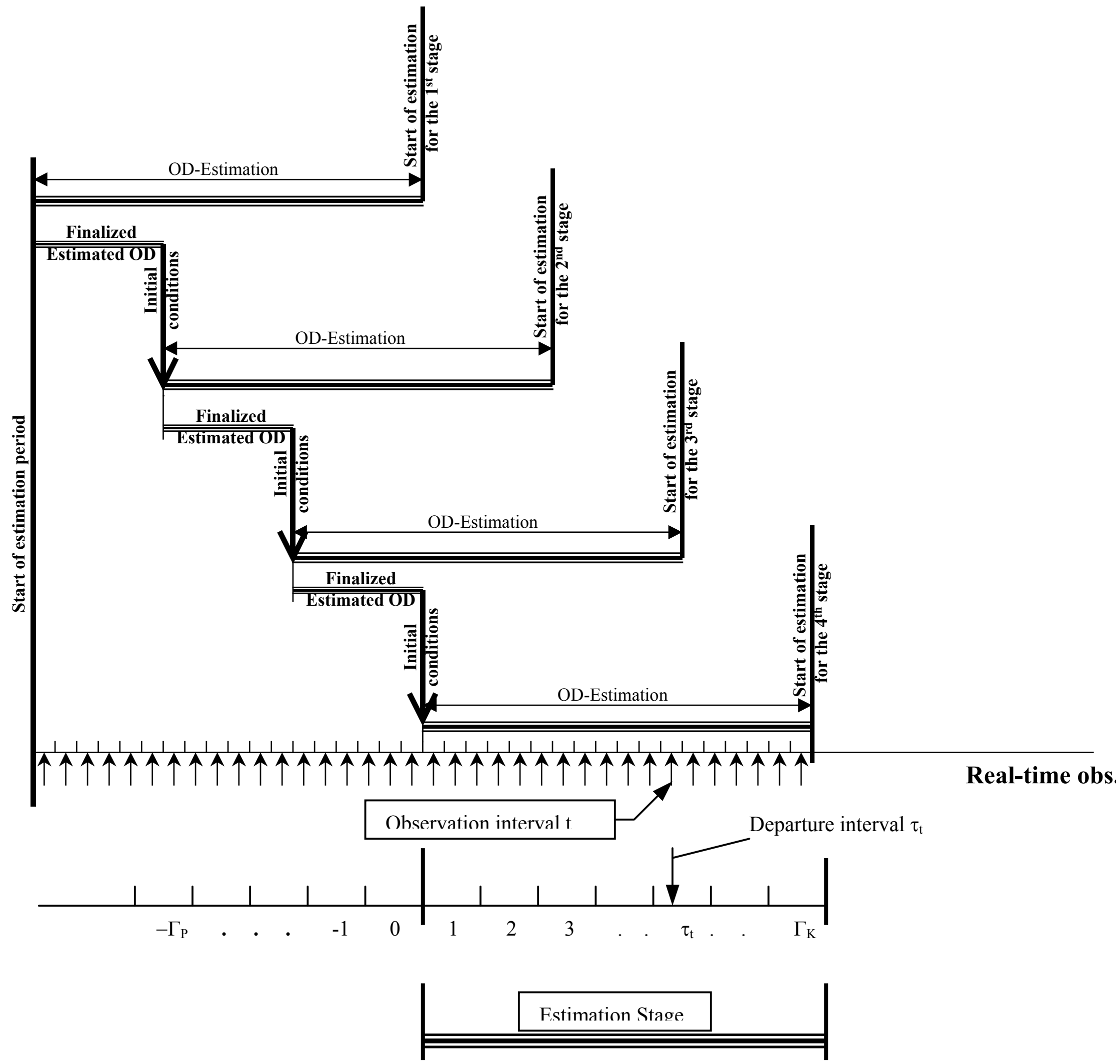


Figure 3.4. Fixed initial-point formulation--time intervals and schematic transfer of initial conditions in the rolling-horizon OD-flow estimation

The demand flow in the current estimation stage should make up for the "remainder" of the observed flow on the links. We denote the remainder of observed flow by $\hat{\boldsymbol{r}}_{(l,t)}$. It is worth noting that though the estimated OD-flow in the previous stage is assumed to be final and fixed, the estimated link flows originated from those OD-flows, i.e. $\hat{\boldsymbol{c}}_{(l,t)}^{\tau\leq 0}$, are not fixed and might change in the iterative procedure of assigning the estimated OD-flow onto the network (lower level optimization). Based on the definition of $\hat{\boldsymbol{r}}_{(l,t)}$ one can write

$$\hat{\boldsymbol{r}}_{(l,t)} = \boldsymbol{c}_{(l,t)} - \hat{\boldsymbol{c}}_{(l,t)}^{\tau\leq 0} \tag{3.78}$$

where $\hat{\boldsymbol{c}}_{(l,t)}^{\tau\leq 0}$ is the estimate of link-flows obtained from the OD-flows that have started before the current estimation stage.

On the other hand, the unknown OD flows in the current estimation stage should make up for the remaining traffic flows on the links. That is

$$\hat{\boldsymbol{r}}_{(l,t)} = \sum_{\tau=1}^{\tau_t} \sum_{i=1}^{I} \sum_{j=1}^{J} \hat{\boldsymbol{p}}_{(l,t)(\tau,i,j)} \boldsymbol{d}_{(\tau,i,j)} + \boldsymbol{e}_{(l,t)} \qquad l=1,\ldots L;\ t=1,\ldots,T_K \tag{3.79}$$

where $\boldsymbol{e}_{(l,t)}$ are the observed residuals.

Therefore, equation (3.12) can be re-written as

$$\boldsymbol{Min} \quad \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D})) = \sum_{n=1}^{L.T_K} \boldsymbol{e}_n^2 = \boldsymbol{E}^T\boldsymbol{E} = (\hat{\boldsymbol{R}}_K - \hat{\boldsymbol{P}}_K\boldsymbol{D}_K)^T(\hat{\boldsymbol{R}}_K - \hat{\boldsymbol{P}}_K\boldsymbol{D}_K) \tag{3.80}$$

where the subscript K denotes that the matrices and their elements correspond to the current estimation stage, and $\hat{\boldsymbol{R}}_K$ is the vector of elements $\hat{\boldsymbol{r}}_{(l,t)}$.

The non-linear optimization solution to equation (3.80) is

$$\left[\boldsymbol{R}_{\boldsymbol{K}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\nabla\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}-\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}_{\boldsymbol{K}}\right]^{-1}\left[\boldsymbol{D}_{\boldsymbol{K}}^{*\boldsymbol{T}}\nabla\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}_{\boldsymbol{K}}\,\boldsymbol{D}_{\boldsymbol{K}}^{*}-\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\boldsymbol{R}_{\boldsymbol{K}}^{\boldsymbol{T}}\right]=\boldsymbol{D}_{\boldsymbol{K}}^{*} \qquad (3.81)$$

As before, if we ignore the partial derivatives of the link-flow proportions with respect to demand flows, that is if we assume that $\nabla\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}=0$, the conventional generalized least-squares estimate will be obtained:

$$\boldsymbol{D}_{\boldsymbol{K}}^{*}=\left(\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{P}}_{\boldsymbol{K}}\right)^{-1}\left(\hat{\boldsymbol{P}}_{\boldsymbol{K}}^{\boldsymbol{T}}\boldsymbol{W}^{-1}\hat{\boldsymbol{R}}_{\boldsymbol{K}}\right) \qquad (3.82)$$

As explained in detail in Sections (3.4) and (3.7), equations (3.81) and (3.82) are the solution to the upper-level optimization, and in each iteration the values of $\hat{\boldsymbol{R}}_{\boldsymbol{K}}$, $\hat{\boldsymbol{P}}_{\boldsymbol{K}}$ and $\nabla\hat{\boldsymbol{P}}_{\boldsymbol{K}}$ should be re-estimated by solving the lower-level problem.

In the case of constrained least-squares optimization, we should solve the following mathematical programming formulation

$$\begin{aligned}&\boldsymbol{Min}\quad \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}))=(\hat{\boldsymbol{R}}_{\boldsymbol{K}}-\hat{\boldsymbol{P}}_{\boldsymbol{K}}\boldsymbol{D}_{\boldsymbol{K}})^{\boldsymbol{T}}(\hat{\boldsymbol{R}}_{\boldsymbol{K}}-\hat{\boldsymbol{P}}_{\boldsymbol{K}}\boldsymbol{D}_{\boldsymbol{K}})\\&\boldsymbol{S.T.}\quad \boldsymbol{d}_{(\tau,\boldsymbol{i},\boldsymbol{j})}\geq 0 \qquad\qquad \forall(\tau,\boldsymbol{i},\boldsymbol{j})\end{aligned} \qquad (3.83)$$

The discussion presented in the single-horizon constrained estimation case (Section 3.9) applies here too, with the exception that the "remainder" traffic volume in the current stage, $\hat{\boldsymbol{R}}_{\boldsymbol{K}}$, should be substituted for the traffic counts, $\boldsymbol{C}$.

In summary, we should solve equations (3.81) or (3.82) and set the obtained negative demand flows equal to zero and re-solve the problem. As a heuristic, we can ignore the negative estimated values, if any, in the intermediate iterations or in the final results.

3.10.2. Free Initial-Point Estimation

In a rolling horizon implementation of dynamic traffic assignment models, at the beginning of each estimation stage, the state of the system in the real world might be different from the estimated state obtained from the simulator (Peeta and Mahmassani, 1995a). To address this problem, we introduce a new set of constraints to the OD-flow optimization problem. We assume that the information on the number of vehicles on all links is available. If some links do not have detectors, we can use the simulator's results from the last estimation/prediction stage as the estimated flows on those links.

In the free initial-point estimation approach, we take a snapshot of the vehicles in the *real network* and pass it to the estimation process. Figure 3.5 illustrates the definition of time intervals and schematic transfer of initial-state constraints to the estimation process from the real-world system observations.

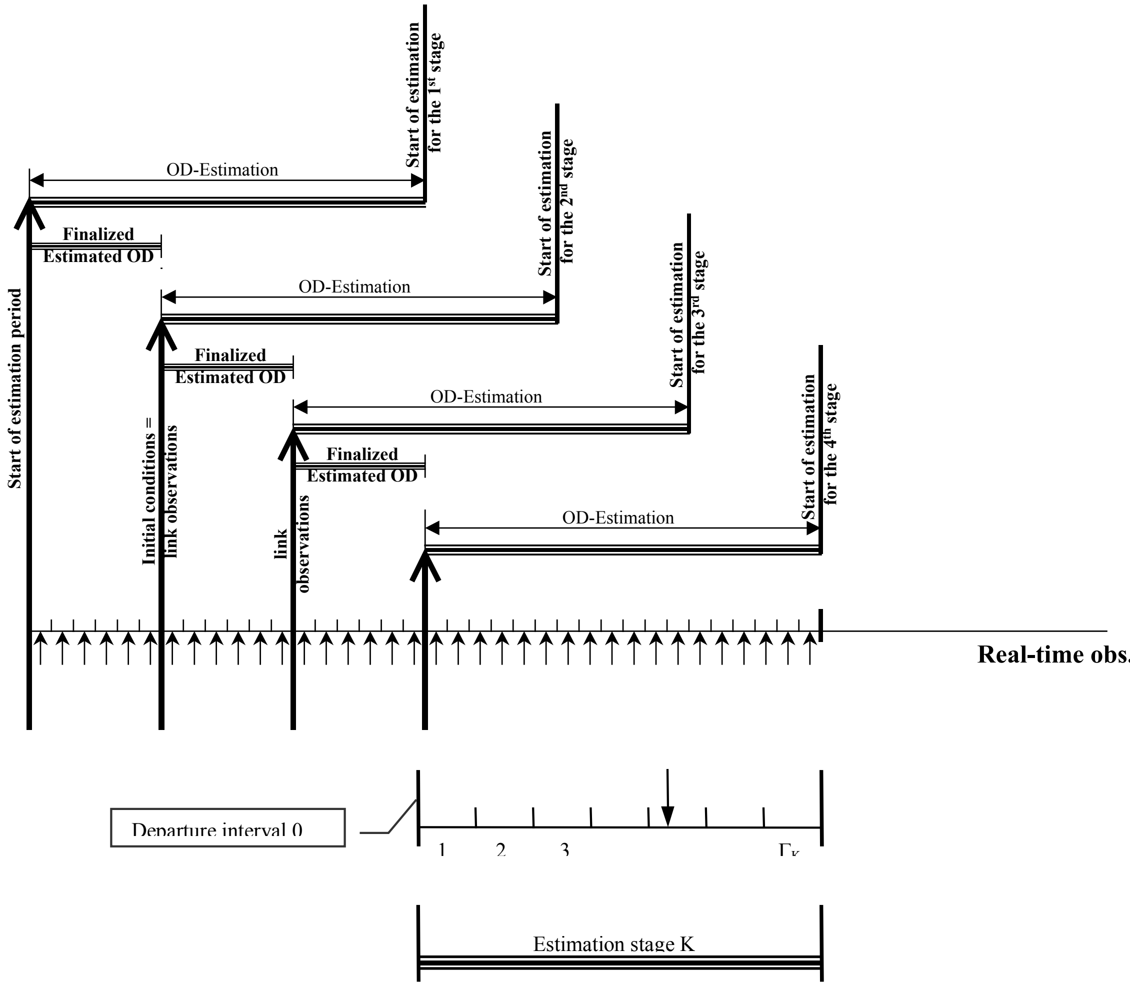


Figure 3.5. Free initial-point estimation--time intervals and schematic transfer of initial conditions in the rolling-horizon OD-flow estimation

At the beginning of each estimation stage, the number of vehicles residing on the links located in each origin zone is added up. The resulting number of vehicles is considered as the originating demand value for the zone at the beginning of the estimation stage. We assume a virtual departure interval numbered zero as the departure interval of these vehicles. The destination zones of these vehicles are unknown which should be estimated along with the OD-flows at the current stage. The mathematical formulation for this problem is as follows:

$$\begin{aligned}
&\boldsymbol{Min} \quad \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D})) = (\boldsymbol{C}_{\boldsymbol{K}_0} - \hat{\boldsymbol{P}}_{\boldsymbol{K}_0}\boldsymbol{D}_{\boldsymbol{K}_0})^{\boldsymbol{T}}(\boldsymbol{C}_{\boldsymbol{K}_0} - \hat{\boldsymbol{P}}_{\boldsymbol{K}_0}\boldsymbol{D}_{\boldsymbol{K}_0}) \\
&\boldsymbol{S.T.} \quad \sum_{\boldsymbol{l}=1}^{\boldsymbol{L}_{\boldsymbol{i}}} \boldsymbol{c}_{(\boldsymbol{l},0)} = \sum_{\boldsymbol{j}=1}^{\boldsymbol{J}} \boldsymbol{d}_{(0,\boldsymbol{i},\boldsymbol{j})} \qquad \boldsymbol{i} = 1,\ldots,\mathrm{I} \\
&\qquad\quad \boldsymbol{d}_{(\tau,\boldsymbol{i},\boldsymbol{j})} \geq 0 \qquad\qquad \forall(\tau,\boldsymbol{i},\boldsymbol{j})
\end{aligned} \tag{3.84}$$

where:

$\boldsymbol{c}_{(l,0)}$ is the vehicle count on link l at the beginning of the estimation stage

$\boldsymbol{d}_{(0,i,j)}$ is the unknown demand flow between origin i and destination j at the beginning of the estimation stage.

L_i is the number of links that reside in origin zone i.

$\Gamma_{\boldsymbol{K}_0}$ is the number of aggregate departure intervals in the current estimation stage plus the initial virtual departure interval zero.

K_0 denotes the augmented current estimation stage with the initial virtual departure interval zero added.

The first constraint in the formulation (3.84) represents the initial condition of the estimation stage based on the state of the system in the real world.

The Lagrange multipliers, $\boldsymbol{u_i}$, should be introduced to solve the above constrained optimization problem, that is

$$\begin{aligned} &\boldsymbol{Min} \quad && \boldsymbol{L}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}),\boldsymbol{U}) = \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D})) + \sum_{\boldsymbol{i}=1}^{\boldsymbol{I}} \boldsymbol{u_i}\left[\sum_{\boldsymbol{i}=1}^{\boldsymbol{L_i}} \boldsymbol{c}_{(\boldsymbol{l},0)} - \sum_{\boldsymbol{j}=1}^{\boldsymbol{J}} \boldsymbol{d}_{(0,\boldsymbol{i},\boldsymbol{j})}\right] \\ &\boldsymbol{S.T.} && \boldsymbol{d}_{(\tau,\boldsymbol{i},\boldsymbol{j})} \geq 0 \qquad \forall(\tau,\boldsymbol{i},\boldsymbol{j}) \qquad (3.85) \\ & && \boldsymbol{u_i} \geq 0 \qquad \forall \boldsymbol{i} \end{aligned}$$

where $\boldsymbol{U}$ is the vector of Lagrange multipliers $\boldsymbol{u_i}$.

To find the optimal solution to the above formulation, we should find $\boldsymbol{D}^*$ such that the following set of conditions are met (Sheffi, 1985):

$$\begin{aligned} \boldsymbol{d}_{(\nu,\boldsymbol{o},\boldsymbol{d})} \nabla_{\boldsymbol{D}} \boldsymbol{L}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}),\boldsymbol{U}) &= 0 \qquad \forall(\nu,\boldsymbol{o},\boldsymbol{d}) \\ \nabla_{\boldsymbol{D}} \boldsymbol{L}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}),\boldsymbol{U}) &\geq 0 \qquad (3.86) \\ \nabla_{\boldsymbol{u}} \boldsymbol{L}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}),\boldsymbol{U}) &= 0 \end{aligned}$$

First order conditions can be written as:

$$\boldsymbol{d}_{(\boldsymbol{v},\boldsymbol{o},\boldsymbol{d})}\left\{\frac{\partial \boldsymbol{Z}(\boldsymbol{D},\boldsymbol{h}(\boldsymbol{D}))}{\partial \boldsymbol{d}_{(\boldsymbol{v},\boldsymbol{o},\boldsymbol{d})}} - \delta_\tau \boldsymbol{u_i}\right\} = 0 \qquad \forall(\boldsymbol{v},\boldsymbol{o},\boldsymbol{d}) \quad (3.87)$$

where $\begin{cases} \delta_\tau = 1 & \boldsymbol{if}\ \tau = 0 \\ \delta_\tau = 0 & \boldsymbol{o.w.} \end{cases}$

Equation (3.87) transforms to

$$\boldsymbol{d}_{(\nu,o,d)}\left\{2\left[\frac{\partial \boldsymbol{D}_{K_0}^{T}}{\partial \boldsymbol{d}_{(\nu,o,d)}}\hat{\boldsymbol{P}}_{K_0}^{T}+\boldsymbol{D}_{K_0}^{T}\frac{\partial \hat{\boldsymbol{P}}_{K_0}^{T\ T}}{\partial \boldsymbol{d}_{(\nu,o,d)}}\right]\left[\hat{\boldsymbol{P}}_{K_0}\boldsymbol{D}_{K_0}-\boldsymbol{C}_{K_0}\right]-\delta_{\tau}\boldsymbol{u}_{i}\right\}=0$$

$$\forall(\nu,o,d) \qquad (3.88)$$

$$\boldsymbol{d}_{(\nu,o,d)}\left\{2\left[\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}+\boldsymbol{D}_{K_0}^{T}\nabla\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}\right]\left[\hat{\boldsymbol{P}}_{K_0}\boldsymbol{D}_{K_0}-\boldsymbol{C}_{K_0}\right]-\delta_{\tau}\boldsymbol{u}_{i}\right\}=0$$

$$\forall(\nu,o,d) \qquad (3.89)$$

Condition (3.89) can be satisfied in two ways:

1) $\boldsymbol{d}_{(\nu,o,d)}=0$, (3.90a)

or

2) $\boldsymbol{D}_{K_0}^{T}\nabla\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}\hat{\boldsymbol{P}}_{K_0}\boldsymbol{D}_{K_0}=\left[\boldsymbol{C}_{K_0}^{T}\nabla\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}-\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}\hat{\boldsymbol{P}}_{K_0}\right]\boldsymbol{D}_{K_0}+\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}\boldsymbol{C}_{K_0}+\frac{1}{2}\delta_{\tau}\boldsymbol{u}_{i}$

$$\forall(\nu,o,d) \qquad (3.90b)$$

The second condition can be rewritten as:

$$\left\{\left[\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}+\boldsymbol{D}_{K}^{T}\nabla\hat{\boldsymbol{P}}_{(\nu,o,d)}^{T}\right]\left[\hat{\boldsymbol{P}}_{K}\boldsymbol{D}_{K}-\boldsymbol{C}_{K}\right]-\frac{1}{2}\delta_{\tau}\boldsymbol{u}_{i}\right\}\geq 0 \qquad \forall(\nu,o,d) \qquad (3.91)$$

The non-negativity condition of (3.91) should be checked if condition (3.90a) is binding, that is when $\boldsymbol{d}_{(\nu,o,d)}=0$.

The third condition transforms to:

$$\sum_{l=1}^{L_i} c_{(l,t)} - \sum_{j=1}^{J} d_{(0,i,j)} = 0 \qquad i = 1,\ldots,\mathrm{I} \qquad (3.92)$$

We should also include the non-negativity constraints, that is

$$d_{(\tau,i,j)} \geq 0 \qquad \forall(\tau, i, j) \qquad (3.93)$$

The two sets of equations (3.90) and (3.92) provide 'Γ.I.J+I' equations to solve for 'Γ.I.J' unknown demands, $d_{(\nu,o,d)}$, and 'I' unknown Lagrange multipliers, u_i.

In summary, as a heuristic approach, one can solve the following sets of simultaneous equations:

$$\begin{cases} D_{K_0}^T \nabla \hat{P}_{(\nu,o,d)}^T \hat{P}_{K_0} D_{K_0} = \left[C_{K_0}^T \nabla \hat{P}_{(\nu,o,d)}^T - \hat{P}_{(\nu,o,d)}^T \hat{P}_{K_0} \right] D_{K_0} + \hat{P}_{(\nu,o,d)}^T C_{K_0} + \tfrac{1}{2} \delta_\tau u_i \\ \qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \forall(\nu, o, d) \\ \sum_{l=1}^{L_i} c_{(l,t)} - \sum_{j=1}^{J} d_{(0,i,j)} = 0 \qquad\qquad \forall i \end{cases} \qquad (3.94)$$

If $d_{(\nu,o,d)}$ was negative, set it to zero, i.e. $d_{(\nu,o,d)} = 0$ and solve the lower-level problem with the estimated values of OD-flows.

3.10.3. Analogy of Single-Horizon and Rolling-Horizon Estimation with Linear Regression

To illustrate the differences between the estimation frameworks, they are compared to piecewise linear regression models. In the single-horizon estimation, we concurrently include all observations over the estimation period and estimate the parameters of the model (optimal OD-flows) such that the dependent variables (link flows) would best fit the observed link-flow data. In a simple linear regression model, the parameters of the model, the slope and the intercept of the line, are estimated such that the corresponding line would best fit the whole dataset. Figure 3.6 depicts this concept when a single regression line is fitted over the whole set of an arbitrary time series dataset.

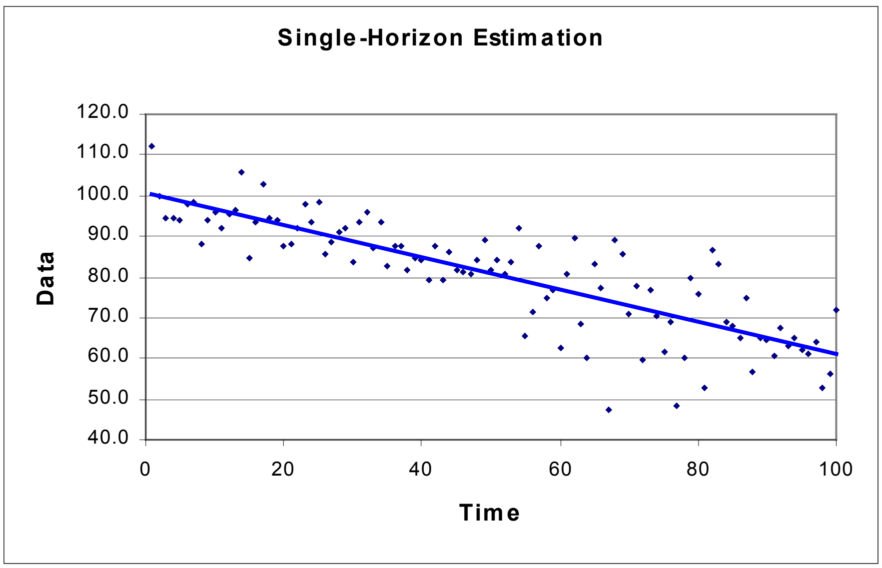


Figure 3.6. Analogy of single-horizon OD-flow estimation to simple linear regression

In the rolling-horizon estimation, we divide the estimation period into stages and fit separate models to each estimation stage. In the fixed initial-point implementation of rolling- horizon estimation, the terminal condition of one stage is the initial condition of the next. Figure 3.7 shows a similar concept in fitting a "piecewise" regression line to an arbitrary set of time-series data points. For instance, a line is fit to the data from time zero to 60. The estimation stage, then, is rolled by, say, 20 intervals and a new line is fit to the data from interval 20 to 80.

In the fixed initial-point estimation, the initial condition of the estimation stage is fixed at the value estimated in the preceding stage. In the example of piecewise regression line, this is equivalent to fixing the starting point of the line at interval 20 to its estimated value in the previous estimation stage and estimate the slope of the regression line which is fitted to the data points from interval 20 to 80.

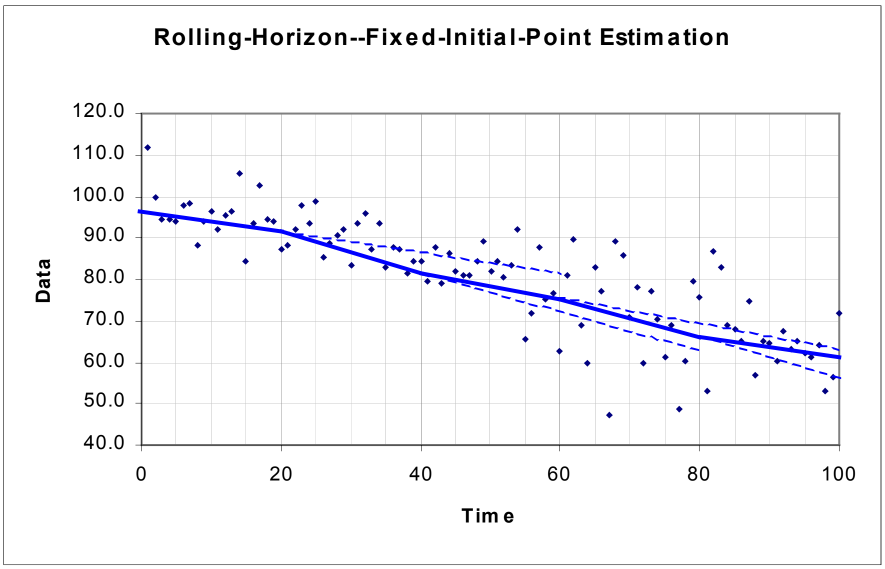


Figure 3.7. Analogy of fixed initial-point rolling-horizon OD-flow estimation to piecewise linear regression

In the free initial-point formulation of rolling-horizon OD-flow estimation, we re-estimate the initial conditions in each estimation stage. Therefore, there are more degrees of freedom in the estimation process. This formulation is similar to a piecewise linear regression where both the slope and the intercept are estimated for each piece independent of the previous estimates. In our example (Figure 3.8), it means that a line is fitted to the data from interval zero to, say, 60. After rolling the estimation stage for, say, 20 intervals, a new line is fitted to the observed data from interval 20 to 80 and the process is repeated every time the estimation stage is rolled ahead.

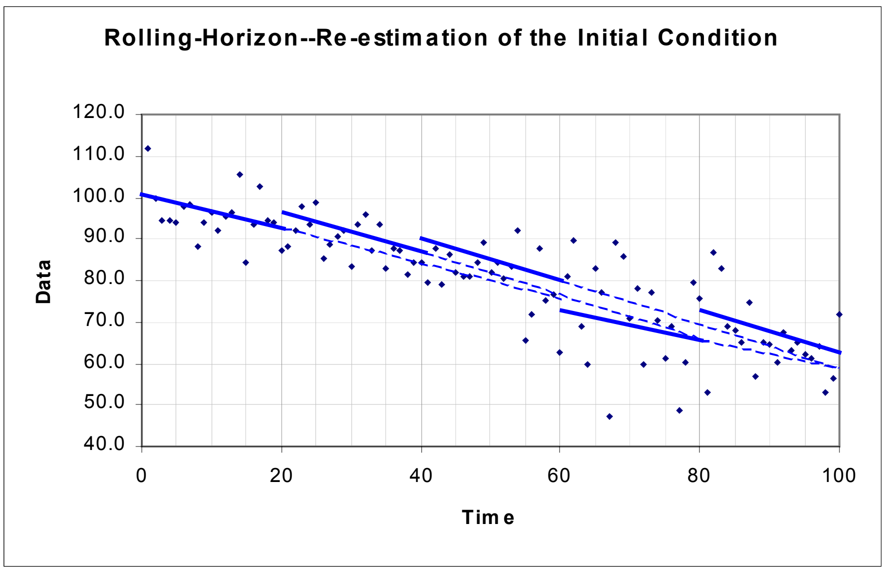


Figure 3.8. Analogy of free initial-point rolling-horizon OD-flow estimation to piecewise linear regression

3.11. Summary

In this chapter, detailed formulations of the time-dependent OD-flow estimation problem from the link-flow measurements were presented. First, the problem was formulated for the unconstrained case, which constitutes the upper-level problem. The ordinary least-squares and generalized least-squares estimation methods were presented. The bi-level optimization solution to the problem was then discussed. The problem was also formulated as a non-linear optimization and its ordinary and generalized solutions were explored. The Bi-level optimization was also explained in the context of a theoretical game and the conditions to obtain the Stackelberg or Nash solutions were discussed. The solution to the constrained optimization formulation was presented. In this context, the problem was formulated in a rolling horizon framework with specific application to the real-time implementation of Dynamic Traffic Assignment models.

The next chapter examines how available prior information on the OD flows may be incorporated in the estimation process. Application of Bayesian inference models for this purpose is suggested and discussed in more detail.

CHAPTER 4. BAYESIAN INFERENCE

4.1. Introduction

Bayesian inference provides a statistical method to update prior beliefs on OD flows by the evidence obtained from the time-varying traffic flows. Prior information on OD demand flows could be obtained from other methods, such as direct surveys or modeling techniques, or it could be the result of recent estimation using traffic counts for similar past time intervals.

As mentioned before, several previous methods have included a target OD-flow matrix in the formulation of the least-squares objective function. This approach is adopted particularly in static formulations, where the number of unknowns is far greater than the number of equations obtained from link flow observations. As discussed before, in the dynamic case, the availability of more information on link flows usually results in an over-specified problem or one that could be converted to an over-specified problem by increasing the length of the departure intervals. It has also been shown that the solution to the formulation including the target matrix is a special case of the Bayesian approach in the static case (Maher, 1983).

In this research, we adopt the Bayesian inference method because it can be used to fuse the prior OD-flow information with the (archived or real-time) sample link-flow observations, independent of the utilized demand estimation method. This gives us the flexibility to use any method, such as generalized least-squares or non-linear optimization to estimate the OD-flows (or in Bayesian terminology, to infer the likelihood information) from the sample of link-flow observations. We can then combine the estimated demand flows (likelihood function) with the historical information (*a priori* distribution) using the Bayesian inference method. In this research, we realize the formulation where demand flows are assumed to follow a multivariate normal distribution and are estimated by the generalized least-squares method. However, the method could be generalized to the cases where OD-flows are estimated by non-linear optimization or where other distributions are assumed for OD-flows.

4.2. Problem Statement

Consider the following general quasi-linear model (the notation used in this chapter is the same as introduced at the beginning of Chapter 3 unless explicitly noted otherwise):

$$\boldsymbol{C} = \hat{\boldsymbol{P}}.\boldsymbol{D} + E \qquad (4.1)$$

where, as mentioned before,

D is the vector of demand values of size Γ.I.J×1,

C is the vector of observations, L.T×1

P is a L.T×Γ.I.J matrix of link-flow proportions which is assumed to be known (or is estimated) for any given demand, and

E is a L.T×1 vector of error terms.

The residuals are assumed to be normal i.i.d (identically and independently normally distributed), that is $\boldsymbol{\varepsilon} \sim \boldsymbol{N}(0, \gamma^{-1}\boldsymbol{I}_{L.T})$. The precision matrix of $\boldsymbol{\varepsilon}$ is $\gamma \boldsymbol{I}_{L.T}$, and $\sigma^2 \boldsymbol{I}_{L.T}$ is variance-covariance matrix of $\boldsymbol{\varepsilon}$, that is $\sigma^2 = \gamma^{-1} > 0$ is an unknown scalar. $\boldsymbol{I}_{L.T}$ is a L.T×L.T identity matrix.

The objective is to provide inferences for $\boldsymbol{D}$ and γ when observing:

$\boldsymbol{C} = \{\boldsymbol{c}_{(l,t)} | \; l=1\ldots L; \; t=1\ldots T\}$ (observation on link *l* at observation interval *t*).

The word inference implies a procedure that extracts information about ***D*** from the sample ***C***.

4.3. Bayes' Theorem

Suppose one's prior information about ***D*** is represented by a probability density function $\xi(\boldsymbol{D},\gamma)$, $\boldsymbol{D} \in \Re^{(\Gamma,I,J)}$, $\gamma > 0$ (gamma), then Bayes' theorem combines this information with the information contained in the sample of observations. The likelihood function $\boldsymbol{L}(\boldsymbol{D},\gamma)$ for ***D*** and γ is given by (Broemeling, 1985)

$$\boldsymbol{L}(\boldsymbol{D},\gamma) \propto \gamma^{L.T/2} \boldsymbol{e}^{-\frac{\gamma}{2}(\boldsymbol{C}-\hat{\boldsymbol{P}}\boldsymbol{D})^T(\boldsymbol{C}-\hat{\boldsymbol{P}}\boldsymbol{D})} \tag{4.2}$$

where $\propto$ means proportional to. The likelihood function is our sample information about the parameters and is the conditional density function of the sample random variables given ***D*** and γ.

Bayes' theorem gives the conditional density of ***D*** and $\boldsymbol{\gamma}$ given ***C***

$$\xi^*(\boldsymbol{D},\gamma \,|\, \boldsymbol{C}) \propto \boldsymbol{L}(\boldsymbol{D},\gamma \,|\, \boldsymbol{C})\xi(\boldsymbol{D},\gamma) \tag{4.3}$$

The posterior density of **D** is $\xi^*(\boldsymbol{D},\gamma \,|\, \boldsymbol{C})$ (pronounced xi) which represents our knowledge of ***D*** and γ after observing the sample ***C***. On the other hand, our information about ***D*** and γ before ***C*** is observed is contained in the prior density.

Note that the posterior density (4.3) is written with a proportionality symbol and ξ is used to denote the prior and ξ^* the posterior densities. If in equation (4.3) we use an equality sign, the posterior density is

$$\xi^*(\boldsymbol{D},\gamma \mid \boldsymbol{C}) = \mathrm{K}\, \boldsymbol{L}(\boldsymbol{D},\gamma \mid \boldsymbol{C})\xi(\boldsymbol{D},\gamma)$$

where K is the normalizing constant and is given by

$$\mathrm{K}^{-1} = \int_0^\infty \int_{\boldsymbol{R}^{(\Gamma,\mathrm{I},\mathrm{J})}} \boldsymbol{L}(\boldsymbol{D},\gamma \mid \boldsymbol{C})\xi(\boldsymbol{D},\gamma)\, \boldsymbol{dD}\, \boldsymbol{d}\gamma$$

which is the marginal probability density of ξ. (In the case of discrete demand flow values, the integration will be substituted with summation sign.)

4.4. Prior Information

4.4.1. <u>Normal-Gamma Density</u>

If $\xi(\boldsymbol{D},\gamma)$ follows a normal-gamma prior density function then

$$\xi(\boldsymbol{D},\gamma) = \xi_1(\boldsymbol{D} \mid \gamma)\xi_2(\gamma) \tag{4.4}$$

where

$$\xi_1(\boldsymbol{D} \mid \gamma) \propto \gamma^{\Gamma.\mathrm{I}.\mathrm{J}} \boldsymbol{e}^{-\frac{\gamma}{2}(\boldsymbol{D}-\boldsymbol{\mu})^{\boldsymbol{T}} \boldsymbol{\Psi} (\boldsymbol{D}-\boldsymbol{\mu})} \tag{4.5}$$

In the OD-flow estimation problem under consideration, $\boldsymbol{\mu}$ (mu) is a Γ.I.J×1 given mean vector of OD flows and $\boldsymbol{\Psi}$ (psi) is a known Γ.I.J×Γ.I.J positive definite matrix representing the variance-covariance matrix of the *OD flows* (The matrix ***W***, introduced in Chapter 3, is the variance-covariance matrix of the link flow observations). Thus, ξ_1 is the conditional density of ***D*** given γ and is normal with mean vector $\boldsymbol{\mu}$ and precision matrix $\gamma\boldsymbol{\Psi}$. (Dispersion matrix is the inverse of variance-covariance matrix).

The marginal prior density of γ is gamma with parameters α>0 and β>0.

$$\xi_2(\gamma) \propto \gamma^{\alpha-1} \boldsymbol{e}^{-\gamma\beta} \quad , \gamma > 0 \tag{4.6}$$

Since (4.4) is the prior density of ***D*** and γ, the marginal density of ***D*** is found by integrating (4.5) with respect to γ. Substituting equations (4.6) and (4.5) in (4.4) and integrating, the marginal density of ***D*** can be obtained as (Broemeling, 1985)

$$\xi_1(\boldsymbol{D}) \propto \int_0^\infty \xi(\boldsymbol{D},\gamma)\boldsymbol{d}\gamma \propto \left[2\beta + (\boldsymbol{D}-\boldsymbol{\mu})^{\boldsymbol{T}} \boldsymbol{\Psi} (\boldsymbol{D}-\boldsymbol{\mu})\right]^{-(\Gamma.\text{I}.\text{J}+2\alpha)/2} \tag{4.7}$$

which is a t density function with 2α degrees of freedom, location vector $\boldsymbol{\mu}$ and precision matrix $(2\alpha)(2\beta)^{-1}\boldsymbol{\Psi}$.

By using the normal-gamma density function as a prior for the parameters, we cannot stipulate the prior information about ***D*** separately from that of γ. The parameters of the marginal distribution of ***D*** involve α and β, which are parameters of the prior distribution of γ, but the marginal prior density of γ does not involve parameters of the marginal density of ***D***.

The parameter vector $\boldsymbol{\mu}$ is our prior mean for $\boldsymbol{D}$, while our opinion of the correlation between the components of $\boldsymbol{D}$ is given by $\Psi^{-1}(2\beta)(2\alpha-2)^{-1}$, which is the marginal prior dispersion matrix of $\boldsymbol{D}$. Since this involves α and β, the marginal prior information about γ depends on the choice of the dispersion or precision matrix of $\boldsymbol{D}$.

With regard to the information about γ, it is convenient to think of γ as the inverse of the residual variance σ^2, that is

$$\boldsymbol{E}(\gamma^{-1}) = \beta(\alpha-1)^{-1}, \qquad \alpha > 1 \tag{4.8}$$

$$\text{var}(\gamma^{-1}) = \beta^2(\alpha-1)^{-2}(\alpha-2)^{-1}, \qquad \alpha > 2 \tag{4.9}$$

or

$$\text{E}(\gamma) = \alpha/\beta \quad \text{and} \quad \text{var}(\gamma) = \alpha/\beta^2 .$$

These two equations together with

$$\text{E}(\boldsymbol{D}) = \boldsymbol{\mu} \tag{4.10}$$

and

$$\text{S}(\boldsymbol{D}) = \Psi^{-1}(2\beta)(\text{L.T} + 2\alpha - 2)^{-1} \tag{4.11}$$

which are the prior mean vector and prior dispersion matrix of $\boldsymbol{D}$, will assist us in choosing the four hyperparameters, α, β, $\boldsymbol{\mu}$ and Ψ, for the prior distribution of $\boldsymbol{D}$ and γ.

The normal-gamma prior density function is a member of a conjugate class of distributions, that is the posterior density $\xi(\boldsymbol{D},\gamma \mid \boldsymbol{C})$ is also a normal-gamma density. Conjugate families have the advantage that one has a scale by which to judge the amount of information added by the sample, beyond the amount given *a priori*.

4.4.2. Non-Informative Density

Based on another early work suggested by Jeffreys (1977), we can use a "vague" non-informative prior density for OD flows, that is:

$$\xi(\boldsymbol{D},\gamma) \propto 1/\gamma \qquad (4.12)$$

The Jeffreys' prior implies that, *a priori*, $\boldsymbol{D}$ and γ are independent and that $\boldsymbol{D}$ has a constant density over $\Re^{(\Gamma,I,J)}$ and that the marginal prior density of γ is $\xi_2(\gamma) \propto 1/\gamma$, $\gamma > 0$. The Jeffreys prior density, although improper (not having the same prior and posterior density function distribution), produces a normal-gamma posterior density for D and γ.

4.5. Posterior Analysis

4.5.1. Normal-Gamma Prior Density

Using Bayes' theorem given by (4.3) and using the normal-gamma prior density (4.7), the posterior density of $\boldsymbol{D}$ and γ is obtained as (Broemeling, 1985)

$$\xi^*(\boldsymbol{D},\gamma\,|\,\boldsymbol{C}) \propto \gamma^{((\mathrm{L.T}+2\alpha+\Gamma.\mathrm{I.J})/2)-1}\boldsymbol{e}^{-\frac{\gamma}{2}[2\beta+(\boldsymbol{D}-\mu)^{\boldsymbol{T}}\boldsymbol{\Psi}(\boldsymbol{D}-\mu)+(\boldsymbol{C}-\boldsymbol{P}\,\boldsymbol{D})^{\boldsymbol{T}}(\boldsymbol{C}-\boldsymbol{P}\,\boldsymbol{D})]} \tag{4.13}$$

That is the joint posterior density of $\boldsymbol{D}$ and γ, $\xi^*(\boldsymbol{D},\gamma\,|\boldsymbol{C})$, is a normal-gamma density. The marginal posterior density of γ is gamma with parameters

$$(\mathrm{L.T}+2\alpha) \text{ and } \beta+\frac{\boldsymbol{C}^{\boldsymbol{T}}\boldsymbol{C}-(\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{C}+\boldsymbol{\Psi}\boldsymbol{\mu})^{\boldsymbol{T}}(\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{P}+\boldsymbol{\Psi})^{-1}(\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{C}+\boldsymbol{\Psi}\boldsymbol{\mu})}{2} \tag{4.14}$$

The marginal posterior density of $\boldsymbol{D}$, $\xi^*{}_1(\boldsymbol{D}|\boldsymbol{C})$, is found by integrating (4.13) with respect to γ. Doing so results that marginal posterior density of $\boldsymbol{D}$ is a Γ.I.J-dimensional t density with L.T+2α degrees of freedom, and location vector

$$\mu^* = (\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{P}+\boldsymbol{\Psi})^{-1}(\boldsymbol{P}^{\boldsymbol{T}}\boldsymbol{C}+\boldsymbol{\Psi}\boldsymbol{\mu}) \tag{4.15}$$

Therefore, the posterior analysis of the general linear model reveals that the joint posterior distribution of $\boldsymbol{D}$ and γ is a normal-gamma distribution, the marginal distribution of $\boldsymbol{D}$ is a multivariate t, and the marginal of γ is a gamma if the prior of the parameters is a normal-gamma.

4.5.2. Non-informative Prior Density

The analysis in the second case, where Jeffreys' improper density, $\xi(\boldsymbol{D},\gamma)\propto 1/\gamma$, is used as the prior information, shows that the posterior density of $\boldsymbol{D}$ and γ is normal-gamma where the marginal posterior density of γ is gamma with parameters

$$(\mathrm{L.T} - \Gamma.\mathrm{I.J})/2 \text{ and } \left[\boldsymbol{C}^T\boldsymbol{C} - \boldsymbol{C}^T\boldsymbol{P}(\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{C}\right]/2$$

The conditional posterior density of $\boldsymbol{D}$ given γ is normal with mean vector

$$(\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{C}$$

and precision matrix $\gamma\boldsymbol{P}^T\boldsymbol{P}$.

The marginal posterior density of $\boldsymbol{D}$ is a Γ.I.J-dimensional t distribution with (L.T − Γ.I.J) degrees of freedom and location vector

$$\mathrm{E}(\boldsymbol{D} \mid \boldsymbol{C}) = (\boldsymbol{P}^T\boldsymbol{P})^{-1}\boldsymbol{P}^T\boldsymbol{C}$$

The above equation shows that the results in this case are similar to ordinary least-squares estimates of OD flows (in the upper-level problem of bi-level optimization).

It is worth noting that if we use the normal-gamma density as a prior for $\boldsymbol{D}$ and γ, as in (4.15), $\boldsymbol{P}^T\boldsymbol{P}$ may be singular and OD flows can still be estimated. However, if we use Jeffreys' improper prior, $\boldsymbol{P}^T\boldsymbol{P}$ must be nonsingular, otherwise the posterior density of $\boldsymbol{D}$ and γ is improper.

As mentioned in Chapter 2, the majority of previous studies have included a target matrix in the formulation of the OD-flow estimation problem to overcome the under-specification of the problem. Adding a target matrix results in equations similar to (4.15), which prevents degeneracy of the problem due to singularity of link-flow matrices.

4.6. Point Estimation of OD and Precision Parameter

As previously mentioned, the joint posterior distribution of ***D*** and γ is also normal-gamma with parameters

$$
\begin{aligned}
&\alpha^* = (\mathrm{L.T} + 2)/2 \\
&\beta^* = \frac{2\beta + \boldsymbol{C}^T\boldsymbol{C} - (\boldsymbol{P}^T\boldsymbol{C} + \boldsymbol{\Psi\mu})^T(\boldsymbol{P}^T\boldsymbol{P} + \boldsymbol{\Psi})^{-1}(\boldsymbol{P}^T\boldsymbol{C} + \boldsymbol{\Psi\mu})}{2} \\
&\boldsymbol{\Psi}^* = (\boldsymbol{P}^T\boldsymbol{P} + \boldsymbol{\Psi})\gamma \\
&\boldsymbol{\mu}^* = (\boldsymbol{P}^T\boldsymbol{P} + \boldsymbol{\Psi})^{-1}(\boldsymbol{P}^T\boldsymbol{C} + \boldsymbol{\Psi\mu})
\end{aligned}
\qquad (4.16)
$$

There is no unique solution when ***D*** and γ are estimated jointly, but since ***D*** is a normal random variable with mean $\boldsymbol{\mu}^*$ which does not depend on γ, and since α^* and β^* do not depend on ***D***, it seems that $[\boldsymbol{\mu}^*, \beta^*(\alpha^*\text{-}1)^{-1}]$ is a reasonable choice for a joint estimate of $[\boldsymbol{D}, \gamma^{-1}]$.

The marginal distribution of γ is $\mathrm{G}[\alpha^*, \beta^*]$, hence the mean of γ^{-1} is

$$\mathrm{E}(\gamma^{-1} \mid \boldsymbol{C}) = \beta^*(\alpha^* - 1)^{-1}$$

and its mode is

$$\mathrm{M}(\gamma^{-1} \mid \boldsymbol{C}) = \beta^*(\alpha^* + 1)^{-1}$$

Since the gamma distribution is asymmetric, whether one takes the mean or mode to estimate γ^{-1} is a matter of personal choice.

4.7. Determining Hyperparameters for the Prior Information

The difficulty in using the conjugate class in Bayesian inference is that the hyperparameters in the prior distribution, i.e. the set of parameters ($\boldsymbol{\mu}$, β, $\boldsymbol{\psi}$, γ) in equations (4.16), need to be specified. Consider that we have a set of historical observations for the same time interval. These observations can be for the same time interval the day before, the same weekday the week before or, in the case of annual events, the same day a year before. (Of course, this problem only exists when we are estimating the posterior distribution for the first time; after that the last estimated hyperparameters can be used directly and updated every time).

Assume we have a set of former observations, which would fit in our basic quasi-linear equation (the subscript 'o' is added to the variables to show they belong to an older observation):

$$\boldsymbol{C_o} = \boldsymbol{\hat{P}_o}.\boldsymbol{D_o} + E_o \qquad (4.17)$$

where $\boldsymbol{C_o}$ is $L_o.T_o \times 1$, $\boldsymbol{\hat{P}_o}$ is $L_o.T_o \times \Gamma_o.I_o.J_o$, $\boldsymbol{D_o}$ is $\Gamma_o.I_o.J_o \times 1$, and $\boldsymbol{E_o} \sim (0, \gamma_o^{-1} \boldsymbol{I}_{\boldsymbol{L_o T_o}})$.

To set the values of the hyperparameters, notice that the usual estimators, i.e. the non-informative prior estimates, of $\boldsymbol{D_o}$ and γ_o^{-1} of (4.17) are

$$\boldsymbol{D_o} = (\boldsymbol{\hat{P}_o}^T \boldsymbol{\hat{P}_o})^{-1} \boldsymbol{\hat{P}_o}^T \boldsymbol{C_o} \qquad (4.18)$$

and

$$\gamma_o^{-1} = \frac{\boldsymbol{C_o}^T\boldsymbol{C_o} - \boldsymbol{C_o}^T\hat{\boldsymbol{P}}_o(\hat{\boldsymbol{P}}_o^T\hat{\boldsymbol{P}}_o)^{-1}\hat{\boldsymbol{P}}_o^T\boldsymbol{C_o}}{(\mathrm{L_o.T_o} - \Gamma_o.\mathrm{I_o.J_o})} \tag{4.19}$$

Since the prior mean of $\boldsymbol{D_o}$ is μ, we choose

$$\mu = \boldsymbol{D_o} \tag{4.20}$$

In the same way, since the prior mean of γ_0^{-1} is $\beta/(\alpha\text{-}1)$, we choose α and β such that

$$\gamma_0^{-1} = \beta / (\alpha - 1) \ , \qquad \alpha > 1 \tag{4.21}$$

Of course, the choice of α and β is not unique. The prior dispersion of $\boldsymbol{D}$ is $\beta\boldsymbol{\Psi}(\alpha\text{-}1)^{-1}$, thus we choose $\boldsymbol{\Psi}$ such that $\gamma_o^{-1}\boldsymbol{\Psi} = \gamma_o^{-1}(\hat{\boldsymbol{P}}_o^T\boldsymbol{P}_o)^{-1}$, or we let

$$\boldsymbol{\Psi} = (\boldsymbol{P}_o^T\boldsymbol{P}_o)^{-1} \tag{4.22}$$

Therefore, we have determined the values of the hyperparameters from a former observation, though the choice of α and β is not unique.

4.8. Summary

In this research, the Bayesian inference method is used to direct the estimated OD-flows toward a target matrix. In this chapter, the use of this method to combine the prior information on OD-flows with the information obtained from the sample of link flow observations was discussed. As the experiments conducted in Chapter 5 indicate, if reliable *a priori* information on OD-flows is available, Bayesian inference can significantly improve the quality of the estimation. The improvement is more significant, when due to the congestion in the network, the link-flow observations do not provide enough evidence for reliable OD-flow estimation or when due to inconsistencies in the traffic assignment assumption, the effect of the other sources of error is significant.

CHAPTER 5. ALGORITHM IMPLEMENTATION ASPECTS

5.1. Introduction

In Chapter 3, several methods for estimating dynamic OD-flows from time-dependent traffic count data were presented. The two major methods presented are the generalized least-squares estimation and the non-linear (least-squares) optimization method.

Based on the estimation period, two principal variations to the methods are considered:

- Single-horizon formulation, primarily for planning applications, and
- Rolling-horizon formulation, with primary application to real-time network traffic management.

The Rolling-horizon case is formulated in two different ways:

- Fixed initial-point formulation, and
- Free initial-point formulation.

In Chapter 4, a Bayesian inference method was adapted to systematically incorporate *a priori* information on OD-flows with the estimated OD-flows from traffic counts.

The procedures for the estimation of OD flows from traffic counts are implemented as an integral part of the DYNASMART-P simulation program. Between the two options presented for formulation of the rolling-horizon OD-flow estimation, the first alternative, i.e. fixed initial-point formulation, is implemented.

Issues pertaining to the implementation of the proposed methods are discussed in this chapter. In the next section, the major algorithmic steps for estimation of OD flows using linear and non-linear optimization methods and Bayes' theorem are presented. In Section 5.3, the issues relating to the rolling-horizon implementation of the estimation methods are discussed. The method of successive averaging of the consecutive estimated OD flows are explained in Section 5.4. In Section 5.5, the numerical method adopted for solving the set of simultaneous quadratic equations entailed in the bi-level non-linear optimization formulation is described and its convergence issues are discussed. In Section 5.6, the concluding remarks are presented.

5.2. **Procedural Steps of OD-Flow Estimation**

Figure 5.1 illustrates the main algorithmic steps of the methods presented in this dissertation for the estimation of OD flows from traffic flow observations. The algorithm includes bi-level linear generalized least-squares estimation, bi-level non-linear least-square estimation and Bayes' inference method. The functions, set of variables, input/output files and pseudo codes for each of the modules in the algorithm are presented in appendices to this dissertation. The modules used in the Dynamic Traffic Assignment program (DYNASMART-P) are explained in its user's guide (Mahmassani *et al.*, 2000).

Step 0. Initialize all arrays.

Step 1. Read the link flow observations.

Step 2. Read the initial estimates of the OD flow matrix, $\hat{\boldsymbol{D}}^{0}$.

Step 3. Initialize the iteration counter, i=0.

Step 4. Solve the lower-level optimization problem by assigning $\hat{\boldsymbol{D}}^{\boldsymbol{i}}$ to the network using a dynamic traffic assignment simulation program (DYNASMART-P).

Step 5. Calculate the link-flow proportions, $\hat{\boldsymbol{P}}$.

Step 6. Calculate the statistics of the estimate, i.e. the errors in the link flow estimation and, in the case of conducting experiments where OD demand matrix is presumed, the errors in the estimated demand values.

Step 7. Increment the iteration counter, i=i+1.

Step 8. If i is less than the required number of iterations, or if i is equal to the required number of iterations and the non-linear estimation flag is OFF, go to Step 9.

If i is equal to the required number of iterations and the non-linear estimation flag is ON, go to Step 10.

If i is equal to the required number of iterations plus one and Bayes flag is OFF, go to Step 11.

If i is equal to the required number of iterations plus one and Bayes flag is ON, go to Step 12.

Otherwise, go to Step 14.

Figure 5.1. The OD-flow estimation algorithm

Step 9. Find the generalized least-square estimates of the OD flows, $\hat{\boldsymbol{D}}$, from the identity equation $\boldsymbol{C} = \hat{\boldsymbol{P}} \times \hat{\boldsymbol{D}} + \boldsymbol{E}$ using equations (3.13) or (3.18), and go to Step 13.

Step 10. Compute $\hat{\boldsymbol{D}}$ from the identity equation $\boldsymbol{C} = \hat{\boldsymbol{P}} \times \hat{\boldsymbol{D}} + \boldsymbol{E}$ using equations (3.22) or (3.34), and go to Step 13.

Step 11. Compute the hyperparameters of the prior information using equations (4.19) to (4.22), and go to Step 14.

Step 12. Compute $\hat{\boldsymbol{D}}^{\boldsymbol{i}}$ by updating the prior information with the information obtained from the link-flow observations using equation (4.16), and go to Step 14.

Step 13. Optionally find the weighted average of the estimated OD flow values, i.e. $\hat{\boldsymbol{D}}^{\boldsymbol{i}} = (\hat{\boldsymbol{D}} + \hat{\boldsymbol{D}}^{\boldsymbol{i}-1} \times \boldsymbol{i}) / (\boldsymbol{i} + 1)$, or simply let $\hat{\boldsymbol{D}}^{\boldsymbol{i}} = \hat{\boldsymbol{D}}$. Go to Step 4.

Step 14. Save the results and STOP.

Figure 5.1. continued

In Step 0, the arrays and the variables used in the OD-flow estimation and DTA modules are initialized. The link-flow observations, which are the essential input for the estimation method, are read in Step 1. It should be noted that the algorithm is implemented in a way that link flow observations need not be reported on sequential links or in sequential observation intervals. In the implementation, a procedure is used to record the links and the observation intervals with flow measurements (see Section B.5.2 in Appendix B).

The initial guess of the OD flows are read in Step 2. The closer these initial values are to the actual OD flows, the fewer iterations would typically be needed for obtaining the solution. On the other hand, if the initial guess is too far away from the actual demand flows, the algorithm might converge to other locally optimal solutions. If no information on OD flows are available, a uniform initial demand table may be assumed.

In Step 3, the iteration counter is initialized. In Steps 4 and 5 , using the DTA simulation program, the lower-level optimization problem is numerically solved to find the estimates of link-flow proportions, i.e.

$$\hat{\boldsymbol{p}}_{(\boldsymbol{l},\boldsymbol{t})(\tau,\boldsymbol{i},\boldsymbol{j})} = \frac{\hat{\boldsymbol{f}}_{(\boldsymbol{l},\boldsymbol{t})(\tau,\boldsymbol{i},\boldsymbol{j})}}{\hat{\boldsymbol{d}}_{(\tau,\boldsymbol{i},\boldsymbol{j})}} \tag{5.1}$$

where $\hat{\boldsymbol{p}}_{(\boldsymbol{l},\boldsymbol{t})(\tau,\boldsymbol{i},\boldsymbol{j})}$ is the estimate of link-flow proportions pertinent to link l, observation interval t, aggregate departure interval τ and OD pair i-j, as defined in Section 3.2. The variable $\hat{\boldsymbol{d}}_{(\tau,\boldsymbol{i},\boldsymbol{j})}$ is the initial guess or, in the subsequent iterations, the current estimate of OD flows between OD pair i-j that depart during aggregate departure interval τ. The estimate of the partial link flows observed on link l and observation interval t that is generated by any OD flow $\hat{d}_{(\tau,i,j)}$ is denoted by $\hat{f}_{(l,t)(\tau,i,j)}$. The partial link flow values, which are the output of Step 4 of the algorithm, are estimated in the simulation/assignment program by tracing each vehicle temporally and spatially in the network. The simulation/assignment program emulates the users' presumed route choice behavior and may iterate several times until convergence in terms of the assigned paths of the vehicles is obtained.

In Step 6, the resulting link flows are compared with the actual link flow observations. In the experiments performed in this dissertation, where the actual OD flows are presumed to produce the "ground-truth" link flows, the OD flow estimates are also compared to the assumed actual OD flow values. The Root Mean Squares of Errors (as defined in Section 6.2) along with other statistics may be used for this purpose.

In Step 7, the iteration counter is incremented. In the Bi-GLS estimation method, the algorithm continues in Step 9. In this Step, using the obtained link-flow proportions from Step 5 and the actual link-flow observations from Step 1, the OD flow estimates are updated using equations (3.13) or (3.18). Optionally in Step 13, the OD flows obtained are averaged with the estimated OD flows in the previous iterations. As an alternative, one may ignore the averaging procedure and use the values obtained in Step 9 as the final estimate of the OD flows in the current iteration. The procedures mentioned in Steps 4, 5, 6, 7 and 9 are repeated several times as pre-specified by the 'required number of iterations' parameter.

In the last iteration of the algorithm and after Step 7, if the non-linear optimization is requested, the algorithm continues in Step 10. In this step, based on the latest estimated link-flow proportions (Step 5) and the actual link flow values (Step 1), the OD flows are updated by using equations (3.22) or (3.44). The numerical solution to the set of simultaneous quadratic equations is described in Section 5.5. Optionally, the method of successive averages may be used (Step 13), and Steps 4, 5, 6 and 7 are repeated.

In Step 11, after the last iteration of the algorithm, the parameters of the prior distribution are computed according to Bayes' theorem. These parameters can be used in the subsequent OD-flow estimation as *a priori* information for similar estimation periods. If Bayesian inference is desired and the prior parameters are known, the posterior OD flows and distribution parameters are computed in Step 12.

For a more detailed description of each step of the algorithm, readers are referred to appendices A through E.

5.3. Implementing Rolling Horizon OD-Flow Estimation

Between the two formulations presented in Sections 3.10 and 3.11 for the rolling-horizon OD-flow estimation, the former, i.e. the fixed initial-point estimation, is implemented. As explained in this section, the implemented single-horizon OD flow estimation method can be easily adapted to the fixed initial-point estimation case. However, the free initial-point estimation method needs an explicit implementation that is substantially different from the single-horizon implementation. The implementation of the free initial-point estimation method is left for future work.

In the fixed initial-point estimation, the simulation period is divided into two parts: loading and estimation periods (Figure 5.2). The demand flows in the loading period are assumed equal to their estimates in the previous estimation periods. It should be noted that the estimated OD-flows toward the end of each estimation period have high variance because some of the vehicles may not have reached their destination. Therefore, it is highly recommended that the rolling windows have temporal overlap in consecutive estimation periods, as illustrated in Figure 5.2.

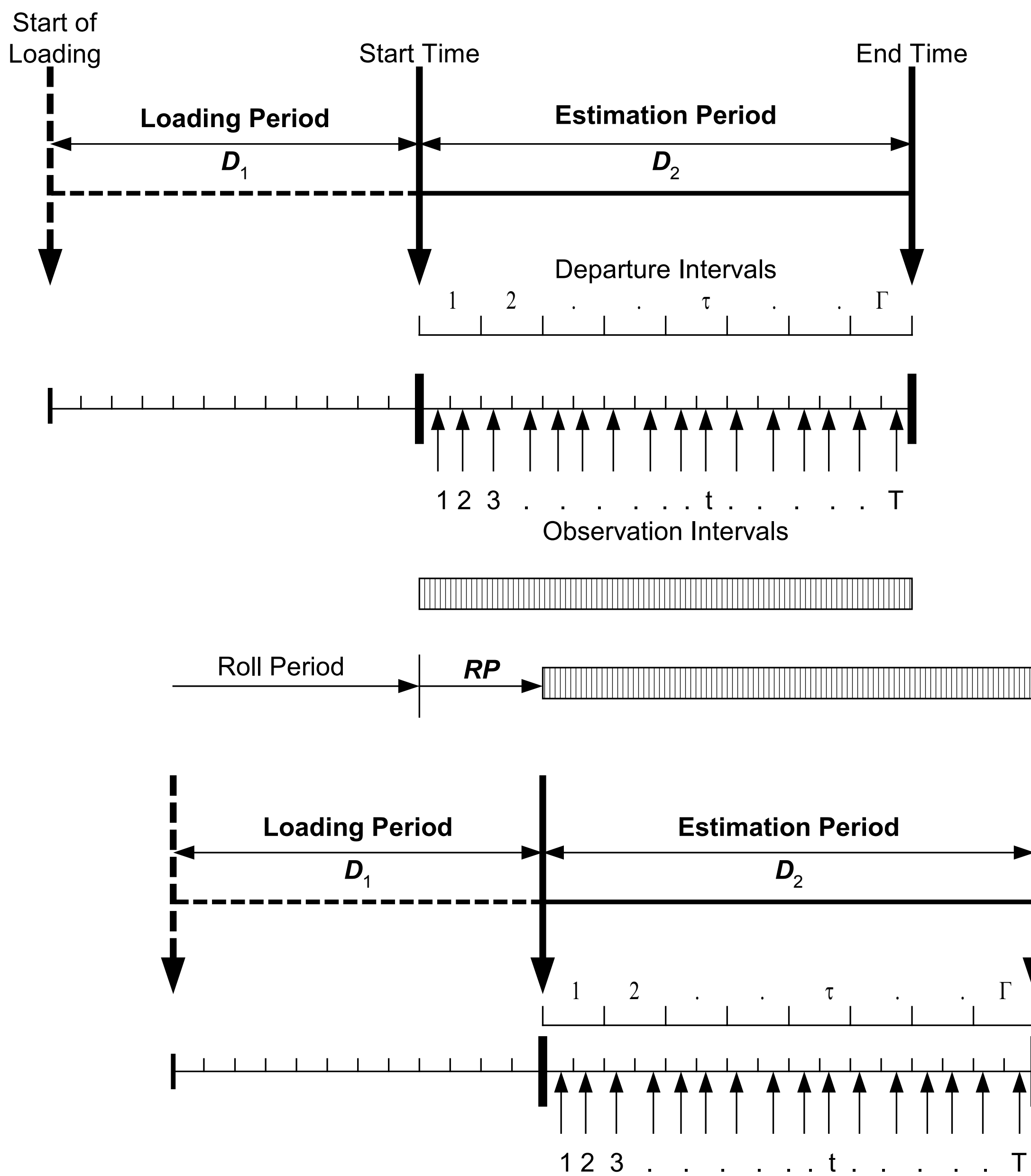


Figure 5.2. Definition of time intervals in the rolling-horizon implementation

In this implementation, the link-flow proportion matrix and the OD-flow vector are accordingly divided into loading and estimation periods. Figure 5.3 illustrates the partitioning of the link-flow proportion matrix and the OD-flow vector. As a reminder, the rows of the link-flow proportion represent the combination of link and observation intervals and the columns represent the combination of departure intervals and the OD pairs.

$\boldsymbol{pl^{l,t}_{\tau',k}}$ is the loading period link-flow proportion element pertaining to link *l*, observation interval *t*, aggregate departure interval τ' (within the loading period) and the k^{th} OD pair (OD pair *ij*).

$\boldsymbol{pe^{l,t}_{\tau,k}}$ is the same as above except for the link-flow proportion element within the estimation period.

$\boldsymbol{dl_{\tau',k}}$ is the demand departing during the aggregate departure interval τ' in the loading period and between OD-pair *k*.

$\boldsymbol{de_{\tau,k}}$ is the same as above except for the demand departing during the aggregate departure interval τ in the estimation period.

$\boldsymbol{c_{l,t}}$ is the link-flow observation on link *l* and in observation interval *t* within the estimation period.

It should be noted that in the above notation the observation interval *t* in all cases is within the estimation period and not the loading period.

The example shown in Figure 5.3, without loss of generality, depicts a case where the loading period has one aggregate departure interval, consisting of ten observation intervals. In this example, there are K OD pairs, Γ departure intervals, L links with flow observations and T observation intervals. It should be noted that the rows of the link-flow proportion matrix and the observation vector only include the elements pertaining to the estimation period (i.e. in the example, the rows start from the observation interval 11), while in the link-flow proportion matrix, the columns include the elements pertaining to the loading period and the estimation period. The previously estimated OD-flows are augmented at the top of the OD-flow vector (loading period OD-flows).

$$
\left[\begin{array}{cccc|ccccc}
pl_{1,1}^{1,11} & pl_{1,2}^{1,11} & \cdots & pl_{1,K}^{1,11} & pe_{1,1}^{1,11} & \cdots & pe_{1,K}^{1,11} & pe_{\Gamma,1}^{1,11} & \cdots\ pe_{\Gamma,K}^{1,11} \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & \vdots \\
pl_{1,1}^{L,1} & pl_{1,2}^{L,1} & \cdots & pl_{1,K}^{L,1} & pe_{1,1}^{L,1} & \cdots & pe_{1,K}^{L,1} & pe_{\Gamma,1}^{L,1} & \cdots\ pe_{\Gamma,K}^{L,1} \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & \vdots \\
\vdots & \textit{loading period} & \vdots & \vdots & & & & \textit{estimation period}\ \vdots & \vdots \\
\vdots & \vdots & & \vdots & & & \vdots & \vdots & \vdots \\
pl_{1,1}^{1,T} & pl_{1,2}^{1,T} & \cdots & pl_{1,K}^{1,T} & pe_{1,1}^{1,T} & & pe_{1,K}^{1,T} & pe_{\Gamma,1}^{1,T} & \cdots\ pe_{\Gamma,K}^{1,T} \\
\vdots & \vdots & \cdots & \vdots & & & \vdots & \vdots & \vdots \\
pl_{1,1}^{L,T} & pl_{1,2}^{L,T} & \cdots & pl_{1,K}^{L,T} & pe_{1,1}^{L,T} & \cdots & pe_{1,K}^{L,T} & pe_{\Gamma,1}^{L,T} & \cdots\ pe_{\Gamma,K}^{L,T}
\end{array}\right]
\times
\left[\begin{array}{c}
dl_{1,1} \\ \textit{loading} \\ dl_{1,K} \\ \hline\hline de_{1,1} \\ \vdots \\ \textit{estimation} \\ \vdots \\ \vdots \\ de_{\Gamma,K}
\end{array}\right]
\equiv
\left[\begin{array}{c}
c_{1,11} \\ \vdots \\ c_{L,11} \\ \vdots \\ \vdots \\ c_{l,t} \\ \vdots \\ \vdots \\ c_{1,T} \\ \vdots \\ c_{L,T}
\end{array}\right]
$$

Figure 5.3. Partitioning of matrices in a rolling horizon implementation

The (unknown) OD-flows during the estimation period should generate a net flow, as computed below:

Net flow = total flow – (link-flow proportion)×(loading-period demand flows)

That is, the estimate of the net flow is calculated from the following equation:

$$\hat{nc}_{l,t} = c_{l,t} - \sum_{\tau \in loading\ period} \sum_{k=1}^{K} \hat{pl}_{\tau,k}^{l,t} . \hat{dl}_{\tau,k} \qquad (5.2)$$

where

$\hat{nc}_{l,t}$	is the estimate of the net flow on link l during observation interval t.
$c_{l,t}$	is the observed link flow on link l during observation interval t.
$\hat{pl}_{\tau,k}^{l,t}$	is the estimate of link-flow proportion pertinent to link l, observation interval t, OD pair k and departure interval τ in the loading period.
$\hat{dl}_{\tau,k}$	is the estimate of OD-flow between OD pair k during aggregate departure interval τ in the loading period. The estimated value in the previous rolling-horizon estimation period can be used for this variable.
K	is the total number of OD pairs (equivalent to I.J).

If $\hat{P}_{loading}$ and $\hat{D}_{loading}$ respectively denote the link-flow proportion matrix and the OD-flow vector of the loading period, equation (5.2) can be rewritten in the matrix form:

$$\hat{NC} = C - \hat{P}_{loading} . \hat{D}_{loading}$$

5.4. Method of Successive Averages to Estimate OD-Flows

To address the convergence issues and to make the OD-flow estimation procedure more stable, the method of successive averages (MSA) is used in the consecutive iterations of the algorithm to avoid jumping from one local optimal region to another. As mentioned in Step 13 of Figure 5.1:

$$\hat{\boldsymbol{d}}^{\boldsymbol{r}}_{(\boldsymbol{\tau},\boldsymbol{k})} = \frac{\left[\hat{\boldsymbol{d}}^{*\boldsymbol{r}}_{(\boldsymbol{\tau},\boldsymbol{k})} + \hat{\boldsymbol{d}}^{\boldsymbol{r}-1}_{(\boldsymbol{\tau},\boldsymbol{k})} \times \boldsymbol{r}\right]}{\boldsymbol{r}+1} \qquad (5.3)$$

where

$\hat{\boldsymbol{d}}^{*\boldsymbol{r}}_{(\boldsymbol{\tau},\boldsymbol{k})}$ is the estimated OD-flow between OD pair *k* departing in the aggregate departure interval *τ*. This value is the direct result of the estimation process in its r^{th} iteration.

$\hat{\boldsymbol{d}}^{\boldsymbol{r}}_{(\boldsymbol{\tau},\boldsymbol{k})}$ is the modified value of the estimated OD-flow between OD pair *k* departing in the aggregate departure interval *τ*. This variable may be used as the estimate of the OD-flow $\boldsymbol{d}_{(\boldsymbol{\tau},\boldsymbol{k})}$ in the r^{th} iteration of the algorithm.

$\boldsymbol{r}$ is the iteration number.

It should be noted that the use of successive averaging in the implemented algorithm is optional.

The convergence issues in the bi-level formulation of the problem, in some aspects may be similar to the Expectation-Maximization (EM) algorithms mostly used as a numerical technique for the evaluation of maximum likelihood estimates of the parameters describing incomplete data sets (Dempster, Laird and Rubin, 1977). Researchers have applied this method in transportation modeling applications (Bhat,

1997) and there have been several discussions on the convergence of the algorithm and on the theorems initially proposed by Dempster *et al.* (Wu, 1983; Boyles 1983).

It should be noted that the formulated optimization problem for estimation of OD flows in a dynamic system is not well behaved. Though the quadratic optimization problem in the upper level is convex, the dynamic traffic assignment problem in the lower level inherently does not have unique solution. Therefore, there is no theoretical proof that the iterative bi-level GLS estimation described in Section 3.4 or the algorithmic steps depicted in Figure 5.1 for either Bi-GLS or Bi-NLP estimation will converge. The issue is discussed in detail in the next section where the more general non-linear optimization method is used in the upper-level problem.

5.5. Estimation of OD-Flows by Non-Linear Optimization

As explained in Sections 3.5 and 3.6, the non-linear optimization formulation for the OD-flow estimation entails solution of a set of simultaneous quadratic equations. In the following two subsections, the numerical method used to solve the set of equations and the issues regarding the convergence of the solution are discussed.

5.5.1. Numerical Solution of the Set of Simultaneous Quadratic Equations

When the derivatives of the link-flow proportion with respect to demand are included in the formulation, a set of simultaneous quadratic equations, shown in equation (3.34), should be solved. The set of equations are rewritten as follows:

$$0 = \boldsymbol{f}_{(v,o,d)}(\boldsymbol{d}_1, \boldsymbol{d}_2, \boldsymbol{d}_3, ..., \boldsymbol{d}_{\Gamma IJ}) = \sum_{m=1}^{\Gamma IJ} \sum_{n=1}^{\Gamma IJ} \boldsymbol{a}_{mn}^{(v,o,d)} \boldsymbol{d}_m \boldsymbol{d}_n - \sum_{m=1}^{\Gamma IJ} \boldsymbol{b}_m^{(v,o,d)} \boldsymbol{d}_m - \boldsymbol{g}^{(v,o,d)} \tag{5.4}$$

where *m* and *n* are sequential index numbers for each time-dependent OD pair.

An iterative procedure is used to solve the above set of Γ.I.J equations. In each iteration of the algorithm, the equations are linearized using Taylor series expansion (expanded only up to the first degree derivatives). The set of linearized equations to be solved in iteration *r* is:

$$
\begin{aligned}
0 &= (f_1) + (f_1)_{d_1}(d_1 - \hat{d}_1^{r-1}) + (f_1)_{d_2}(d_2 - \hat{d}_2^{r-1}) + (f_1)_{d_3}(d_3 - \hat{d}_3^{r-1}) + \cdots \\
0 &= (f_2) + (f_2)_{d_1}(d_1 - \hat{d}_1^{r-1}) + (f_2)_{d_2}(d_2 - \hat{d}_2^{r-1}) + (f_2)_{d_3}(d_3 - \hat{d}_3^{r-1}) + \cdots \\
0 &= (f_3) + (f_3)_{d_1}(d_1 - \hat{d}_1^{r-1}) + (f_3)_{d_2}(d_2 - \hat{d}_2^{r-1}) + (f_3)_{d_3}(d_3 - \hat{d}_3^{r-1}) + \cdots \\
&\vdots \\
&\vdots \\
0 &= (f_{\Gamma IJ}) + (f_{\Gamma IJ})_{d_1}(d_1 - \hat{d}_1^{r-1}) + (f_{\Gamma IJ})_{d_2}(d_2 - \hat{d}_2^{r-1}) + (f_{\Gamma IJ})_{d_3}(d_3 - \hat{d}_3^{r-1}) + \cdots
\end{aligned}
\tag{5.5}
$$

where

$(f_p)_{d_q}$ is the derivative of the p^{th} equation in (5.4) with respect to the q^{th} sequential time-dependent OD pair.

$\hat{d}_n^{r-1}$ is the estimated value of n^{th} sequential OD-flow in iteration r-1.

d_n is the (unknown) OD-flow value of the n^{th} sequential time-dependent OD pair.

All functions f_p and their derivatives $(f_p)_{d_q}$ are evaluated at the current value of the demand flow, which is estimated in the previous iteration r-1, i.e. $\hat{d}_n^{r-1}$.

In each iteration r, the linearized equations (5.5) are solved for the unknowns $(d_n - \hat{d}_n^{r-1})$. These unknown values should be added to the estimated OD flows in the previous iteration, $\hat{d}_n^{r-1}$, to obtain an improved estimate of the OD flows d_n. In other words:

$$
\hat{d}_n^r = \hat{d}_n^{r-1} + (d_n - \hat{d}_n^{r-1}) = d_n \tag{5.6}
$$

The iteration is repeated until the amount of incremental correction becomes small or a pre-specified number of iterations are performed. For more detailed implementation aspects of this method, readers are referred to Appendix C.

5.5.2. Convergence Issues

In practice, it is not always possible to find a solution to the set of equations in (5.5). Consider a hypothetical case with one departure interval and only two unknown demand flows. A typical solution is illustrated graphically in Figure 5.4.

In each iteration, each surface is approximated by planes tangent to the surface at the current solution point. Although in Figure 5.5 the process is shown by fitting a tangent *line* to the two curves at an arbitrary solution point $\boldsymbol{d}_1$, in reality each function $\boldsymbol{f}_i$ $(\boldsymbol{d}_1,\boldsymbol{d}_2)$ represents a surface where the height from the zero-height plane (the plane containing the coordinate axes $\boldsymbol{d}_1$ and $\boldsymbol{d}_2$) is equal to the value of the function $\boldsymbol{f}_i$ at the point $(\boldsymbol{d}_1,\boldsymbol{d}_2)$. Therefore, the algorithm computes the equations of the tangent *planes* to the surfaces (first-degree Taylor expansion of the function) at each point $(\boldsymbol{d}_1,\boldsymbol{d}_2)$. If we call the approximated function representing the tangent plane $\bar{\boldsymbol{f}}_i(\boldsymbol{d}_1,\boldsymbol{d}_2)$, the equation of the intersection line along the two tangent planes can be represented by:

$$\bar{\boldsymbol{f}}_1(\boldsymbol{d}_1,\boldsymbol{d}_2)=\bar{\boldsymbol{f}}_2(\boldsymbol{d}_1,\boldsymbol{d}_2)\,. \qquad (5.7)$$

The coordinates of the point where this line intersects the zero-height surface is the solution to the set of linearized equations at the current iteration. In other words, this point is the solution to the set of equations:

$$\bar{\boldsymbol{f}}_1(\boldsymbol{d}_1, \boldsymbol{d}_2) = \bar{\boldsymbol{f}}_2(\boldsymbol{d}_1, \boldsymbol{d}_2) = 0 \tag{5.8}$$

As depicted in Figure 5.4, there may exist multiple solutions to a set of quadratic simultaneous equations. If one starts from a point close enough to A (if the process converges), the solution will collapse at A, and if the starting point is close to B, the solution will collapse at B. If the starting point is away from A or B, the iterative procedure might never converge. Furthermore, there might not exist any solution to the set of equations.

Bard (1998) has presented the required conditions for the uniqueness of the solution to a bi-level non-linear optimization problem. If in the lower-level problem, the follower's rational reaction set, as defined below, is not single-valued, the leader in the upper-level problem may not achieve its minimum objective value. The follower's rational set, P(***d***), is the set of values that the follower takes to minimize his objective function, for any selected value by the leader in the upper-level problem (Bard, 1998).

In the context of OD-flow estimation, the follower's rational set is the set of link-flow proportion matrices obtained from optimization in the lower-level problem, given the optimal demand obtained in the upper level. Though the problem in the upper level is convex, the dynamic traffic assignment problem in the lower level is not well behaved and does not have a unique optimal solution. Therefore, since the follower's rational reaction set is not single-valued, the upper-level problem may not obtain a unique optimal value. This implies that the iterative procedure for solving the problem may not converge to a unique solution.

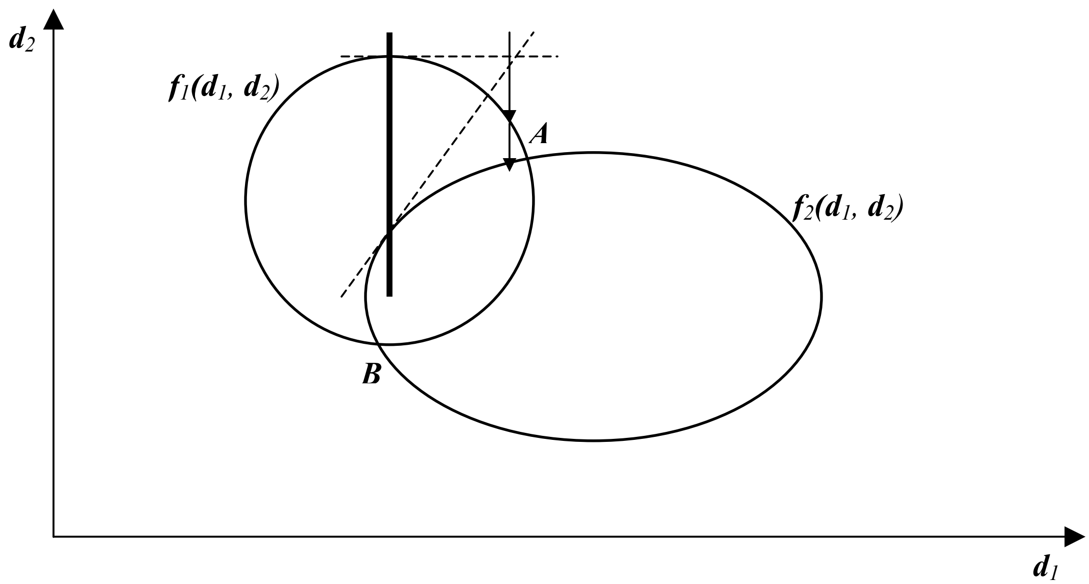


Figure 5.4. A graphical solution to a set of quadratic simultaneous equations

In the realm of dynamic OD-flow estimation, the problem of convergence aggravates as the number of unknowns increases. In order to converge to the expected solution, the implemented algorithm described in Section 5.2 is designed such that the process is iterated a predefined number of times using only the GLS OD-flow estimation method. When the solution converges in the neighborhood of its local optimum, a single run of the non-linear optimization method is executed. As shown in Section 6.5.7, the amount of improvement in the estimation's performance might not be significant. If the procedure is executed before GLS estimation converges, the process may not converge at all.

5.6. Summary

In this chapter, the procedural steps for estimation of dynamic OD flows from traffic counts using bi-level generalized least-squares, non-linear programming and Bayes' inference methods were presented. Furthermore, the implementation aspects of the rolling-horizon estimation method were elaborated. As mentioned, the method of successive averages was adopted to obtain a more stable solution in consecutive estimation iterations. In closing, the numerical algorithm for solving the non-linear optimization method was presented and the issues regarding the convergence of algorithm were discussed. The reader is referred to the appendices for more detailed implementation aspects of the proposed algorithms.

CHAPTER 6. EXPERIMENTS

6.1. Introduction and Objectives

As mentioned in Chapter 1, we have two main objectives in estimating the OD flows:

1) Estimate the time-dependent OD-flow values as close as possible to their true values.
2) Improve the external consistency of the DTA simulation program in terms of the estimated flows on links, that is when the estimated OD flows are assigned to the network, the estimated time-varying traffic volumes would be as close as possible to the observed flows.

Unfortunately, no real time-dependent OD-flow table was available to measure the actual performance of the proposed methods in terms of the first objective. Therefore, in all the experiments, it is assumed that the true time-dependent OD-flow table is known. The assumed demand is then loaded onto the network using DYNASMART-P as a simulator program. The resulting simulated flows are considered the ground-truth observations on the network links. The OD-flow estimation process requires an initial estimate of the demand table as its starting point. Therefore, an initial guess of the OD-flow table is assumed and the proposed iterative procedures are then executed.

To evaluate the performance of the proposed methods in fulfilling the main objectives of the study, several experiments have been designed. The experimental objectives are to evaluate the following aspects:

- Performance of different algorithms in terms of the quality of the solution
- Convergence of proposed algorithms
- Computational performance
- Sensitivity of algorithms to various endogenous parameters and exogenous factors

In Section 6.2, we describe the adopted measures of performance for evaluating each objective. In Section 6.3, the endogenous parameters and the exogenous factors considered in the experiments are addressed. The networks used in the experiments are presented in Section 6.4. In Section 6.5, the design of experiments and their numerical results are discussed in detail. Concluding remarks are presented in Section 6.6.

6.2. Measure of Performance

The two main research objectives are improvement in the quality of the OD-flow estimation and improvement in the external consistency of the DTA system. To quantify the performance of the estimation method or to examine the effect of the experimental factors, the Root Mean Square of Errors of OD-flow estimates and traffic flows are used. These measures of performance specify the quality of solution in terms of the main two research objectives.

$$\boldsymbol{RMSE_d} = \sqrt{\frac{\sum_{\tau=1}^{\Gamma}\sum_{i=1}^{I}\sum_{j=1}^{J}(\hat{\boldsymbol{d}}_{(\tau,i,j)} - \boldsymbol{d}_{(\tau,i,j)})^2}{\Gamma \times \mathrm{I} \times \mathrm{J}}} \qquad (6.1)$$

$$\boldsymbol{RMSE_v} = \sqrt{\frac{\sum_{l=1}^{L}\sum_{t=1}^{T}(\hat{\boldsymbol{v}}_{(l,t)} - \boldsymbol{v}_{(l,t)})^2}{\mathrm{L} \times \mathrm{T}}} \qquad (6.2)$$

where

$\boldsymbol{RMSE_d}$	is the root mean square of errors of the estimated OD flows.
$\boldsymbol{RMSE_v}$	is the root mean square of errors of the estimated link flows.
$\boldsymbol{d}_{(\tau,i,j)}$	is the actual demand departing in the aggregate departure interval τ going from zone *i* to zone *j*.
$\hat{\boldsymbol{d}}_{(\tau,i,j)}$	is the estimated demand departing in the aggregate departure interval τ going from zone *i* to zone *j*.
$\boldsymbol{v}_{(l,t)}$	is the link flow in observation interval *t* on link *l*.
$\hat{\boldsymbol{v}}_{(l,t)}$	is the estimated link flow in observation interval *t* on link *l*.
I	is the number of origin zones.
J	is the number of destination zones.
L	is the number of links with flow observations in the network.
Γ	is the number of aggregate departure intervals.

Convergence of the algorithms is measured by the amount and the direction of change in the above variables, in consecutive iterations. For this purpose, the RMSEs of estimates in each iteration are illustrated graphically for each of the conducted experiments.

The computational performance has not been measured directly. However, its order of magnitude in the comparison between different algorithms is discussed wherever applicable.

Sensitivity of the algorithms to different endogenous and exogenous factors is measured in terms of the mentioned RMSE variables and the rate of convergence in the solution.

It should be noted that in all the experiments it is assumed that the error terms, ε, are independently and identically distributed (i.i.d.), that is the variance-covariance matrix, $\boldsymbol{W}$, presented in the previous chapters, is substituted by the identity matrix $\boldsymbol{I}$.

6.3. Experimental Factors

As mentioned in Section 6.1, the experiments are intended to examine the performance of the proposed estimation methods and their sensitivity to different endogenous and exogenous factors. Specifically, the effects of the following factors are investigated.

6.3.1. Congestion Level

The performance of the estimation methods is examined under different OD-flow levels. The network is loaded at different congestion levels referred to as uncongested, congested and over-congested.

6.3.2. Route-Choice Assumptions

As mentioned in Chapter 3, in solving the lower-level problem, a traffic assignment model should be presumed. The need for this assumption is not specific to the methods presented in this research, but generally, in transportation planning or in the estimation of static demand flows from traffic counts, some assumptions regarding user behavior or the link cost functions must be made. The more realistic these assumptions are, the more consistent the estimated flows on the links will be.

The errors due to the estimation of OD-flows and traffic assignment are confounded (Sections 1.1 and 3.3). To investigate the effect of the trip-maker's route-choice on the quality of the OD-flow estimation, experiments are conducted by simulating different route-choice behaviors for real-world observations.

6.3.3. Effect of Imposing Upper Limits on Estimated OD Flows

As a heuristic solution to a constrained optimization problem in the implemented algorithms, one can set the upper and lower limits on the estimated OD. These limits emulate the constraints of the optimization problem. The lower limit assures the non-negativity of the results, however setting it to a small positive value makes adjustments to the estimated OD-flows in consecutive iterations possible. More specifically, say in the Bi-GLS method, if an OD-flow value is zero, solving the lower-level problem results in zero values in the corresponding column of the link-flow proportion matrix. Therefore, the estimated OD-flow in the next iteration, regardless of the link-flow values, will remain zero—refer to Equations (3.13) or (3.18). If there are more than one dynamic OD flows with zero values, the link-flow proportion matrix becomes singular and the solution degenerates.

The upper limit may be chosen based on experience. This limit can expedite the convergence of the process and may prevent assigning unacceptable values to OD flows, especially when the estimation process degenerates due to the singularity in the link-flow proportion matrix. To investigate the effect of this upper value on the quality of the solution and its convergence, a set of experiments are designed and conducted.

6.3.4. Network Size Effect

As explained in Chapter 3, a large number of parameters and unknown variables emerge in the estimation of dynamic OD flows. This property might make the application of the proposed methods to a large network infeasible. To investigate whether the proposed bi-level GLS OD-flow estimation method can be implemented in large networks, two experiments are conducted on a large network.

6.3.5. Effect of Observation Intervals and Departure Intervals Sizes

The ratio of departure interval size to observation interval size determines the degree of over-specification of the problem (Section 3.3). To investigate the effect of these parameters on the performance of OD-flow estimation and the external consistency of the simulation program, a set of experiments with varying observation intervals and departure interval sizes are designed and conducted.

6.3.6. Non-Linear vs. Linear Optimization Algorithms

The performance of the non-linear optimization method in the estimation of OD flows is investigated by implementing it on a small network with a limited number of departure intervals. The practical problems involved in implementing this method on large networks are explained in Chapter 5.

6.3.7. Effect of *a Priori* Information on Estimation Quality

As explained in Chapter 4, the Bayesian inference method is proposed to incorporate *a priori* information in the estimation process. To examine how the existing information on OD flows can improve the quality of the solution, several experiments are designed assuming that *a priori* information on OD flows is available.

6.4. Test Networks

The experiments are conducted on one real and two hypothetical test networks. The first test network, 'Network A', is a small network and consists of only two origin and destination zones (Figure 6.1). There are 14 links including one freeway segment, six nodes, inclusive of the origin and destination nodes. All the intersections are controlled by STOP signs. This network, except for a few reported experiments, was used primarily for test and development purposes.

The second network, 'Network B' is a medium-sized network consisting of 22 nodes, 68 links and six origin and destination zones (Figure 6.2). Fourteen intersections are controlled by pre-timed signals, and the remaining eight do not have any control. Most of the experiments are conducted on this network.

The large 'FW Network' represents the south central corridor in Fort Worth, Texas (Figure 6.3). This network consists of 13 origin and destination zones, 178 nodes and 441 links. It includes a major freeway section (IH-35) between I-20 and I-

30. Sixty-one of the intersections are controlled by pretimed signals, thirty-one by STOP signs, twenty-four by YIELD signs and sixty-two have no control.

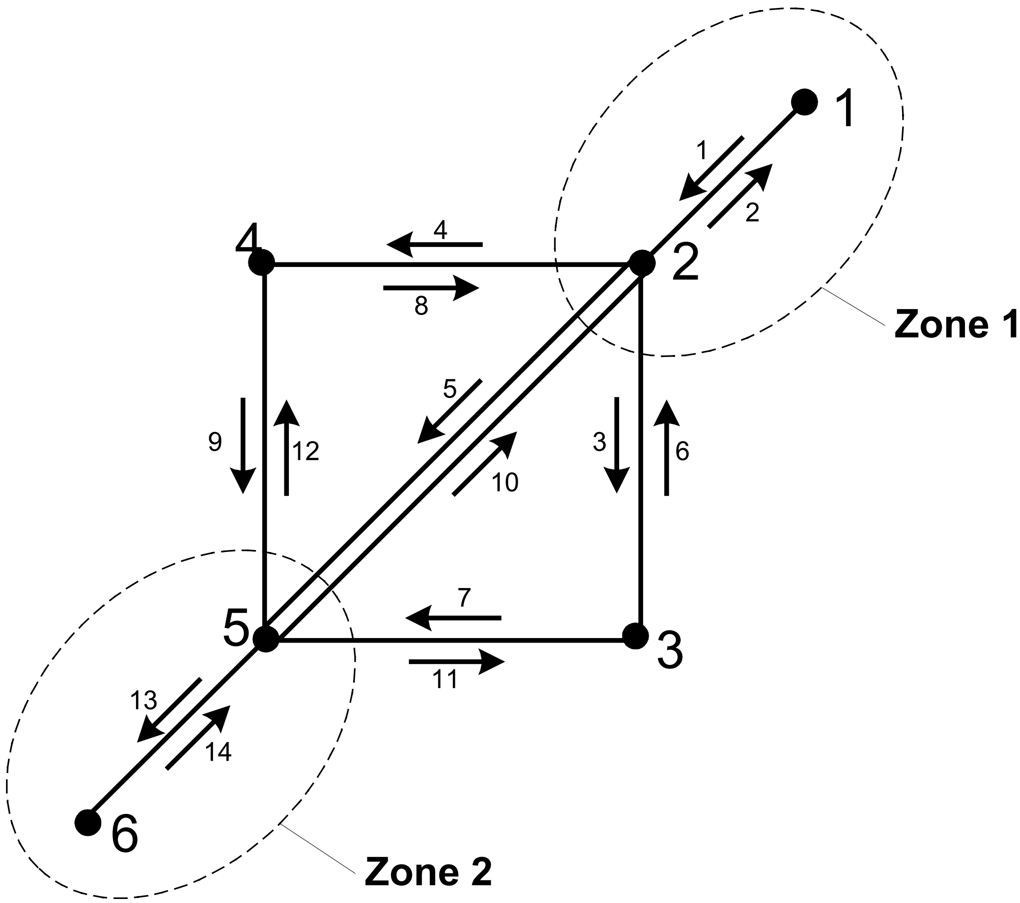


Figure 6.1 Test Network A

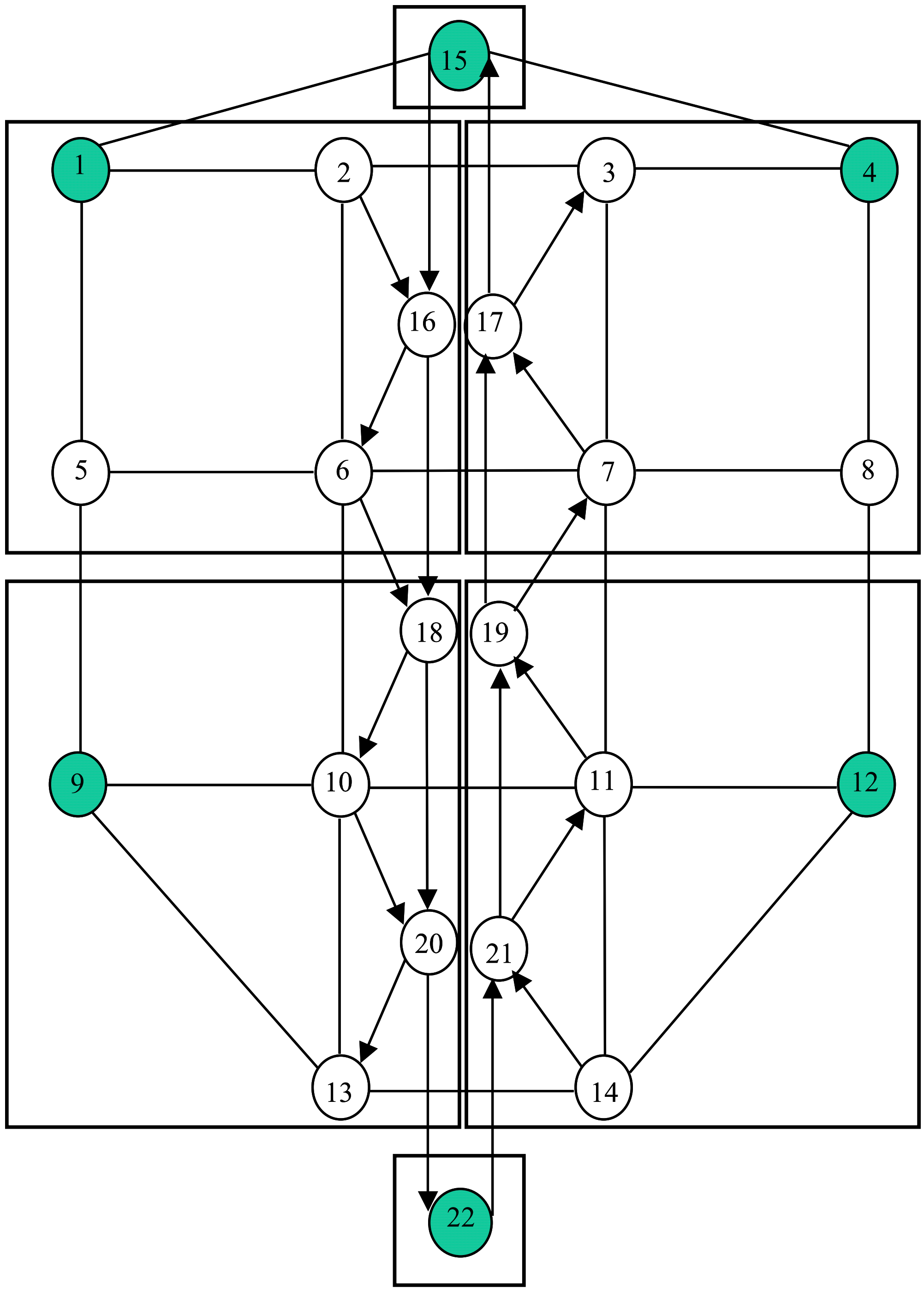


Figure 6.2 Test Network B

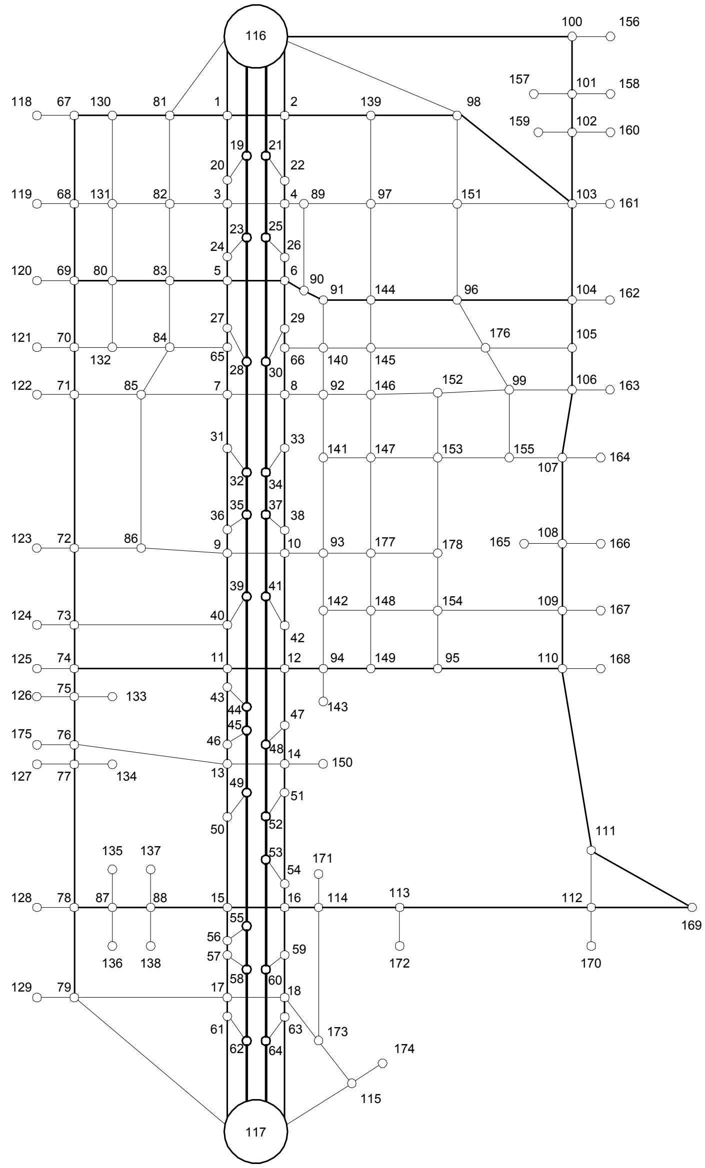


Figure 6.3 Fort Worth Network

6.5. Experiment Design and Numerical Results

To examine the objectives described in Section 6.2, different experiments are designed and conducted. In the following subsections the design of each experiment and the associated numerical results are explained.

6.5.1. Congestion Level

6.5.1.1 Experiment Design

The performance of the bi-level generalized least-squares (Bi-GLS) OD-flow estimation procedure, as explained in Section 3.4, is examined under different OD-flow levels. Network B is used for this experiment and is loaded at different congestion levels referred to as uncongested, congested and over-congested. The uncongested scenario is referred to as the "base case". The numbers of vehicles loaded onto the network and the average speed of vehicles in the network are shown in Table 6.1. The last column of the table shows the percent of vehicles that have not yet reached their destinations by the end of the estimation period. As explained later, it is not viable to consider the origin-destination estimates of these vehicles since their estimator has a high variance.

Table 6.1. Factor levels in congestion-level experiments

Congestion level	No. of vehicles loaded	Avg. speed	% not reached their destination
Uncongested*	10,000 vph	30 mph	21%
Congested	17,500 vph	25 mph	30%
Over-congested	25,000 vph	17 mph	50%

* Base case scenario

The simulated flows on the links resulting from the above loadings are assumed as the ground-truth link-flow observations. However, in all the pertinent experiments, the same initial guess of the demand table is input to the solution process.

The estimation is performed for a 30-minute interval. The sizes of the observation and aggregate departure intervals in these experiments are one and five minutes, respectively. It is assumed that in the real world, users choose their paths according to user-equilibrium assignment. The UE assignment rule is also used in the lower-level optimization for estimating link-flow proportions. In addition, it is assumed that the network is under free flow conditions at the start of the estimation stage. In the estimation of OD flows in consecutive iterations, the method of successive averages, as explained in Section 5.4, is used.

6.5.1.2 <u>Results and discussion</u>

The results are described for each loading level separately.

a) Uncongested Network

This case is considered as the "base case". Figure 6.4 depicts the RMSE of the estimated OD and the simulated link flows when the network is moderately loaded. Referring to this figure, the following points are noteworthy:

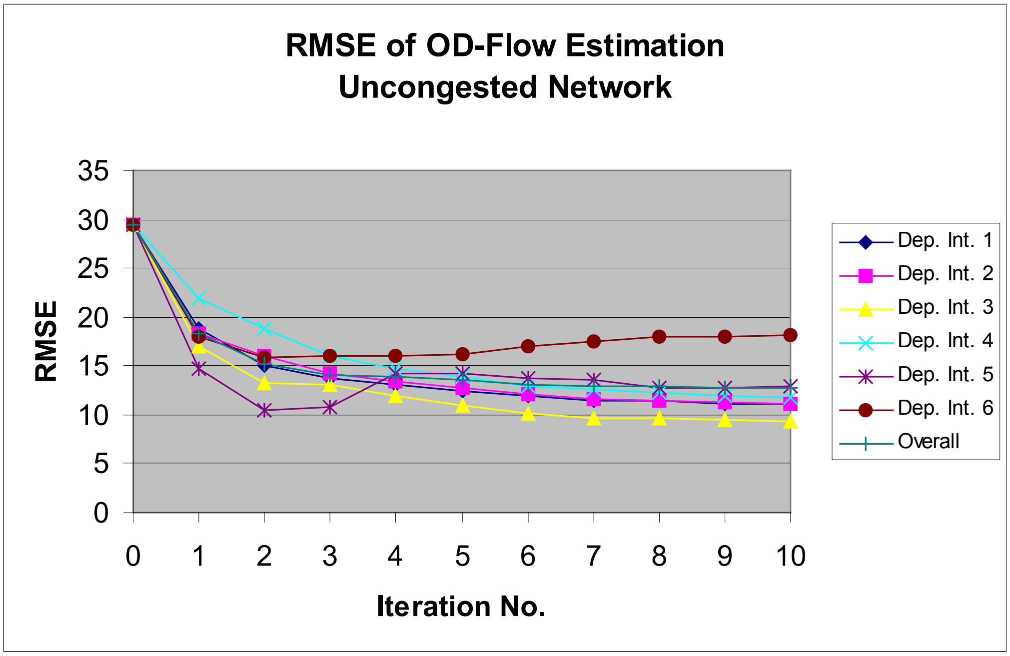
- When the network is uncongested, the algorithm converges in three to four iterations, beyond which only minor reduction in RMSE is achieved.

- Results of the conventional one-level OD-flow estimation, in which link-flow proportions are considered constant, are not optimal. This case corresponds to the first iteration point in Figure 6.4.

- The estimated demands of the departure intervals near the end of the estimation period (departure intervals five and six) are not stable, with the RMSE of the estimated demand increasing in consecutive iterations, as shown in the upper part of Figure 6.4. This behavior occurs because vehicles that have begun their trips toward the end of the estimation stage may not have reached their destination by the end of the estimation stage (21% of loaded vehicles as stated in the above table), therefore the estimation of their origin and destination has a high variance. This is an important aspect in real-time rolling-horizon applications where only demand for the first part of the estimation stage should be considered "final". The length of this period depends on the size and congestion level of the network.

The above observation emphasizes an important point that appears to be overlooked in the estimation of *dynamic* origin-destination flows--regardless of the estimation method used. To obtain a reliable estimate of demand flows, the estimation stage should include several more departure intervals such that the trips initiated during the departure intervals of interest are allowed to be completed. This issue is of particular concern in OD-flow estimation using the Kalman filtering technique, where the estimated OD flow is updated “on the fly” and at the end of each short observation interval. Appropriate safeguards should be introduced to prevent the accumulation of errors and the production of high estimation variance (Kang, 1999).

The estimation for the first departure interval has a lower RMSE, because the network is under free-flow condition at the beginning of the estimation stage; therefore, there are no residual link flows from previous departure intervals to adversely affect the quality of the solution.



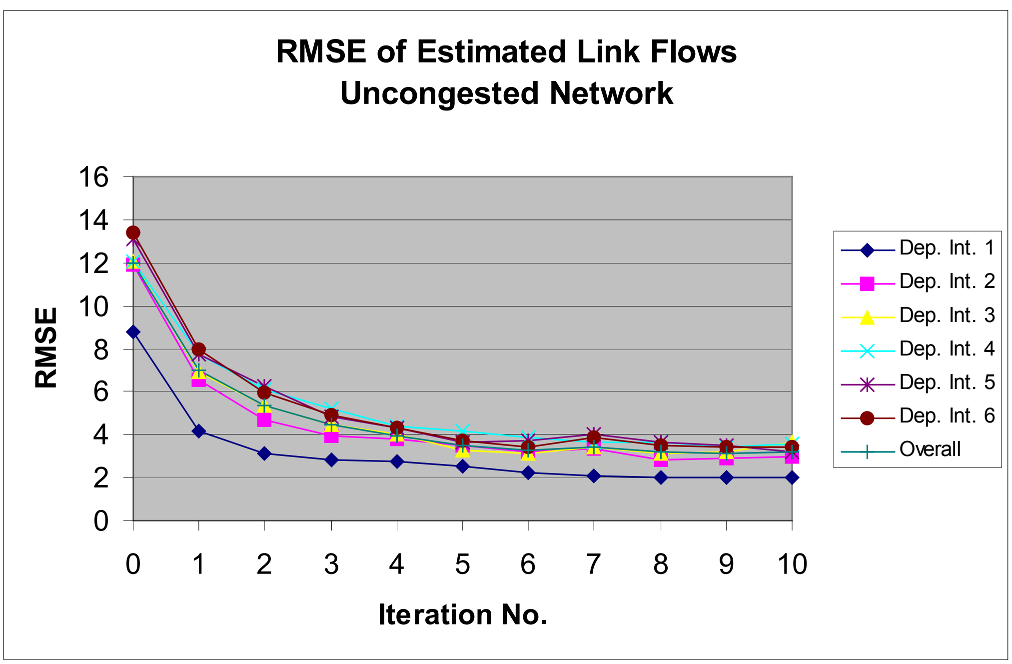
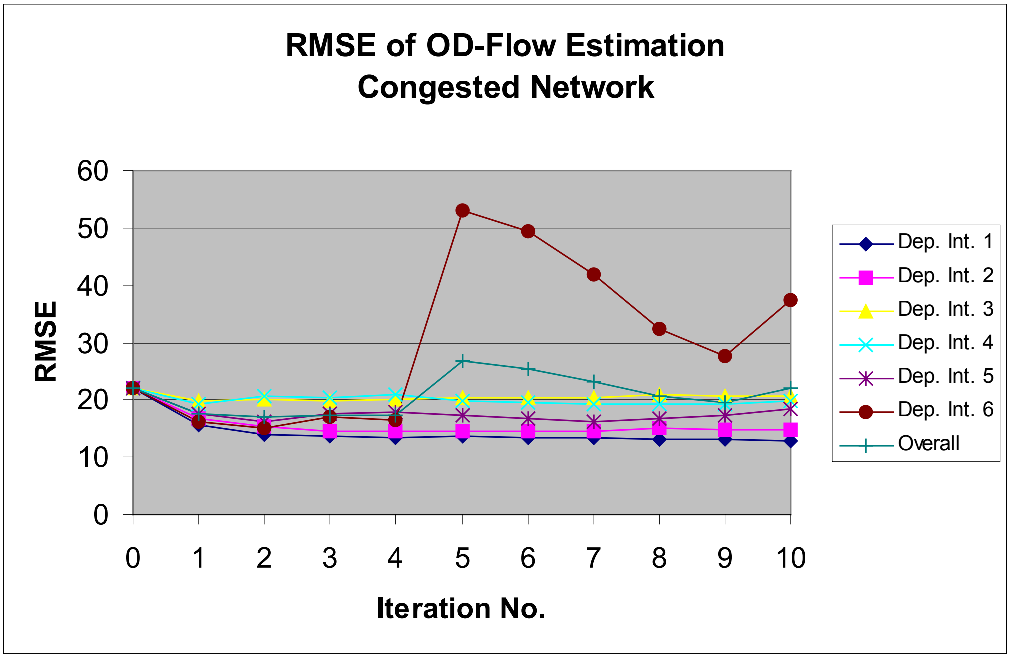


Figure 6.4. Estimation performance—uncongested network

b) Congested Network

Figure 6.5 presents the RMSE of estimation in a congested network. Demand values are increased by 75 percent as compared to the base case. The results are comparable to the uncongested network results, but the instability in the estimation of demand generated toward the end of the estimation stage is more noticeable. In particular, the demand flows that do not reach their destination by the end of the estimation stage may cause identical link-flow proportions, making two or more rows of the link-flow proportion matrix identical and causing singularity in the matrix.



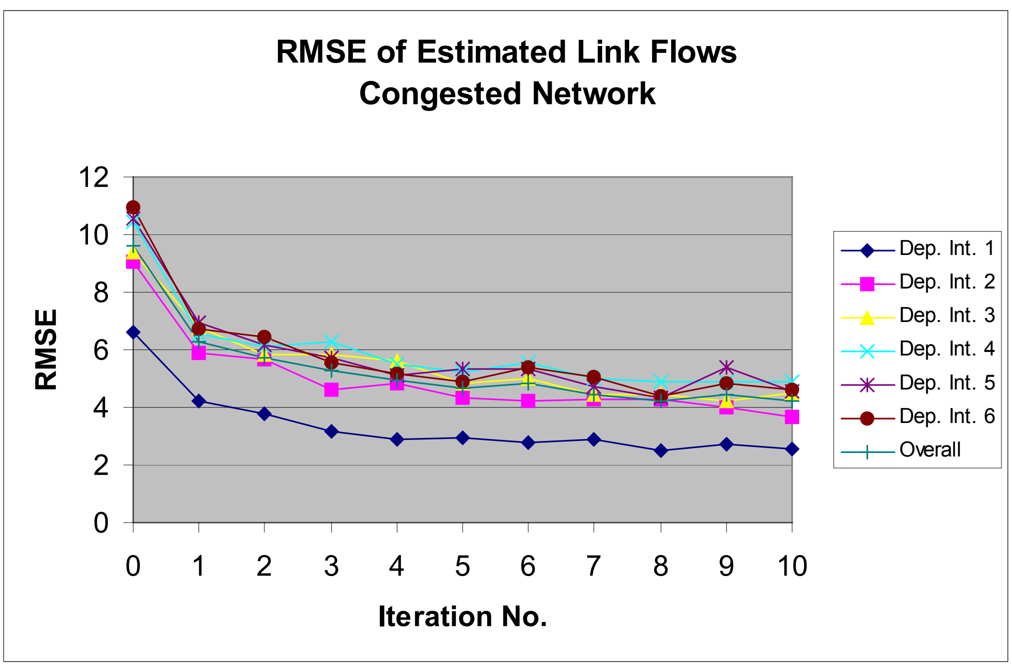
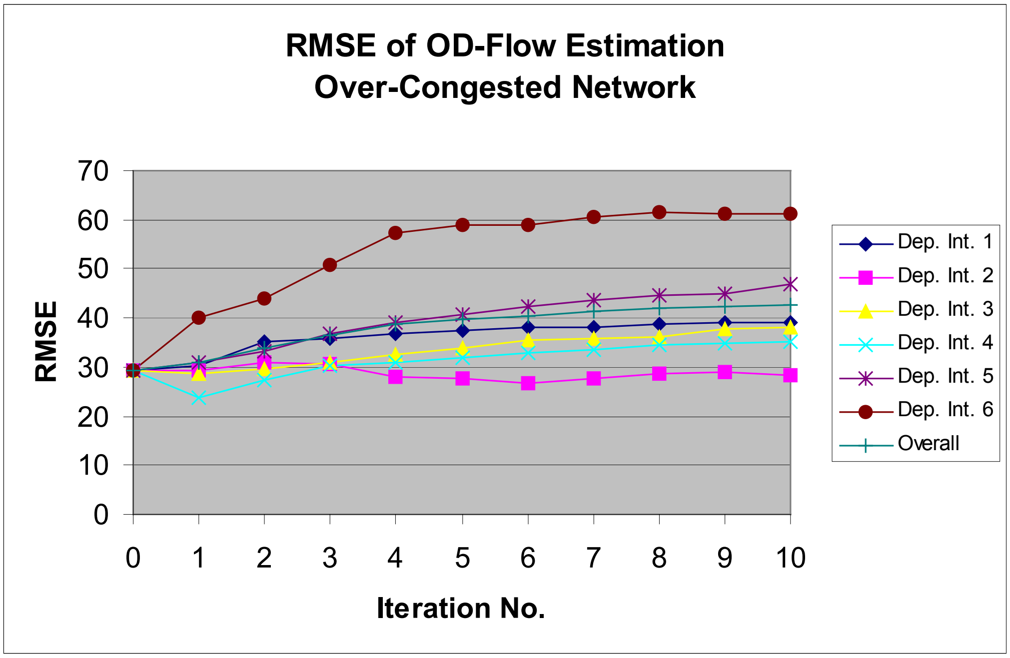


Figure 6.5. Estimation performance—congested network

c) Over-Congested Network

In this case, 2.5 times the demand in the base case is loaded onto the network. As depicted in Figure 6.6, even though the estimation of demand does not converge, the bi-level solution process tends to reduce the errors in the estimated link flow volumes. In this experiment, only 50 percent of the generated vehicles were able to reach their destination by the end of the estimation stage; hence, there is not sufficient evidence to trace the origin and destination of the vehicles in the network. Despite this fact, the procedure increases the consistency of the simulation in terms of flows on the links. As mentioned before, link flows cannot explain the state of the system uniquely (Doan *et al.*, 1999), or in other words, different OD demand flows can produce the same time-varying link flows. This problem is aggravated when the network is over-congested because of insufficient capacity. In this case, low flow volumes on the links are not representative of the actual existing demand. Therefore, in over-congested networks—situations like peak rush hours in downtowns—the use of hybrid models that minimize the deviation of traffic flows and densities on network links may be viable. Furthermore, in these cases, usage of substantially larger aggregate intervals may be justified so that the effect of over-congestion in the network fades out.



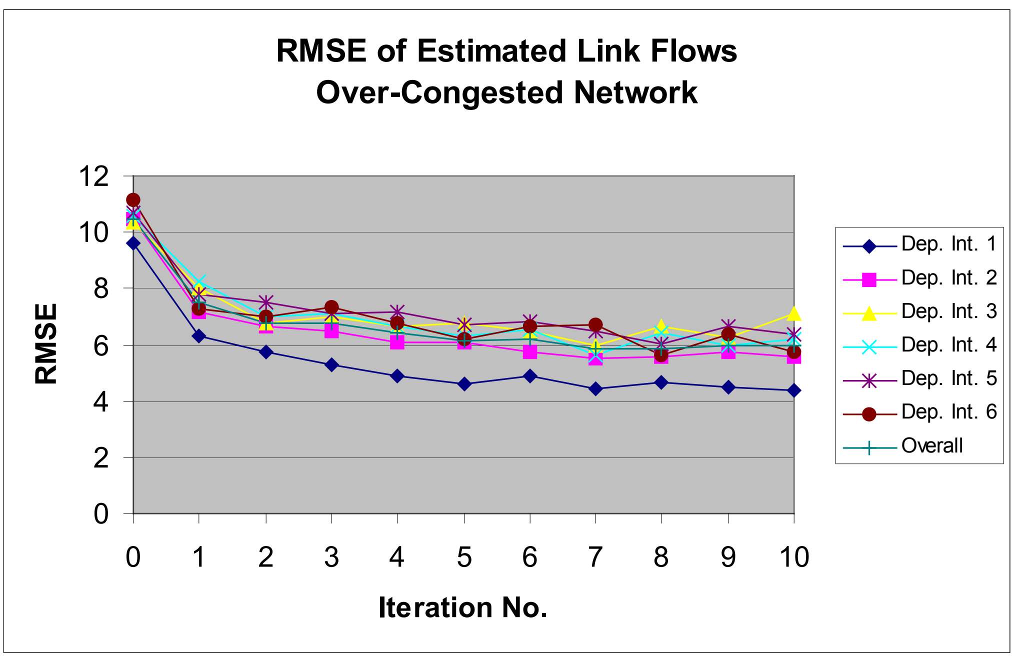


Figure 6.6. Estimation performance—over-congested network

6.5.2. Effect of Inconsistencies in Traffic Assignment Assumptions

6.5.2.1 Experiment Design

To investigate the effect of inconsistent assumptions on the trip-maker's route-choice, the experimental factors shown in Table 6.2 are considered. The experiments are conducted assuming different route-choice behavior in obtaining the ground-truth results. To examine the interaction between assignment assumptions and congestion level on the quality of the solution, some of the experiments are run at two different congestion levels. In all scenarios, user equilibrium assignment is used to estimate the link-flow proportions in the lower-level optimization problem.

Table 6.2. Factor levels considered in route-choice experiments

Scenario no.	**Congestion level**	**Real-world route-choice behavior**	**Assumed assignment rule**
1a*	Uncongested	User Equilibrium	User Equilibrium
1b	Congested		
2a	Uncongested	System Optimal	User Equilibrium
2b	Congested		
3	Uncongested	50% SO-50% UE	User Equilibrium
4	Congested	Imperfect UE	User Equilibrium

* Base case

In case number 1, it is assumed that users in the real world follow the user equilibrium rule for their route choice. This assumption in the conducted experiments is enforced by running the DTA simulation program by assuming UE assignment in obtaining the ground-truth link flow observations and in OD-flow estimation runs. This case is run under two assumptions regarding the congestion level, uncongested and congested as defined in Section 6.2. Scenario 1a constitutes the base scenario.

Case number 2 assumes users in the real world are guided based on the system-optimal assignment rule; however, in the estimation of OD-flows, a user equilibrium assignment is assumed. Based on the congestion level, this case is also divided into two separate scenarios.

In the third case, it is assumed that not all of the users in their route choice follow the same rules, i.e. half of the users pursue the UE assignment and the other half follow the SO assignment rules. This case is only examined under uncongested conditions.

In the imperfect UE assignment, it is assumed that users tend to choose their routes based on their shortest paths, but, say because users don't have complete information, the system is in partial equilibrium. Since this case in the real world is more likely to happen under the congested condition, the test is conducted only under congested conditions.

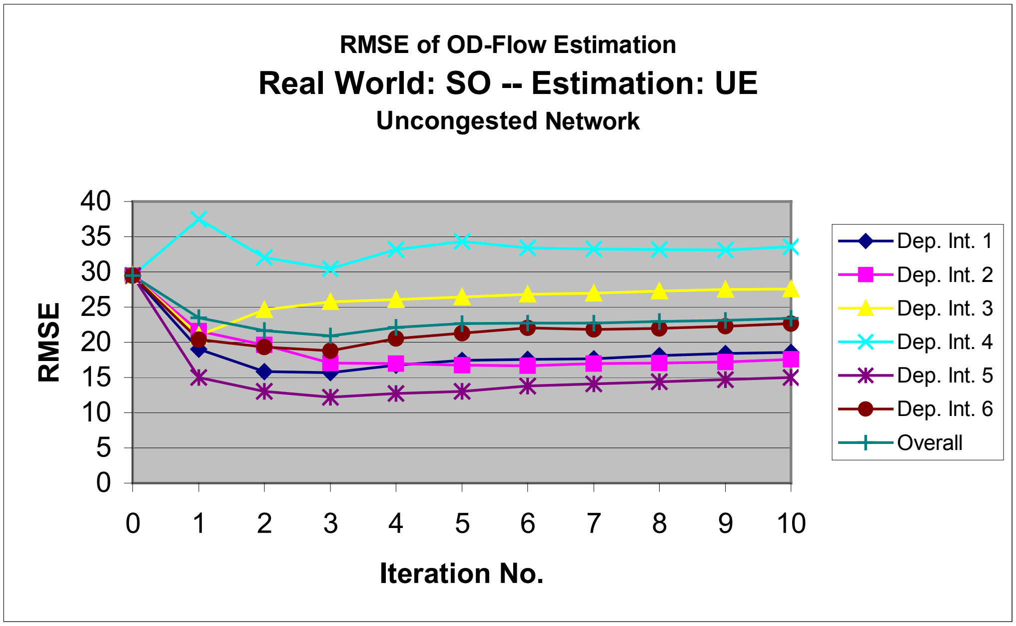
These experiments are performed on Network B, assuming one-minute observation intervals and five-minute aggregate departure intervals.

6.5.2.2 Results and discussion

Scenarios 1a and 1b of the first case are implemented and discussed in Section 6.5.1 (Figures 6.4 and 6.5) and will not be repeated here.

In the second case, it is assumed that users in the real world follow SO assignment rules, but UE assignment is used in OD-flow estimation. Figure 6.7 depicts the results of the uncongested network. The iterative process reduces errors in link flow estimation. The OD-flow estimation also converges, but the final residual is higher than in the case where assignment assumptions were consistent, as shown in the lower part of Figure 6.4. This observation confirms that when the load on the network is not very high, UE and SO solutions are not significantly different (Peeta and Mahmassani, 1995b).

In congested networks, the procedure reduces the errors in the estimated link flows, but OD-flow estimation does not converge (Figure 6.8). As mentioned before, the procedure estimates the OD-flows by minimizing the inconsistencies in estimation of link flows. Consequently, the algorithm minimizes the *combined* errors due to loading the "estimated" demand, traffic assignment assumptions, control strategies, etc. If any of the contributing factors is not consistent with the real world, the accuracy of the estimation degrades. It should be emphasized that this problem is not specific to the algorithm presented nor is it due to the usage of the simulation program to find the link-flow proportions. In the static case or in the one-level estimation of dynamic demand flows, assumptions made on link cost functions and traffic assignment often cause similar inconsistencies.



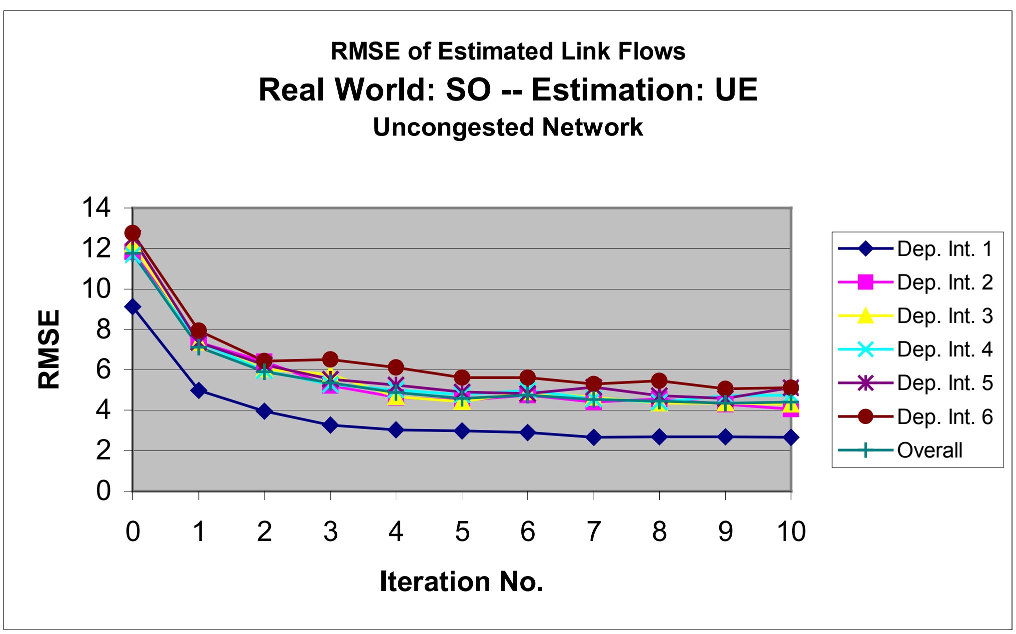
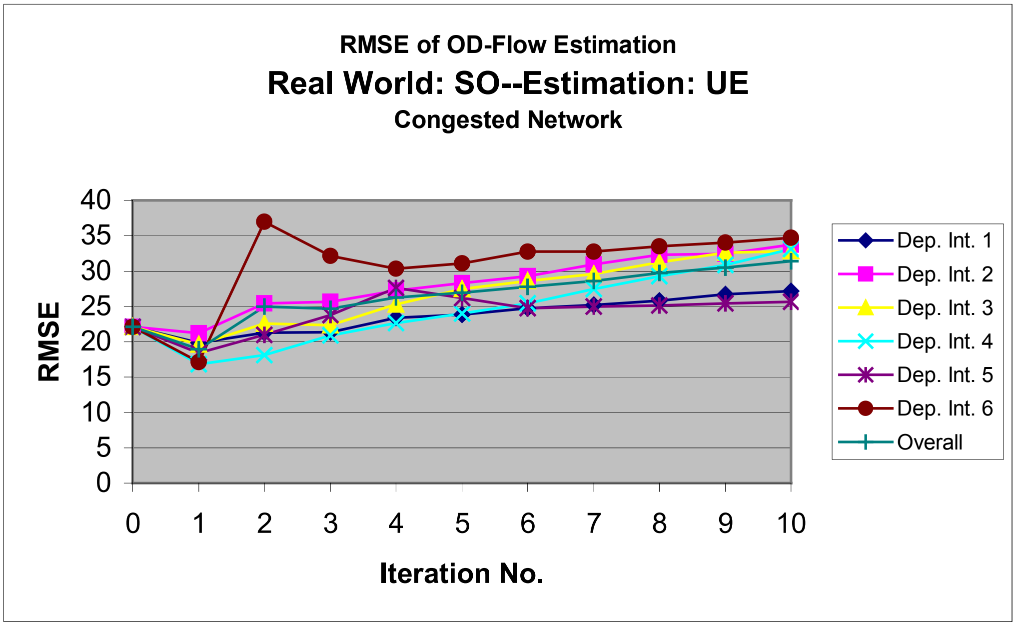


Figure 6.7. Estimation performance—uncongested network, inconsistent assignment assumptions, SO assignment for the real world and UE assignment for OD-flow estimation



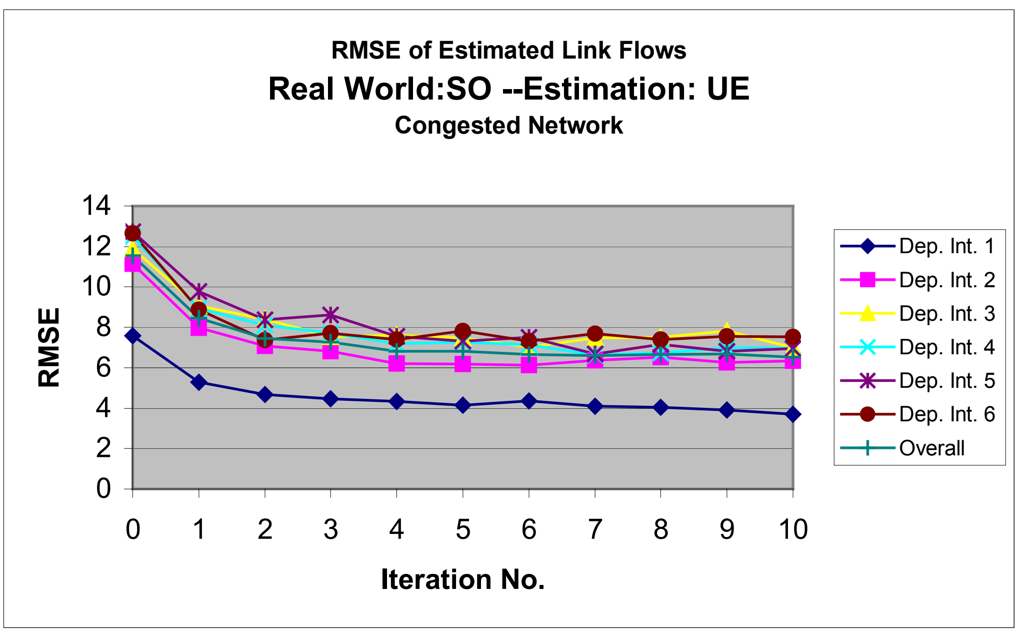


Figure 6.8. Estimation performance—congested network, inconsistent assignment assumptions, SO assignment for the real world and UE assignment for OD-flow estimation

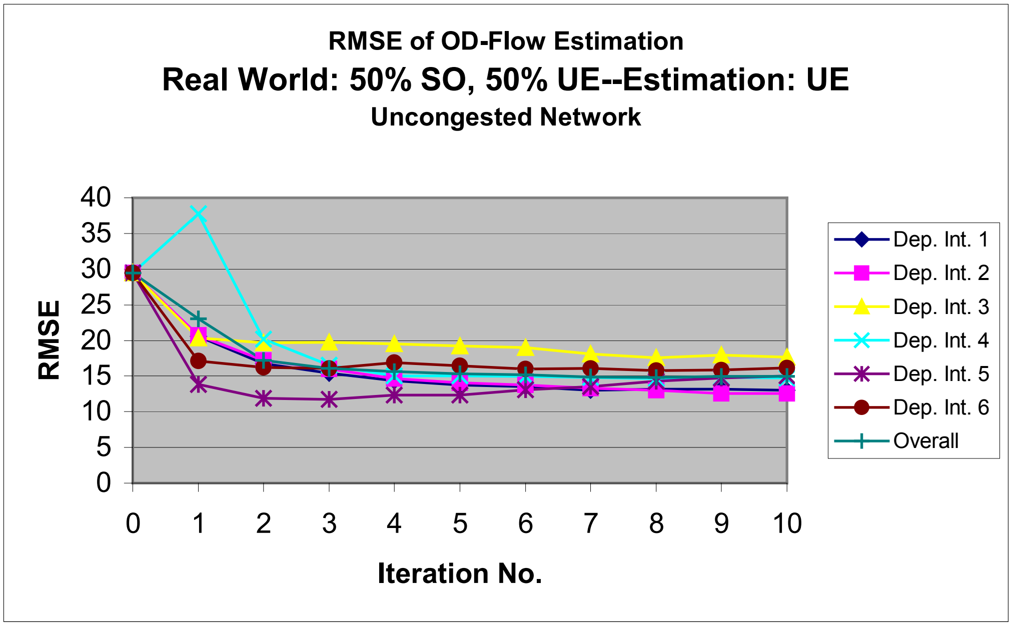
In the third case, a mix of half SO and half UE assignment is assumed for users' route choice behavior in a real-world, uncongested network. The results are illustrated in Figure 6.9. The solutions of both OD flows and link flows converge. The errors, as expected, are something between consistent assignment assumptions, case 1, and case 2 where users were following complete SO assignment rule.

However, in the real world, at least until when a mature traveler route guidance system is not in place, users are not aware of and do not follow an SO assignment solution. But based on their knowledge or perception, they tend to take the perceived shortest paths. This behavior implies that the route-choice behavior of tripmakers would be close to the user equilibrium traffic assignment. However, due to a lack of complete information, tripmakers may not exactly follow the shortest paths as in the results of the simulation. This discussion leads to the fourth case in this set of experiments.

As explained, to implement the fourth case, the DTA simulation program is iterated three times and stopped before reaching the required assignment convergence. While in the OD-flow estimation, the simulator is iterated five times to result in a more stable solution. This scenario is applied to congested loading conditions as stated in Section 6.5.1.

As illustrated in Figure 6.10, the algorithm in terms of link flows exhibits favorable performance and reduces the errors significantly within five iterations. In terms of the OD-flows, the algorithm reduces the error as well, especially in the first and the second departure intervals. The final RMSE is comparable to the third case scenario. The results again emphasize the effect of consistent assignment assumptions on the quality of the OD-flow estimation.

Nevertheless, it should be noted that if a reliable *a priori* OD-flow table is available, one can improve the quality of the OD-flow estimation. This issue is discussed when examining the application of the Bayesian inference method.



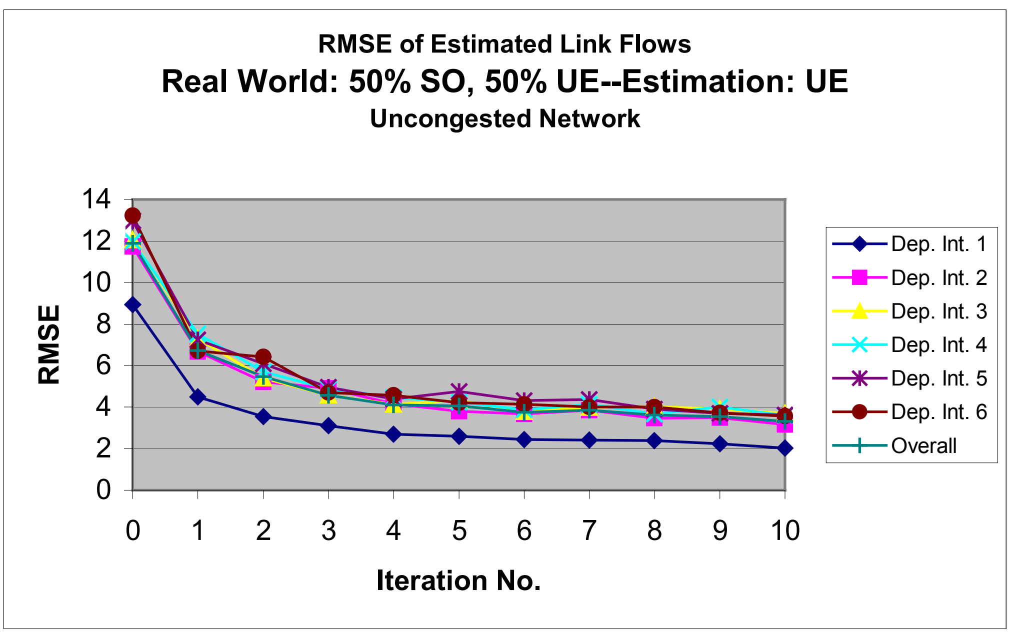
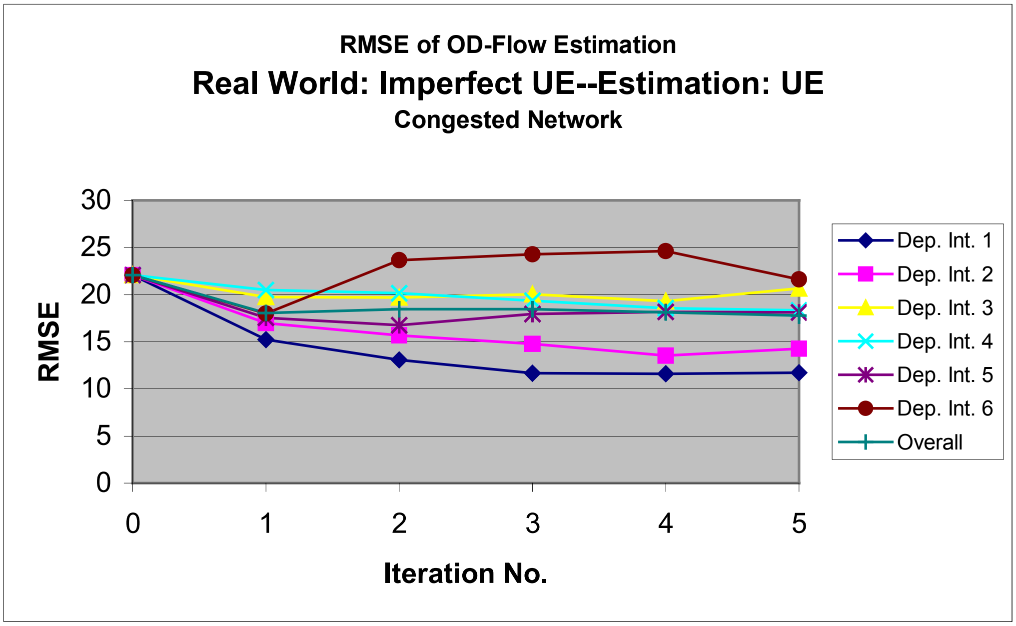


Figure 6.9. Estimation performance—uncongested network, inconsistent assignment assumptions, 50% SO-50%UE assignment for the real world and UE assignment for OD-flow estimation



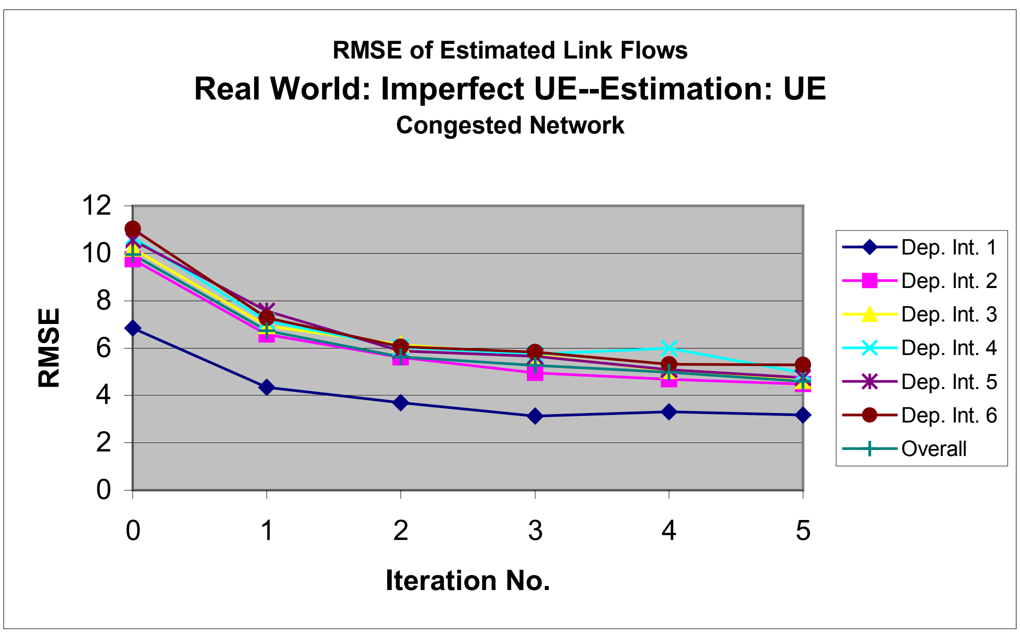


Figure 6.10. Estimation performance—congested network, inconsistent assignment assumptions, imperfect UE assignment for the real world and UE assignment for OD-flow estimation

6.5.3. Effect of Size of Aggregate Departure Interval

6.5.3.1 Experiment Design

In these experiments the observation interval is fixed at one minute, but aggregate departure intervals are chosen between one and fifteen minutes. The chosen aggregate departure intervals for this experiment are as follows:

- 2 minutes
- 5 minutes
- 8 minutes
- 10 minutes, and
- 15 minutes

The tests are conducted on Network B under uncongested loading.

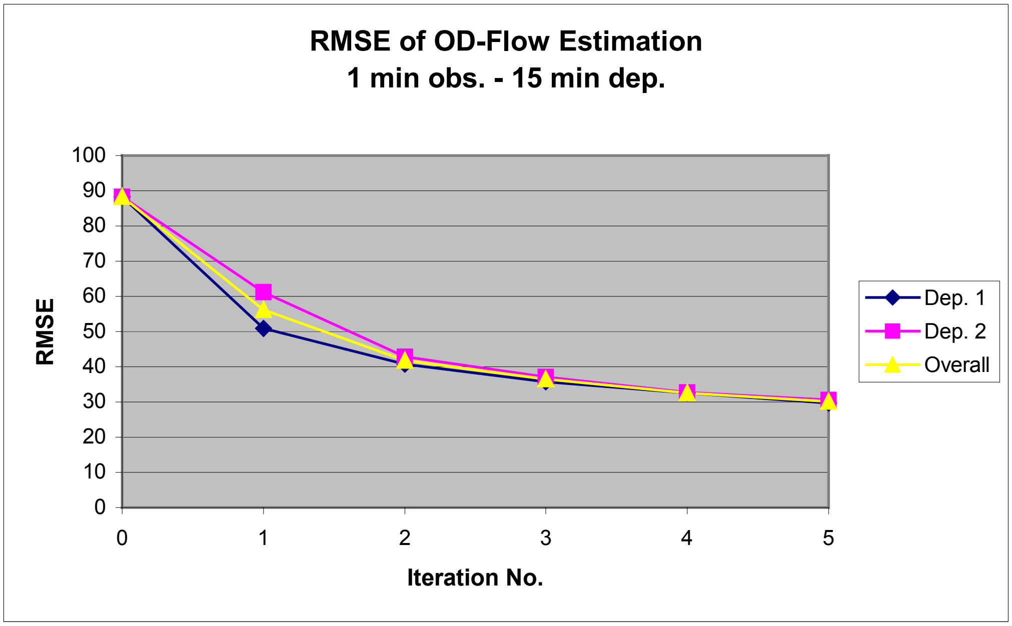
6.5.3.2 Results and discussion

The estimation results for five-minute departure intervals, the base scenario, are already shown in Figure 6.4. The results of the experiments when the aggregate departure interval is fifteen minutes are illustrated in Figure 6.11, and Figure 6.12 depicts the summary of the results. To compare the solutions of different runs, the RMSEs of the OD-flow estimation, shown in Figure 6.12, are divided by the length of the aggregate departure intervals in minutes.

In general, choosing smaller aggregate departure intervals increases the complexity of the problem in terms of the size of the matrices, particularly the link-flow proportion matrix, and increases the number of unknowns and reduces the over-specification of the problem. On the other hand, if the departure intervals are too large, the problem becomes more or less like the static case.

Furthermore, If the size of the departure interval is too small, the number of vehicles departing at each interval becomes too small and the error in terms of the percentage of demand values increases. Short departure intervals will also make the effect of other sources of errors such as traffic signals and flow propagation more significant. In addition, in this case since only a small number of vehicles can depart in each departure interval, the likelihood of obtaining a singular link-flow proportion matrix increases. The small number of vehicles generated in a departure interval may not contribute to the flows on any link, especially when the network is congested, and may cause degeneration in the solution.

Figure 6.12 illustrates that the optimal aggregation size for these experiments is eight minutes. Obviously, this result is not conclusive and the optimal aggregation size depends on the size of the network, the number of unknown OD-flows and congestion levels in the network, among others.



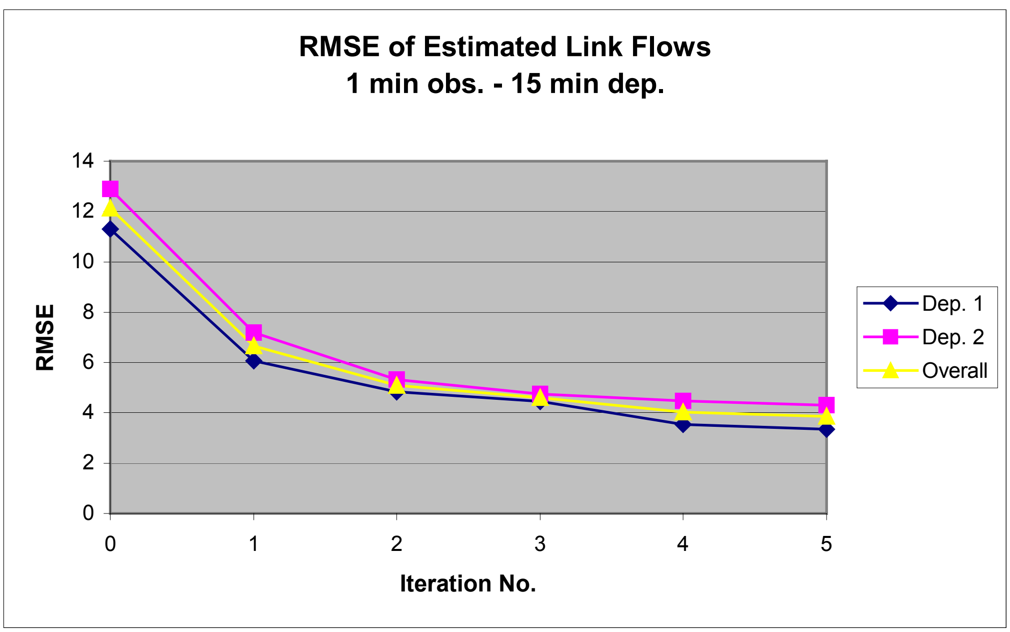
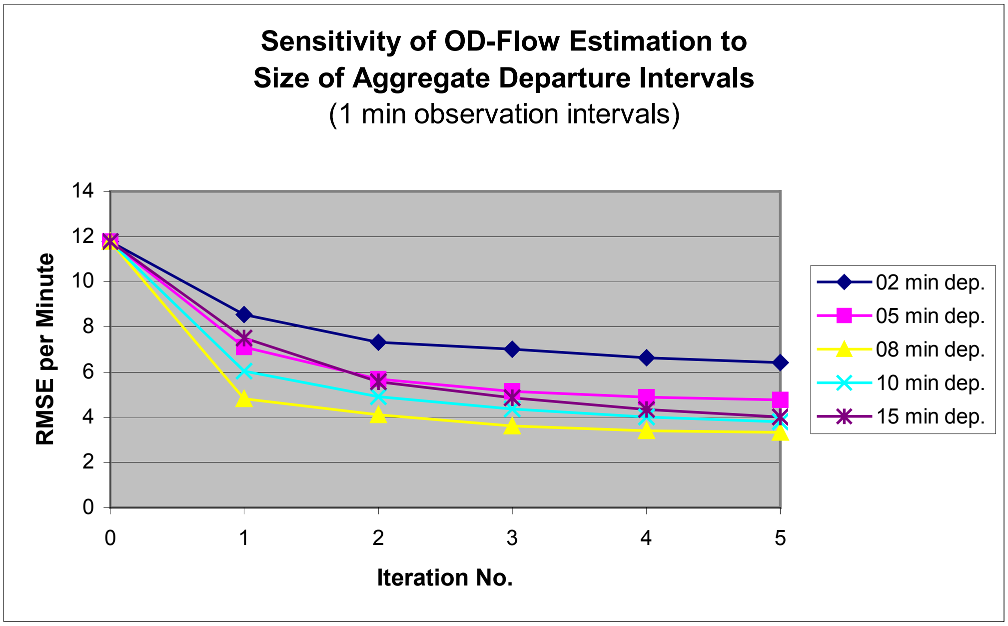


Figure 6.11. Sensitivity of estimation to departure-interval aggregation size; observation interval: 1 min., departure interval: 2 min.



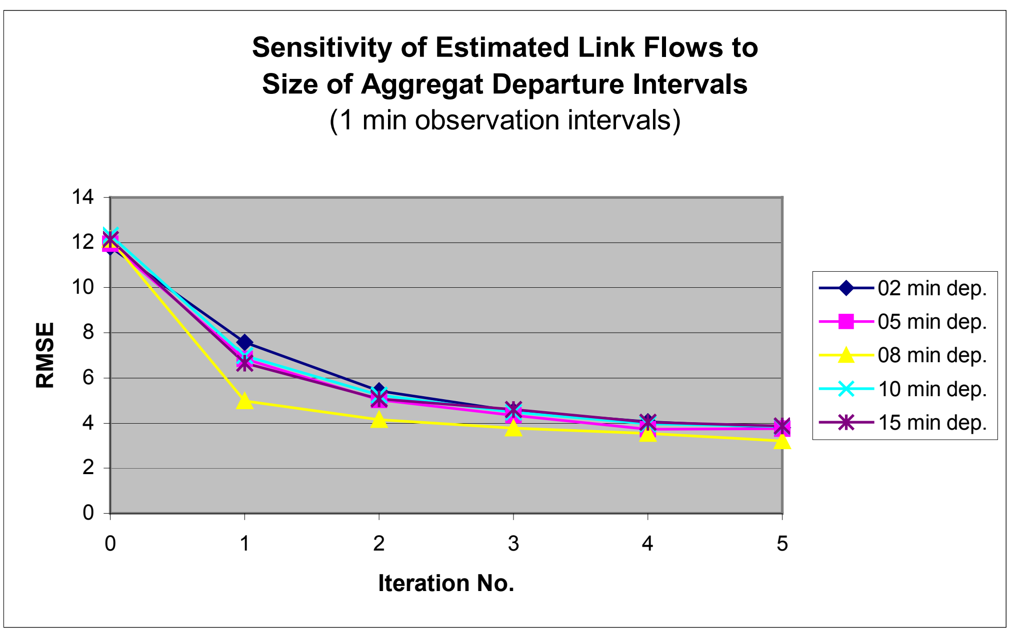


Figure 6.12. Sensitivity of OD-flow estimation to departure-interval aggregation size—summary

6.5.4. Effect of Observation Interval Size

To investigate the effect of observation interval size on the performance of OD-flow estimation and the external consistency of the simulation program, a set of experiments are conducted. In these experiments, the size of the observation intervals is varied between one and ten minutes, that is

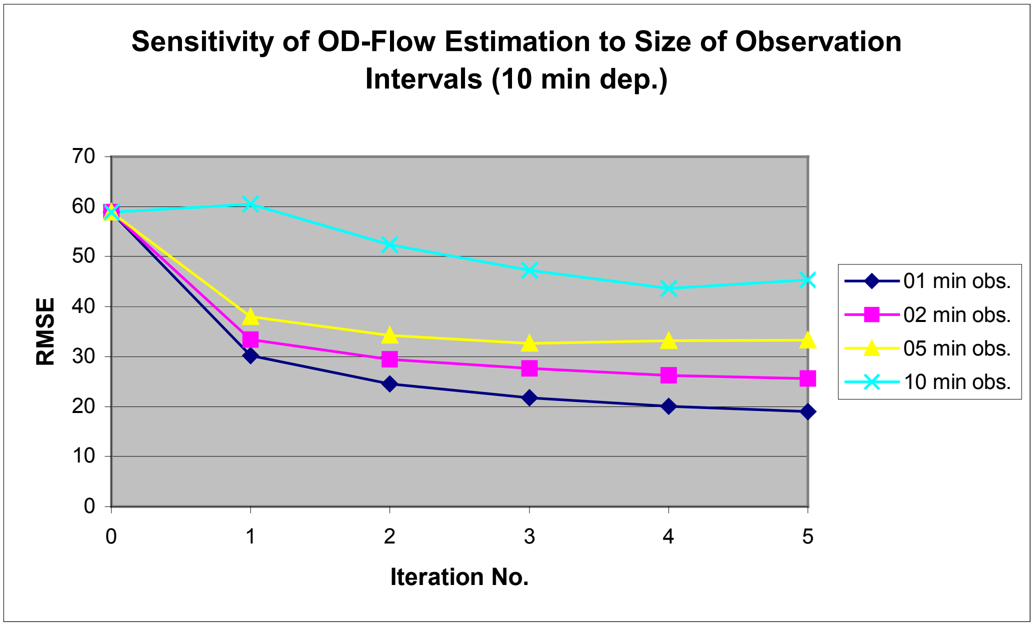
- one minute
- two minutes
- five minutes, and
- ten minutes.

In all runs, the departure interval was fixed at ten minutes. The experiments are performed on Network B under an uncongested condition.

6.5.4.1 Results and Discussion

The estimation results for all departure intervals in the estimation period are summarized and shown in Figure 6.13. The RMSEs of the estimated link-flows are divided by the observation interval size in order to make the results comparable.

Interestingly, the size of observation intervals does not have a significant effect on the accuracy of the estimated link flows, as shown in the lower part of Figure 6.13. However, as the ratio of sizes of the aggregate departure intervals to observation intervals increases, the error in the estimation of OD flows decreases. As mentioned before, this improvement is due to an increase in the degree of over-specification in the problem. The jump in the RMSE of the OD-flow estimation, when the observation interval is equal to the departure interval of ten minutes, is attributed to the singularity in the link-flow proportion matrix, and consequently, degeneracy of the estimation process.



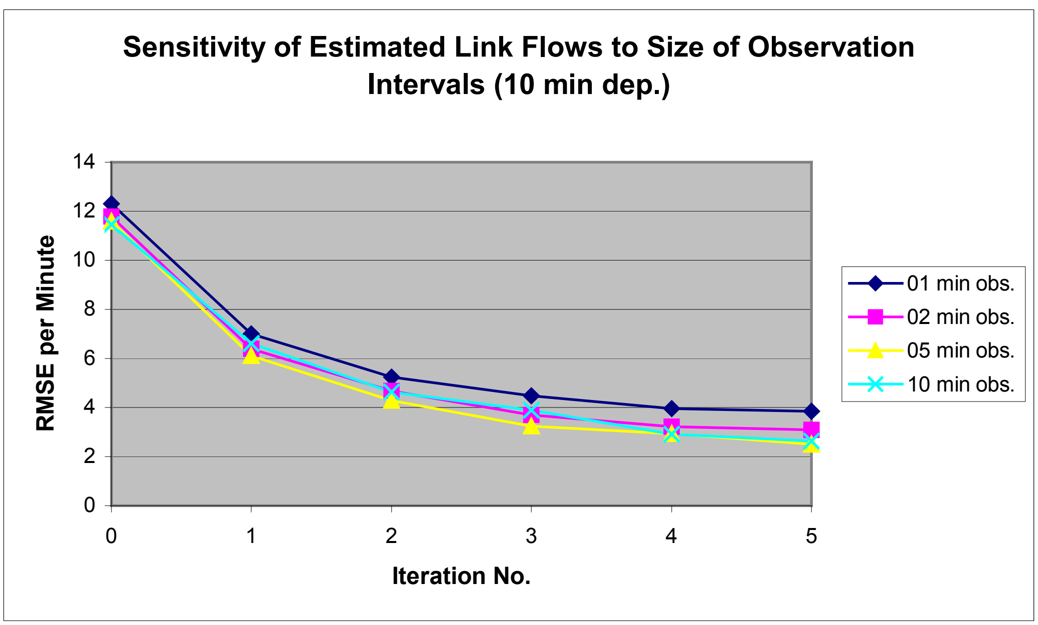


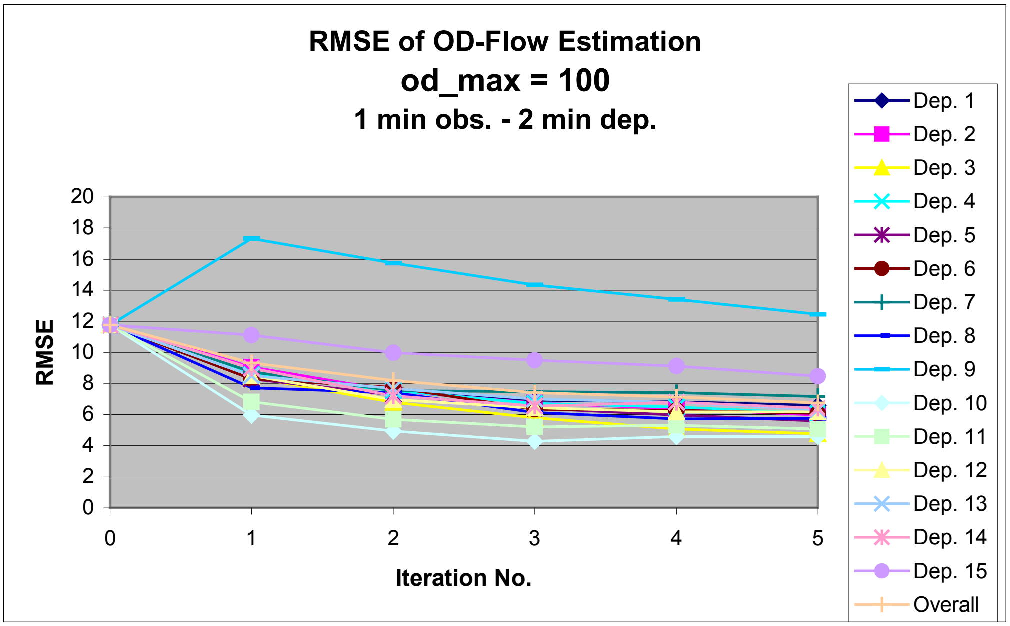
Figure 6.13. Sensitivity of estimation performance to the size of observation intervals

6.5.5. Effect of Upper Limit on OD-Flow Estimation

6.5.5.1 Experiment Design

The upper limit value (the parameter od_max in the implementation, as defined in Section A.2 in the appendices) can be chosen subjectively based on the maximum possible or probable vehicular demand flows generated from each origin to each destination during a departure interval. This parameter can expedite the convergence of the procedure and prevent assigning unacceptable values to OD flows especially when the estimation process degenerates. The sensitivity of OD-flow estimation to the value of od_max is tested by setting it to 50 and 100 in test Network B when the network is uncongested. The observation intervals are one minute and aggregate departure intervals are set to two minutes. The lower limit is set to one and is not included as a factor in the experiments.

As illustrated in Figures 6.14 and 6.15, setting a lower value for od_max results in a smoother estimate of OD-flows and faster convergence of the algorithm. However, the change of od_max, if not set to very small values, does not affect the quality of the solution in terms of the link-flow estimates (external consistency).



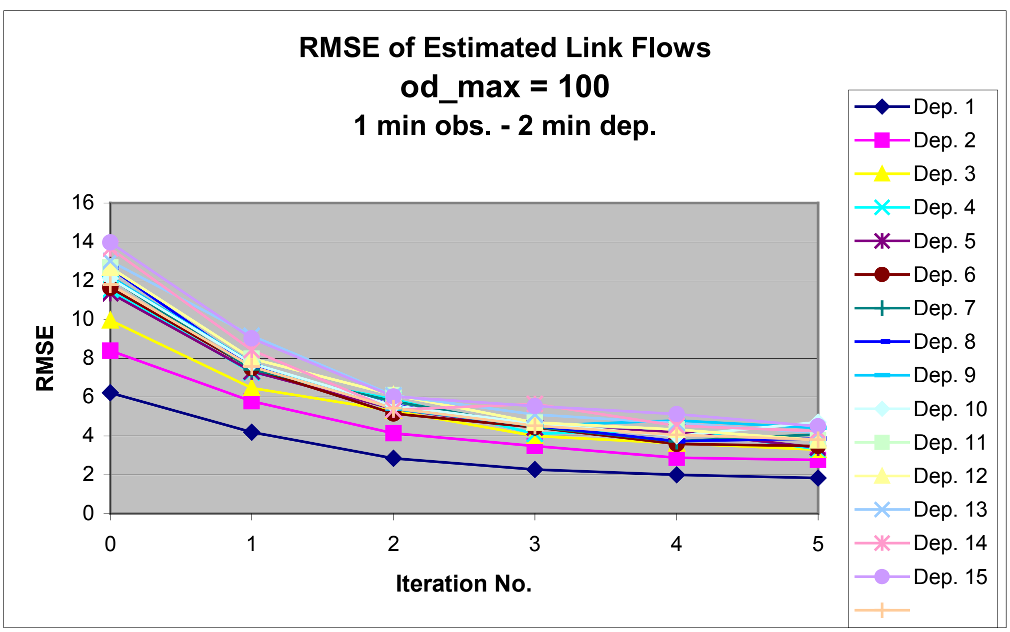
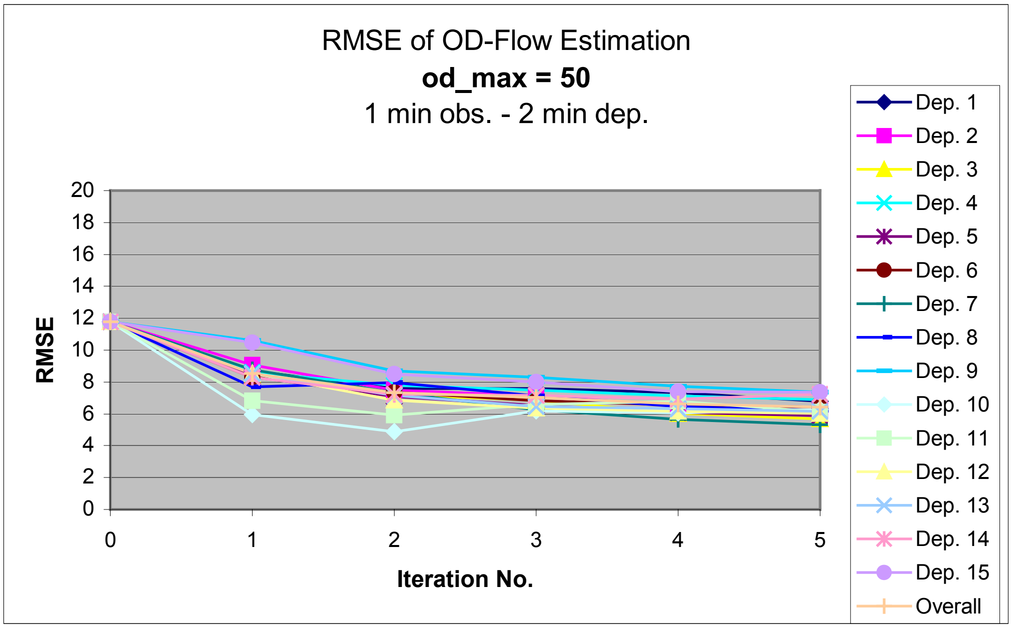


Figure 6.14. Sensitivity of OD-flow estimation to the maximum allowable demand value—od_max=100



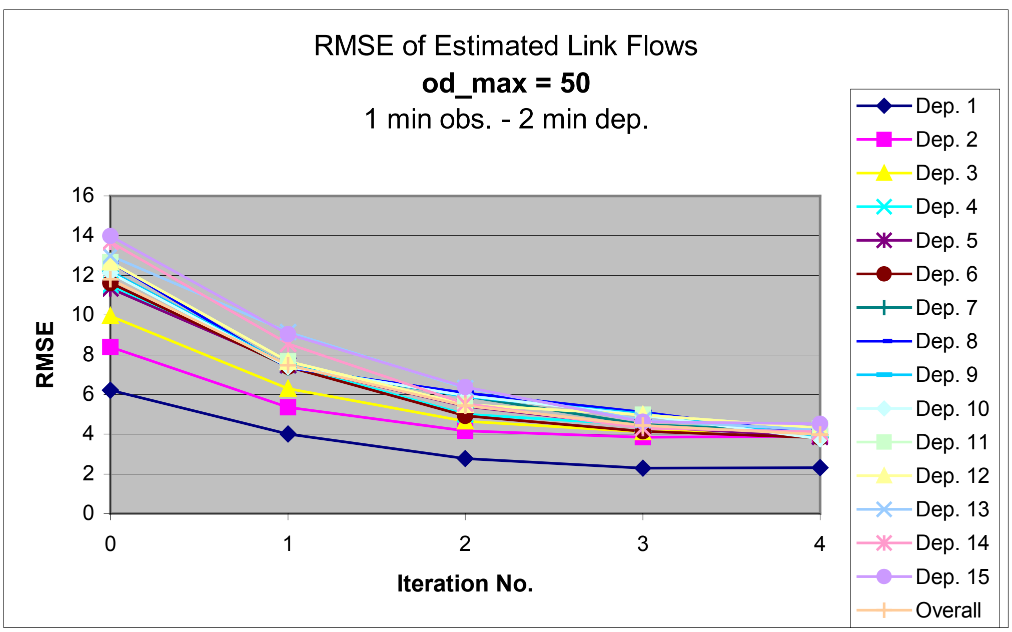


Figure 6.15. Sensitivity of OD-flow estimation to the maximum allowable demand value—od_max = 50

6.5.6. OD-Flow Estimation in Large Networks

6.5.6.1 Experiment Design

To investigate whether the proposed bi-level GLS OD-flow estimation method can be implemented in large networks, two experiments are conducted on the FW Network. The assumed actual demand flow pertains to a peak morning scenario. The estimation period consists of three departure intervals of ten minutes. Observation intervals are set to one minute. The network is loaded with about 19,000 vehicles per hour. The average travel speed over the entire network is about 36 miles per hour and the stopping delay time is about five percent of the average travel time.

The solution quality in terms of OD-flow estimation and improvement in external consistency is tested by assuming two different initial OD-flow tables as input to the estimation algorithm.

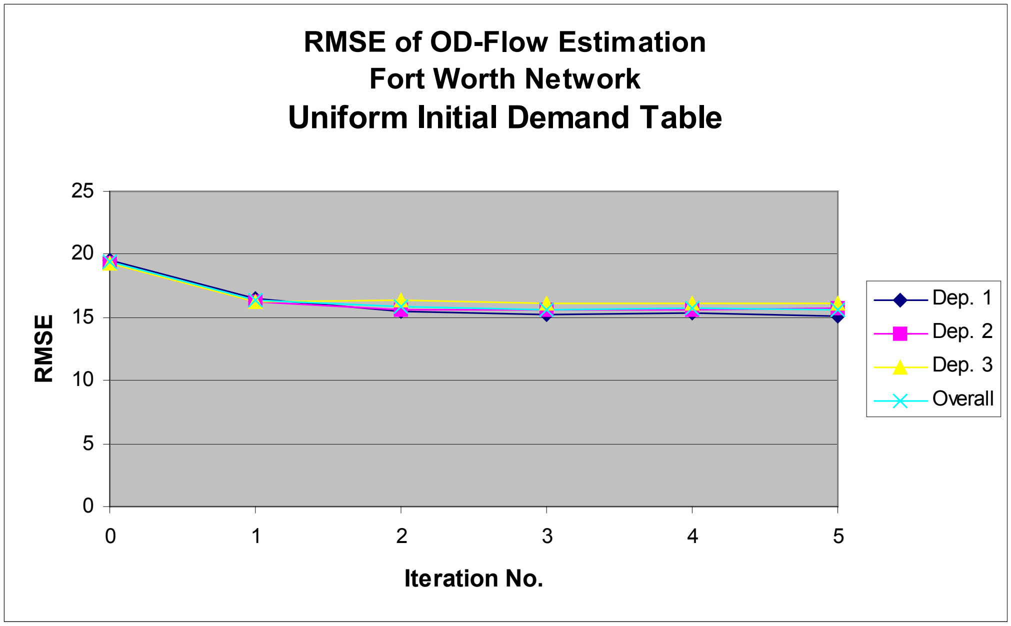
The first initial OD-flow table assumes there is no information on the OD-flow pattern in the network, hence it assumes equal values for all OD flows in all departure intervals. In the second experiment, it is assumed that information on general OD-flow pattern in the network exists. The assumed actual OD flow values are halved and are input as the initial guess of OD flows to the Bi-GLS estimation program. The improvement in OD-flow estimates and consistency checking can be compared to the base case of Section 6.5.1.

6.5.6.2 Results and Discussion

Figure 6.16 depicts the results of the first experiment where a uniform OD-flow table was assumed as the starting point of the estimation process. As it is seen, the RMSE of OD-flow estimation improves rather significantly in the first iteration but subsequently the improvements are not as significant (about one vehicle per departure interval or five percent). However, the improvement in the RMSE of the estimated link flows is more significant in consecutive iterations.

The results of the second experiment are illustrated in Figure 6.17. In this case the initial demand is assumed to be 50% of the actual values. The first iteration of the estimation process has improved the RMSE of OD flows by about 20 percent, but in the second iteration, no improvement is obtained and even slight divergence is observed. However, with regard to the external consistency of the system in terms of the observed link flows, an improvement of 40 percent is witnessed in just two iterations.

The behavior of the estimation method observed in both cases, as mentioned in detail in Chapter 3, occurs because the objective in the upper-level optimization problem is to minimize errors in the estimated *link flows*. If there existed an *a priori* OD-flow table, the estimated OD-flows would further improve by the use of the Bayesian inference method, as explained in Section 6.5.7.



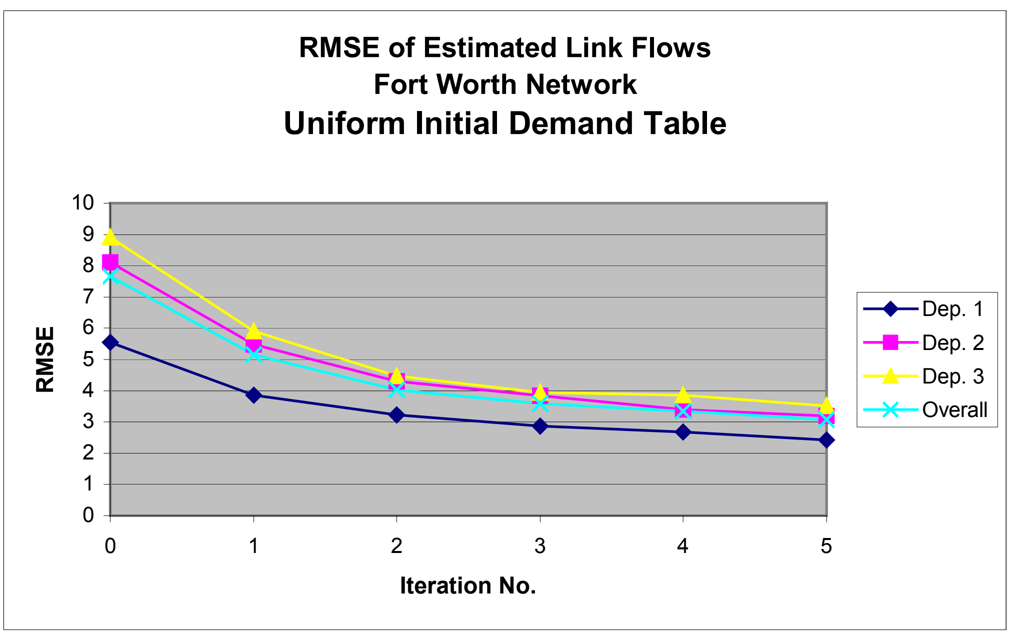
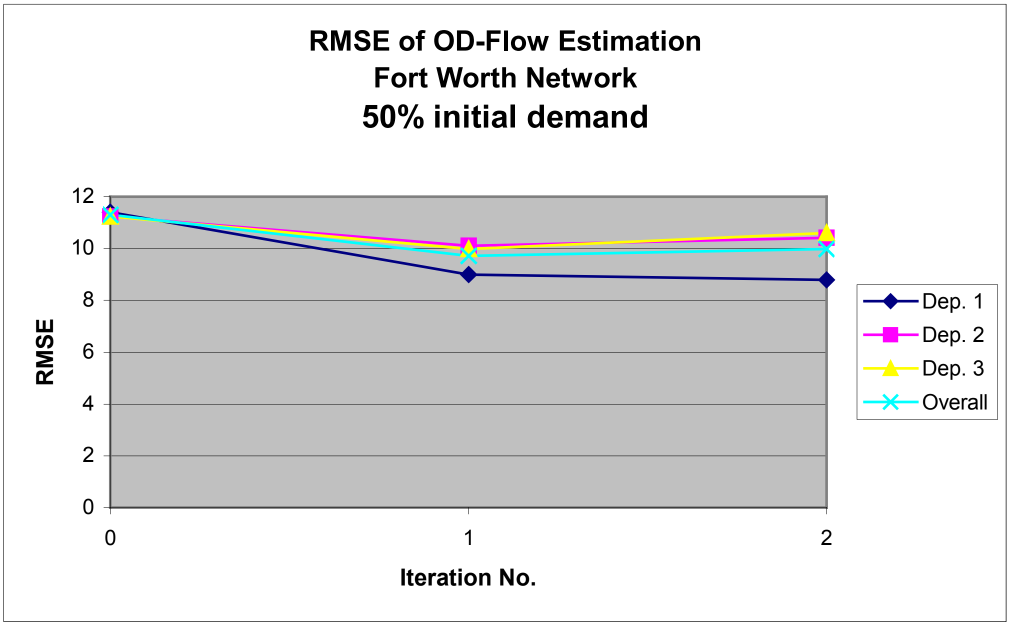


Figure 6.16. Bi-level GLS OD-flow estimation for FW network; uniform initial demand



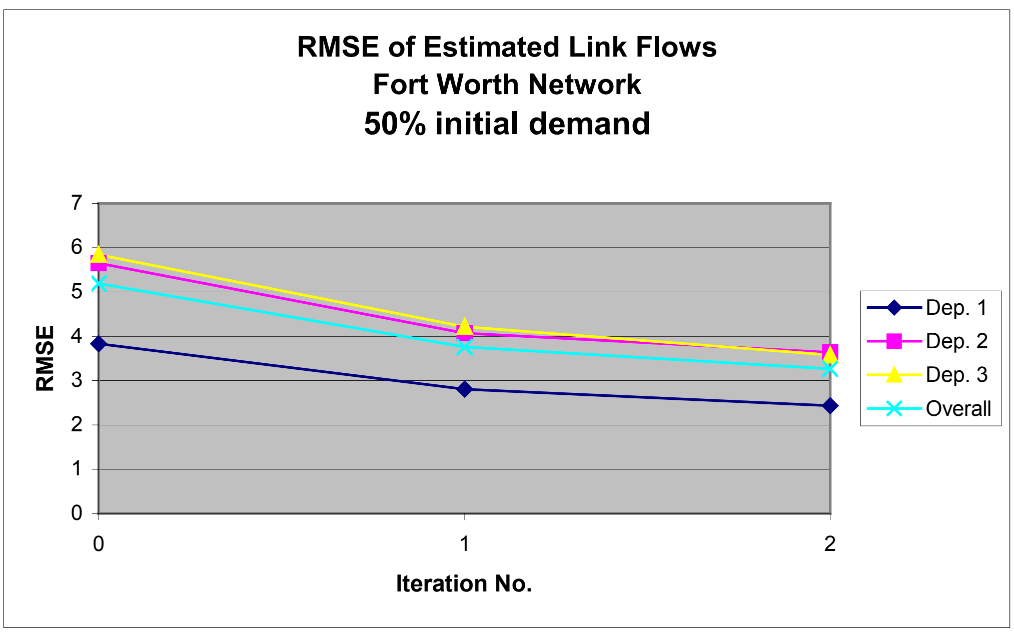


Figure 6.17. Bi-level GLS OD-flow estimation for FW network; initial demand 50% of the actual

6.5.7. Non-Linear Optimization Method

6.5.7.1 Experiment Design

The performance of this estimation method, presented in Sections 3.5 to 3.7, is investigated by its implementation on a small network (Network A, Figure 6.1) with a limited number of departure intervals. The practical problems involved in implementing this method in large networks are explained in Chapter 5.

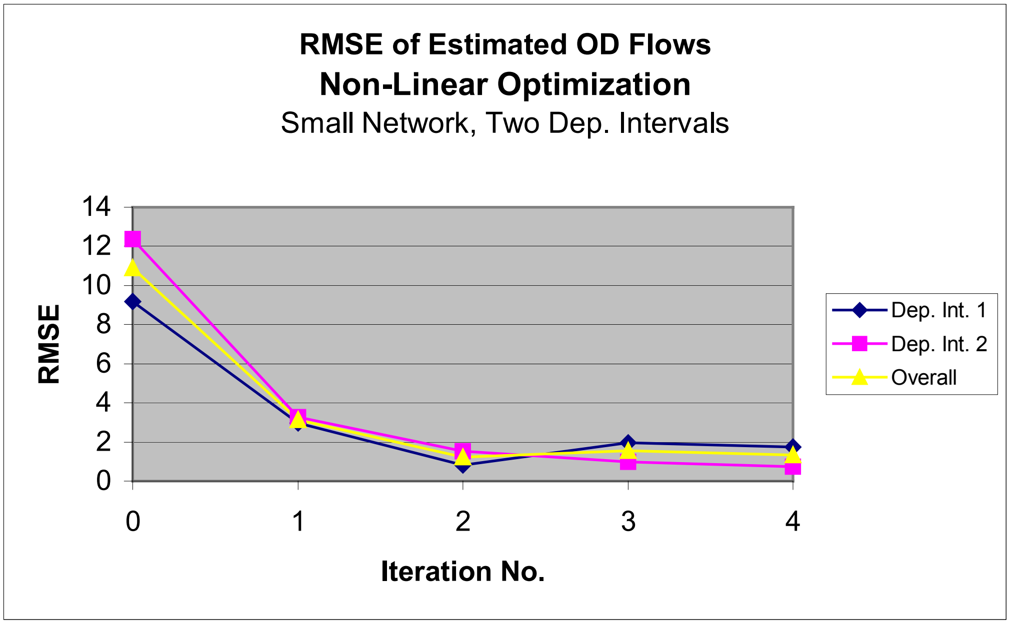
The network is loaded with a demand of about 700 vehicles per hour. The estimation period consists of two departure intervals of five minutes. Average travel speed in the network is about 40 miles per hour, which indicates that the network is not congested. Considering that there are only two origin and destination pairs and two departure intervals, the problem consists of four unknown OD-flow entries.

As explained in Chapter 5, in the first few iterations the problem is solved by using the bi-level GLS method, and in the final iteration, the non-linear optimization method is used.

6.5.7.2 Results and Discussion

The results are illustrated in Figure 6.18. In the first three iterations, the bi-level GLS method significantly reduces the RMSE of OD flows. In the fourth iteration, when the non-linear optimization method is used, since the estimated OD flows have already approached the vicinity of the solution, the improvement in the RMSE of OD-flow estimation is not significant (about 14 percent). As mentioned before, in order to achieve convergence, the initial guess of OD flows in the bi-level optimization should be close enough to the desired solution. As such, we cannot expect much improvement in the RMSE of the estimation. If the initial point is not close enough to the desired solution, the process may not converge or, in other words, may diverge toward other solutions. For instance, the non-linear optimization method was utilized after the first iteration, but the RMSE of OD-flow estimation increased (the results are not shown here).

In large networks, the divergence problem is aggravated by an increase in the number of unknowns. In general, the obtained improvement in OD-flow estimation is often too subtle to warrant the computational overhead of using the non-linear optimization method for OD-flow estimation in practice.



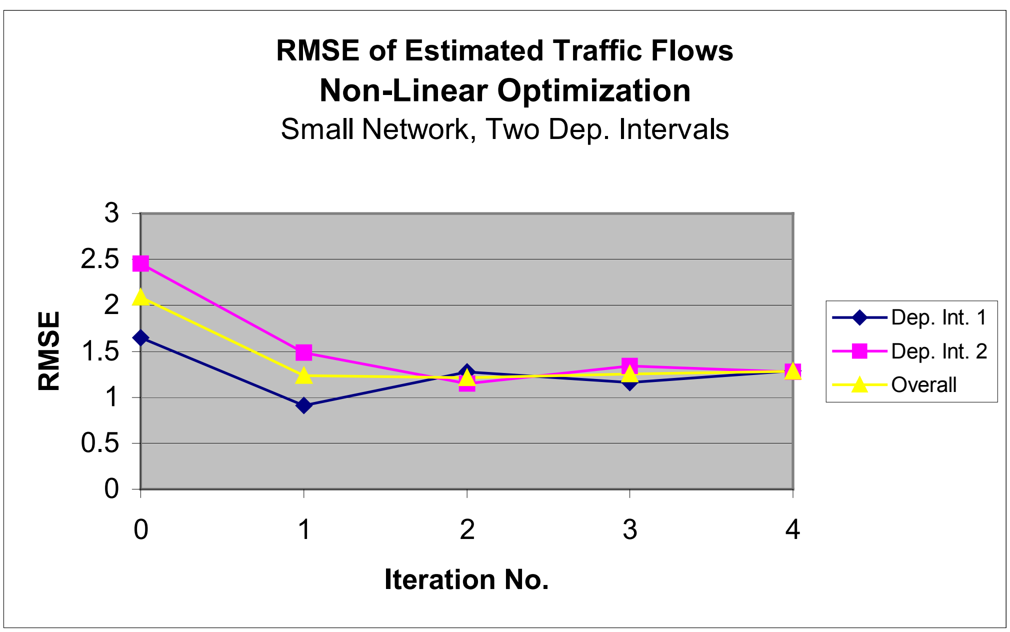


Figure 6.18. Non-linear optimization method to estimate OD flows—small network, two departure intervals

6.5.8. Effect of *a Priori* Information

6.5.8.1 Experiment Design

As explained in Chapter 4, the Bayesian inference method is adopted to incorporate the *a priori* OD-flow information or a target matrix, if available, with the estimates resulting from the flow observations. Use of *a priori* information may overcome the divergence problems that have been observed in some of the experiments. Such divergence is due to the lack of a global optimum, which is an inherent property of the problem. The use of the Bayesian inference method directs the estimated OD-flow values toward the desired target matrix. In the experiments presented in this section, we examine how the existence of an *a priori* OD-flow table improves the quality of the solution in terms of OD and link flows.

As explained in Section 5.2 and Appendix D, if Bayesian inference is invoked in the OD-estimation process, the resulting estimate in the last iteration of Bi-GLS or Bi-NLP methods are fused with *a priori* OD-flow estimates.

Several experiments are designed to examine the effect of *a priori* information on the quality of the solution. In these experiments, the scenarios considered are those in which the Bi-GLS method did not exhibit satisfactory performance, or where the OD-flow estimates were diverging from the presumed solution.

Experiment 1. The first experiment is conducted on Network B under congested loading, with one-minute observation intervals and five-minute aggregate departure intervals. The results of this experiment can be compared to the case in Section 6.5.1 (Figure 6.5) under the same experimental condition but without the

Bayesian inference method. In this test, it is assumed that the *a priori* OD flows (the target matrix elements) are 90 percent of the actual OD-flow values.

Experiment 2. This experiment is designed to examine the quality of the solution in terms of estimated OD flows, where inconsistent route choice assumptions are made. In this experiment on Network B under congested conditions, the real-world route choice is assumed to be an imperfect UE, but UE assignment is used for the OD-flow estimation. Observation intervals and aggregate departure intervals in this case are one and five minutes, and the results can be compared with experiment 4 in Section 6.5.2.

Experiment 3. In this experiment on Network B, both the observation and departure intervals are ten minutes. The *a priori* OD flows are 90 percent of the assumed actual values. The results of this experiment are comparable to the results in Section 6.5.4 and Figure 6.13.

Experiment 4. In all of the previous experiments, the estimation process was initiated with an arbitrary (mostly uniform) initial time-dependent OD-flow table. If the Bayesian inference was used, the resulting estimate in the last iteration was fused with *a priori* OD-flows. In this experiment, we will investigate whether or not the information obtained from the sample of observations improves the performance of OD-flow estimation when the process is initiated with the *a priori* OD flow table instead of an arbitrary one. It is assumed that the *a priori* OD flows are 80 percent of the actual ones. This experiment is conducted on Network B with uncongested loading. The chosen observation and departure intervals are two and ten minutes, respectively. The results can be compared to those of the experiments in Section 6.5.4 (Figure 6.13).

6.5.8.2 *Results and Discussion*

a) Experiment 1.

Figure 6.19 shows the effect of using the Bayesian inference in the last iteration of the OD-flow estimation when the network is congested, as described in Section 6.5.1.

The results indicate that there is significant improvement in the quality of the solution in terms of estimated OD flows if a reliable *a priori* OD-flow table is incorporated into the estimation process. Furthermore, comparing the RMSEs of the estimated OD and link flows in iterations nine and ten underline that the optimization problem does not have a unique solution in terms of the OD flows. As illustrated, there is a considerable change in the RMSE of OD-flow estimates, meaning the solution has changed substantially while the RMSE of the estimated traffic flows on the links has not changed significantly.

b) Experiment 2.

Figure 6.20 shows that by using the Bayesian inference method RMSEs of estimation in terms of estimated OD flows improve significantly. It is assumed that the *a priori* OD-flow table was underestimating the actual demand values by 10 percent. The results emphasize that the estimation method combined with *a priori* information is robust, even if the assignment assumptions are not completely consistent with the users' route choice.

c) Experiment 3.

Experiments in Section 6.5.4 on the effects of observation interval size showed that when both observation intervals and departure intervals are ten minutes, there is degeneration in the solution and the quality of OD-flow estimation in consecutive iterations does not improve. Figure 6.21 depicts that we can overcome this problem if a relatively consistent *a priori* OD-flow table exists. During this experiment, after four iterations of GLS estimation, the resulting estimated OD-flow is fused with the *a priori* information. In this case, there is 72 percent improvement in the RMSE of OD-flow estimation. Even though the *amount* of change in RMSE of estimated link flows is less significant, the relative improvement is more than 55 percent.

d) Experiment 4.

This experiment is conducted by assuming that the *a priori* OD-flow table functions both as the initial estimate of OD-flow (the input to the estimation algorithm) and as the Bayesian inference prior information in the last estimation iteration. Figure 6.22 illustrates that in the consecutive Bi-GLS estimation iterations, the estimates of the OD flows do not improve while there are improvements in the estimated link flows. However, in the last iteration, when the estimated OD-flow in the previous iteration is combined with the *a priori* OD-table by means of Bayesian inference, there is a significant improvement in the estimated OD-flows. Although, there is a slight increase in the RMSE of link flow estimates due to the application of the Bayesian inference, the errors in the link flows are not being minimized.

Comparing the RMSE of OD-flow estimation at the initial condition against the results in the fifth iteration shows that even if the *a priori* OD-flow values were rather close to the actual demand (80 percent of the demand values), the information obtained from traffic counts improves the estimation of the OD flows. It should, however, be noted that this improvement might not always occur.

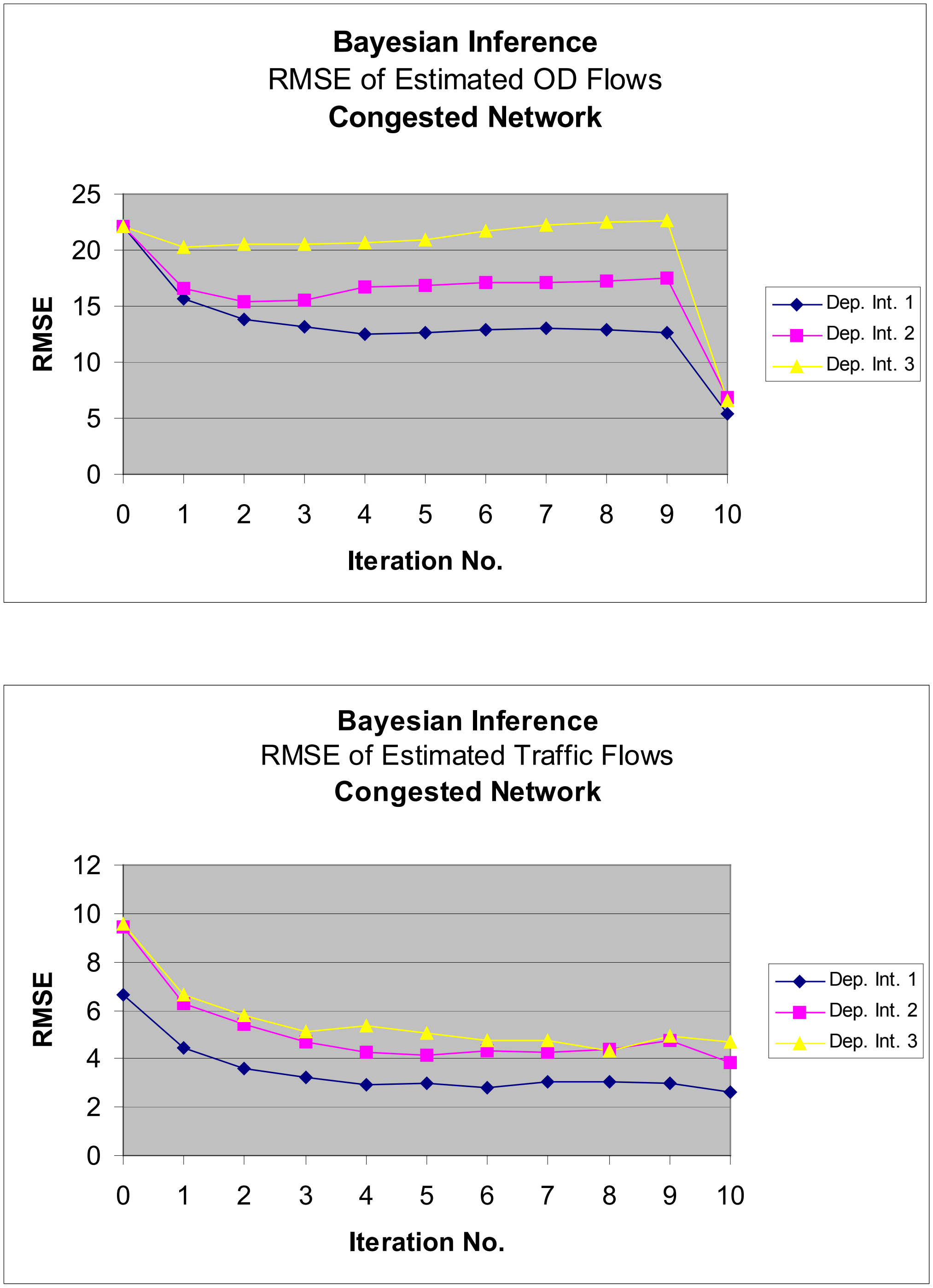
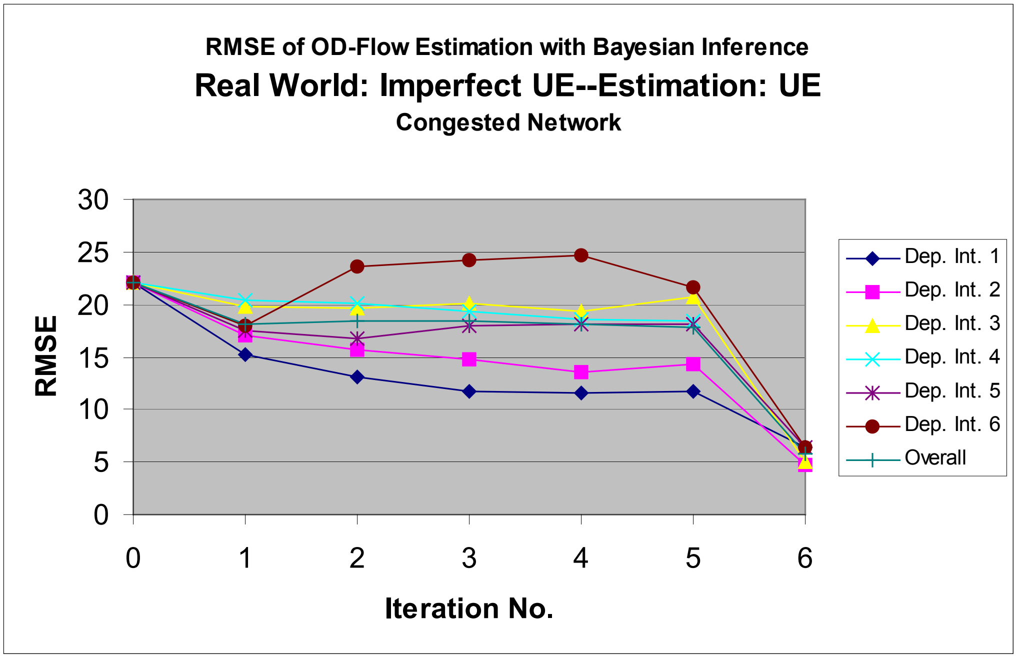


Figure 6.19. Effect of Bayesian inference on OD-flow estimation



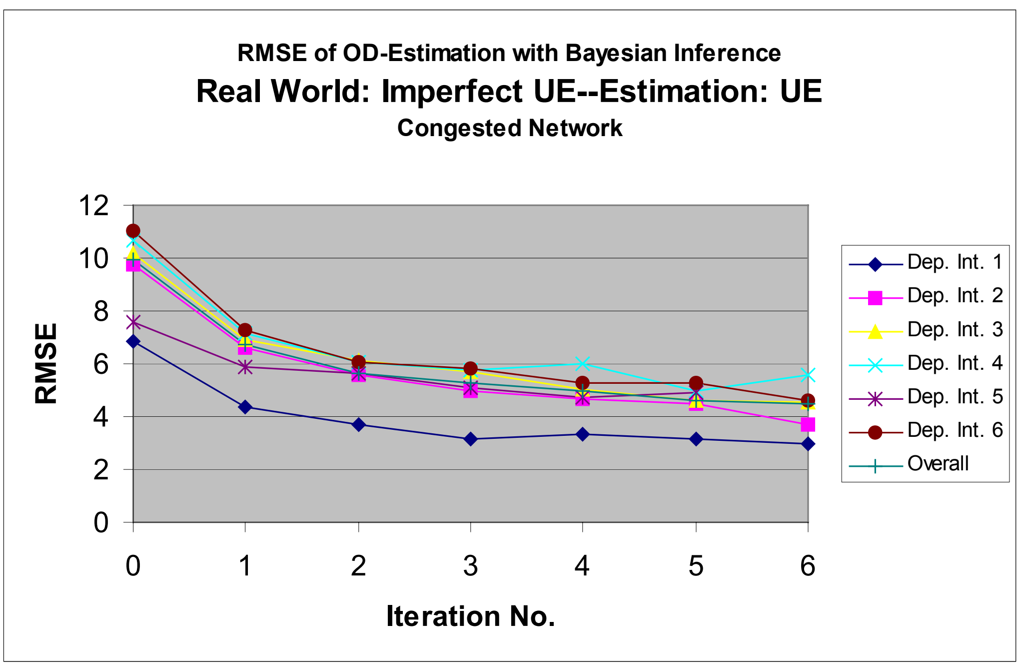
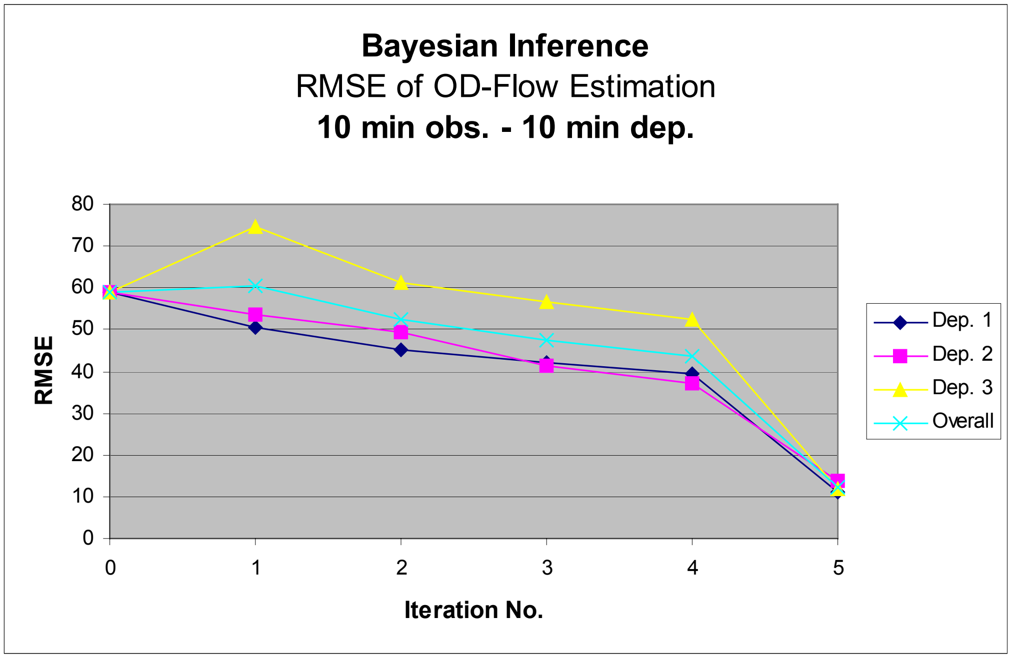


Figure 6.20. Effect of inconsistency in assignment assumptions; quasi-UE in the real-world and UE in OD-flow estimation



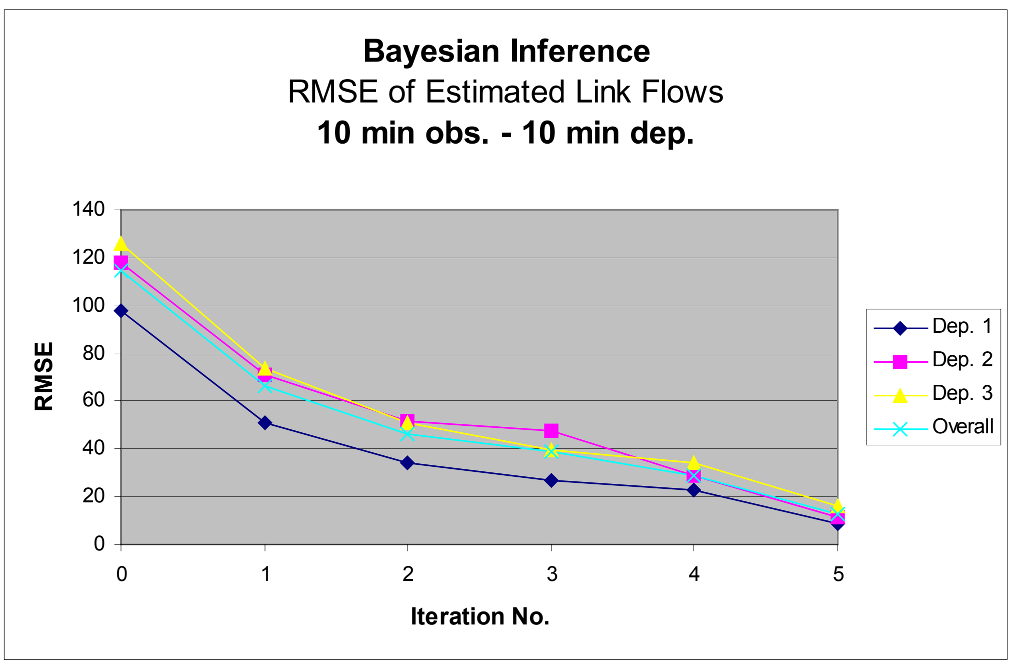
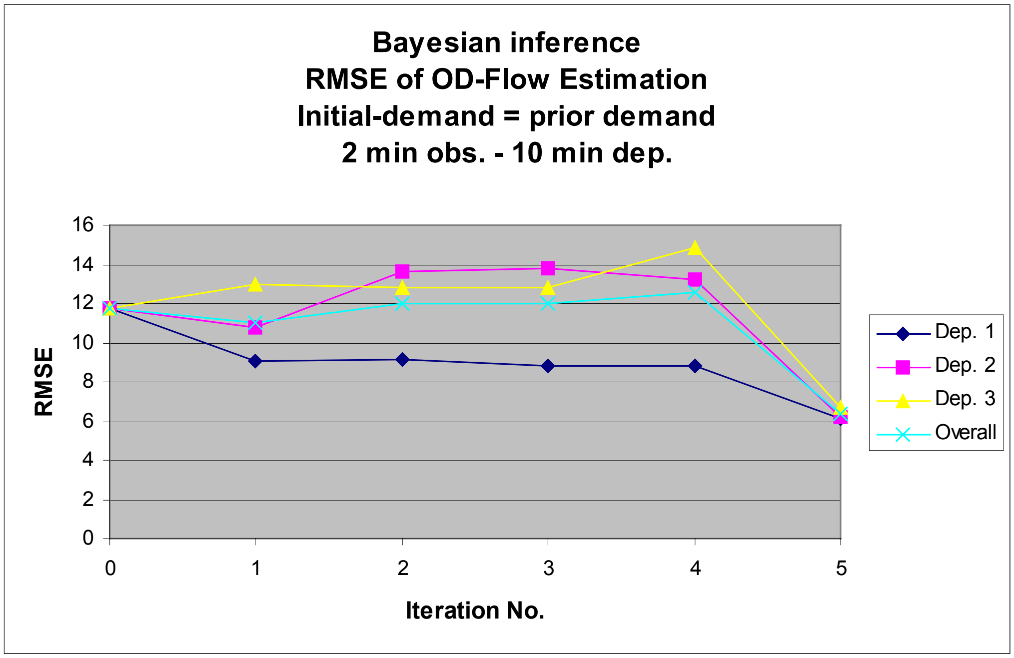


Figure 6.21. Effect of Bayesian inference on improving the OD-flow estimation; ten-minute observation intervals and ten-minute departure intervals



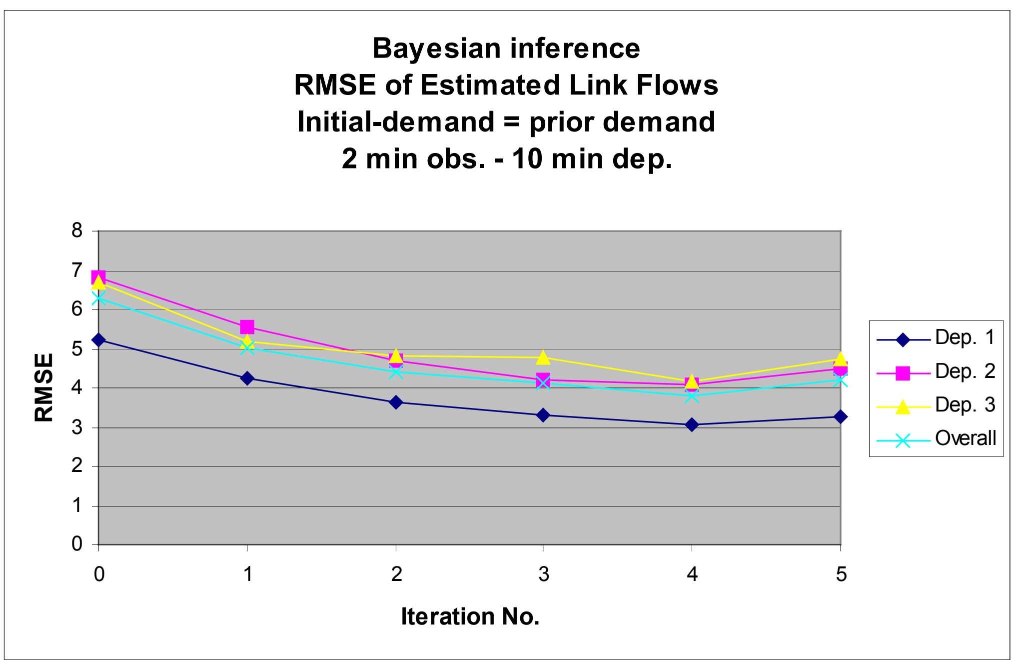


Figure 6.22. Effect of Bayesian inference; *a priori* OD-flow as the initial guess

6.6. Summary

In this chapter, several experiments were designed and performed and their results were discussed. In summary, the following points are noteworthy.

- The Bi-level GLS OD-flow estimation method exhibits a good performance in estimating OD-flows and especially in improving the external consistency of the Dynamic Traffic Assignment simulation program.
- Since the methods presented in this dissertation estimates OD flows based on the traffic flow in the network, it does not provide a good estimate of OD flows, as expected, when the traffic flow in the system is impeded due to, say, congestion. In these situations, the use of a hybrid model that estimates the OD flows based on both traffic flow and density is recommended. Another possible solution in these situations would be to choose larger aggregate departure intervals or to combine the estimates with *a priori* OD-flow information.
- In several experiments, the effects of inconsistencies in traffic assignment assumptions are examined. Since the error terms due to different sources are confounded, the closer the assumptions in the methodology are to the real world, the better the results of the estimation. Nevertheless, in most of the experiments, the Bi-GLS method was robust in the estimation of OD flows, especially when fused with prior OD-flow information.
- The ratio of the size of the aggregate departure intervals to the size of the observation intervals has a significant effect on the quality of the solution. If this ratio is not large enough, the problem may become under-specified (depending on the number of unknown OD-flow values and number of observation flow volumes). On the other hand, if

the chosen ratio is too large, though the effect of the undesirable randomness in the system is alleviated, the problem becomes similar to the static case. In general, the optimum value for this ratio depends on the topology of the network, the observation intervals of the available traffic flow data and the desired detail level of the estimated OD flows.

- The experiments indicate that the Bi-GLS method shows satisfactory performance in the estimation of OD flows in actual, large networks. The computational time of the algorithm depends on the time necessary to run the simulation by the DTA and the preferred number of iterations. As a rule of thumb, the upper-level optimization takes less than one tenth of the simulation run time (the lower-level optimization).
- Considering that the lower-level optimization problem is not well behaved, the non-linear optimization algorithm, though theoretically valuable, does not appear to improve meaningfully the quality of estimation obtained by the Bi-GLS method in practice.
- Availability of reliable *a prior* information on OD flows can significantly improve the performance of the estimation process, especially in terms of OD flows. Considering that the formulated bi-level optimization problem does not have a unique solution, fusion of the prior information and the evidence obtained from time-varying traffic flows directs the solution to the desired (local) optimal point.

CHAPTER 7. CONCLUSIONS AND FUTURE EXTENSIONS

7.1. Overall Conclusion

In this research, several formulations for the problem of estimating dynamic OD flows from time varying traffic flow observations are presented. In these formulations, it is sought to minimize the sum of squared errors in traffic volumes estimated by a dynamic traffic assignment (DTA) simulation program to which the estimated OD flows are input. There are several sources of error in the estimation of link flows. In the methods presented, the combined effect of errors from all sources is minimized. In this way, the estimation process can be used to improve the external consistency of the simulation-assignment program. It should be noted that if one could reduce the errors due to other sources, such as users' route-choice behavior, one would obtain a more accurate estimate of the OD flows.

In the presented bi-level generalized least-squares formulation (Bi-GLS), the relation between traffic flow volumes and OD flows is considered as quasi-linear, that is *in each iteration* link-flow proportions are assumed to be constant. Using this approach in each iteration of the algorithm, a new set of link-flow proportions is estimated by seeking the user equilibrium assignment solution. In solving the lower-level optimization problem, the DTA simulation program is used to assign the estimated OD flows in the previous iteration onto the network. The iterations are repeated a pre-specified number of times or until convergence criteria are met.

In the bi-level non-linear optimization (Bi-NLP) formulation, in which the upper-level is a *non-linear* generalized least-squares problem, the non-linear relation between link flows and OD flows is taken into account and the derivatives of link-flow proportions with respect to OD flows are explicitly included in the formulation.

This approach results in a set of simultaneous quadratic equations for solving the upper-level optimization problem. In each iteration, the derivatives can be obtained numerically by solving the lower-level problem using the DTA simulation program. It is shown that the non-linear optimization formulation is an inclusive form of the generalized least-squares estimation method, and with some simplifying assumptions, the GLS formulation is obtained.

Both methods presented can be used in a single-horizon or a rolling-horizon estimation process. In the rolling-horizon estimation, based on the assumptions made for the initial conditions, two approaches are proposed:

- fixed initial-point and
- free initial-point estimation

In the former, the information from the previous rolling estimation stage is used as the initial condition, and in the latter, the initial condition is set according to the state of the network in the real world at the beginning of each estimation stage.

Finally, the Bayesian inference method is presented to combine existing *a priori* information on OD flows with the OD flows estimated from the sample of time-varying traffic flow counts. The *a priori* information can be historical data obtained from surveys or can be the OD flows estimated from traffic counts in the previous estimation periods, the day, or the week before. In the case of annual events like holidays or special events, the *a priori* information could be the estimated OD flows during suitable periods in the previous years.

The procedures for the estimation of OD flows from traffic counts are implemented as an integral part of DYNASMART-P, the planning version of the

Dynamic Traffic Assignment (DTA) simulation program developed at the University of Texas at Austin. For the rolling-horizon OD-flow estimation, the first alternative, i.e. fixed initial-point formulation, is implemented.

7.2. Research Contribution

Deployment of Intelligent Transportation Systems is providing a large amount of valuable on-line and archived data. These data carry valuable information; however, so far they have mostly been used in real-time operational applications. This research has focused to extract information and build knowledge from on-line or archived data estimation of OD flows and external consistency checking of DTA systems. This information can be used in both transportation planning and on-line traffic control applications. Furthermore, the proposed methods are implemented in a DTA system in a way that may be used to improve the external consistency of the system with the real world.

Gathering time-dependent OD flow information directly by conducting surveys is very costly and time consuming. Therefore, researchers have attempted to use other methods to estimate the OD flows, one of which is to use the available and relatively inexpensive traffic volume counts.

In previous existing work on the estimation of *dynamic* OD flows from traffic counts, the effect of congestion in the estimation of dynamic OD flows is often ignored. Particularly, when trip makers have access to real-time traffic or route-guidance information, they are able to choose new paths, which adds to the dynamic characteristics of the transportation system.

In this research, the effects of congestion are addressed by considering that link-flow proportions are not constant, but are an implicit function of the unknown OD demand values. Considering this aspect, several methods for the estimation of dynamic OD flows from time-dependent traffic counts are presented. It should be noted that there are several sources of errors in the estimation of link flows. One of the characteristics of the presented methods is that the problem is formulated and implemented in a way that the combined effect of errors from all sources is minimized. In this way, the estimation process can be used to improve the external consistency of the simulation-assignment procedure, bearing in mind that the more accurate the estimates of other factors contributing to error are, the more reliable the estimated OD flows will be.

In this research several methods for estimation of OD flows and improvement of consistency of the DTA system are proposed. These methods are based on established optimization methods and are formulated in a way that the availability of other information from, say, cordon line counts, known OD flow values, probe vehicles and/or Advanced Vehicle Identification (AVI) systems can be incorporated in the formulation.

The methods formulated in this research are as follows:

- Unconstrained generalized least-squares estimation method.
- Bi-GLS, a bi-level generalized least-squares estimation method. The assignment problem is formulated as a constraint to the main problem and constitutes the lower-level optimization problem. DYNASMART-P has been used to solve the lower-level problem.
- Bi-NLP, in which the upper-level problem is formulated as a non-linear optimization problem. In the solution to the problem, the

derivatives of OD-flows with respect to link flows and link flow proportions are explicitly included.

- Extension of the problem to on-line DTA application by formulating in a rolling horizon framework. The case has been formulated in two scenarios, fixed-initial point and free-initial point situation. The mathematical solutions to both scenarios are obtained and the algorithm to the former scenario has been implemented.
- To use any available prior information or to direct the solution to a desired target matrix, usage of the Bayesian inference method has been suggested and formulated.

The distinction of the proposed methods as compared to the existing works in the literature can be summarized as follows.

- Bi-level OD-flow estimation does not appear to have been formulated or implemented in a dynamic transportation network. Its static formulation with some variations has been presented by Yang *et al.* (1994).
- Similar dynamic OD flow estimation has been formulated by Cascetta *et al.* (1993), but they have used proportional assignment. That is, the need to update the link-flow proportion values by solving the optimization problem at the lower level was ignored.
- A closed form solution to the upper-level non-linear optimization problem was derived in this research. The resulting formulation is a fixed-point problem format to which a numerical solution has been presented. Yang (1995) has used a similar formulation in the bi-level non-linear optimization of static cases, though he has used a linearized

influence function and has not reported attempts to obtain the solution mathematically.

- It is shown that the non-linear optimization formulation is an inclusive form of the generalized least-squares estimation method, and with some simplifying assumptions, the GLS formulation is obtained.
- Formulation of the problem in a rolling horizon application with free-initial point assumptions. In this formulation, one can explicitly include the real-world state of the system at the start of each rolling estimation period. This property prevents the propagation of estimation errors from one estimation period to another and provides a mechanism for consistency checking of the system at the beginning of each rolling period.
- In this research, the Bayesian inference method has been proposed to incorporate the *a priori* OD-flow information in the solution of the problem. Usage of the Bayesian method in the estimation of OD flows for the static case has already been suggested by Maher (1983), but its application in dynamic OD-flow estimation has not been reported.

7.3. Future Extensions

The following extensions can be considered imminent to the presented methods. In the non-linear optimization approach, to obtain better results, the following improvements in the solution process are suggested:

- Using a more stable method to estimate the derivatives of link-flow proportions with respect to demand. In static cases and in small networks the derivatives might be found analytically. In dynamic cases the simulation can be run several times and with different seeds for the random number generator to find the mean of link-flow proportion derivatives.

- Utilizing a more robust method to solve the set of simultaneous quadratic equations. Finding the solution to the set of simultaneous quadratic equations is one of the drawbacks of the non-linear approach, although finding the mean of the link-flow proportion derivatives, as suggested above, might alleviate the problem. As an alternative, the estimation process could be linked to commercial software packages to find the solution to the set of simultaneous non-linear equations more efficiently.
- Considering alternative approaches in dealing with probable singular link-flow proportion matrices. In the existing implementation, when the link-flow proportion matrix is singular, the associated row that causes the singularity is skipped. The resulting value of estimated OD-flows are then compared against the predetermined upper and lower limits on OD flows. Other alternatives might be to set the value to its estimate in the previous run or to substitute it with OD-flows of similar OD pairs.
- In both the quasi-linear and non-linear formulations, the properties of the DTA simulation program used to compute the link-flow proportions (and their derivatives with respect to demand, in the latter case) could also contribute to a slowdown in the convergence of the results or may cause a jump in the solution from one local optimum to another. In addition to non-linearity, non-convexity and non-continuity of the dynamic traffic assignment, the existence of randomness in some aspects of the simulation-assignment program may contribute to jumping between local optimal solutions. Running the simulator with different random number generation seeds and finding the mean of the

link-flow proportions in several runs might smooth out the process. With this provision, the simulation-assignment process should iterate several times in the lower level within a bigger loop of iterations of the bi-level OD-flow estimation. Besides, each run of the simulation itself consists of several iterations in order to find the equilibrium multiple user class assignment (RHMUC procedure). However, it should be noted that multiple executions of the simulation-assignment program in finding the mean link-flow proportions and their derivatives is a burden on the complexity of the problem in terms of the computation time.

- As mentioned in the rolling-horizon OD-flow estimation only the fixed initial-point formulation is implemented. When the data for a real-world network becomes available, the free initial-point formulation can be implemented. This method could prevent propagation of the estimation error from one estimation stage to the next.

7.4. Future Research

When the required real-world information is available, the proposed methods should be applied to a real network so the performance of the estimation method in terms of replicating the time-dependent traffic volumes on links can be measured. If an estimate of the actual OD flows in the network is available, the performance of the presented methods can also be measured in terms of the estimated OD flows. The performance of the method can also be compared with other estimation methods, such as conventional GLS estimation and the Kalman filtering technique.

The estimation OD flows proposed in this research are drawn from link-based information. Link-based formulation brings up the possibility of having multiple OD-flow solutions. More common utilization of Advanced Vehicle Identification (AVI) systems and Global Positioning Systems (GPS) and technological advances in locating wireless phones will provide (partial) OD-flow or path-based information. By using the presented Bayesian inference method, the route-based or partial OD-flow information can be combined with the estimated OD flows from traffic counts.

In this research we proposed the use of the bi-level GLS method and bi-level non-linear optimization to solve the obtained equation in (3.5) or (3.8). Another (theoretical) approach is to use Maximum Likelihood Estimation methods to find a solution to those equations.

From a broader perspective, the bi-level optimization method has applications in other fields such as revenue management and scheduling in the airline and trucking industries. In practice fleet scheduling and pricing are usually done independently, while in reality scheduling has direct impact on users' choice and the price they are willing to pay to use the service.

Appendices

APPENDIX A. IMPLEMENTATION OF ALGORITHMS

A.1. Pseudo-code of the main program

The pseudo code of the estimation process within the main DYNASMART-P program is shown in Figure A.1. The iterative procedure for OD-flow estimation is implemented in the outer loop of the main rolling horizon multiple user class (rhmucmain) program. The main added procedures are:

- OD_main(): to estimate the OD-flows using the generalized least-squares (GLS) method.
- Deriv_main(): to estimate the OD-flows using non-linear optimization with inclusion of the derivative of link-flow proportions with respect to demand.
- Bayes_main(): to implement the Bayesian inference procedures.
- STAT_OD_main(): to find the statistics of OD-flow estimation by any of the above methods.

To use the memory efficiently, the above subroutines allocate the required arrays to the memory and de-allocate them after returning to the main program, except for the global arrays, which are also used in other modules of the simulation program or the OD-flow estimation procedures. For instance, to avoid time-consuming I/O access, the link-flow proportion and the link flow observation arrays are stored in memory and are accessed when needed.

```
Read System.dat  /*The file containing the execution parameters*/
Call OBS_read()        /* To read the link flow observations */
Od_iter=0              /* In the first iteration, only the simulator is executed
                       (and not the OD-flow estimation). The input demand is
                       the initial guess of OD-flows to find the statistics of the
                       initial guess */
Do while( 0 ≤ od_iter ≤ od_iter_max +1)
        If od_iter < od_iter_max
                Call OD_main ()  /*to estimate OD-flows using GLS method */
        Endif
        If od_iter = od_iter_max        /* in the last iteration*/
                If deriv_flag = 1        /* optimization including derivatives*/
                        Call Deriv_Main ()
                Else                     /* GLS estimation*/
                        Call OD_Main ()
                Endif
        Endif
        If od_iter = od_iter_max + 1            /* after the last iteration */
                If (odest_flag = 1)  then       /* If OD-flow estimation*/
                        Call Bayes_main ()      /* to calculate the a prior
                                                parameters, or the posterior OD
                                                flows if the Bayes_flag is ON */
                Else
                        END
                Endif
        Endif           (continued in the next page)
```

Figure A.1. Pseudo code of DYNASMART-P main program including the OD-flow estimation procedures

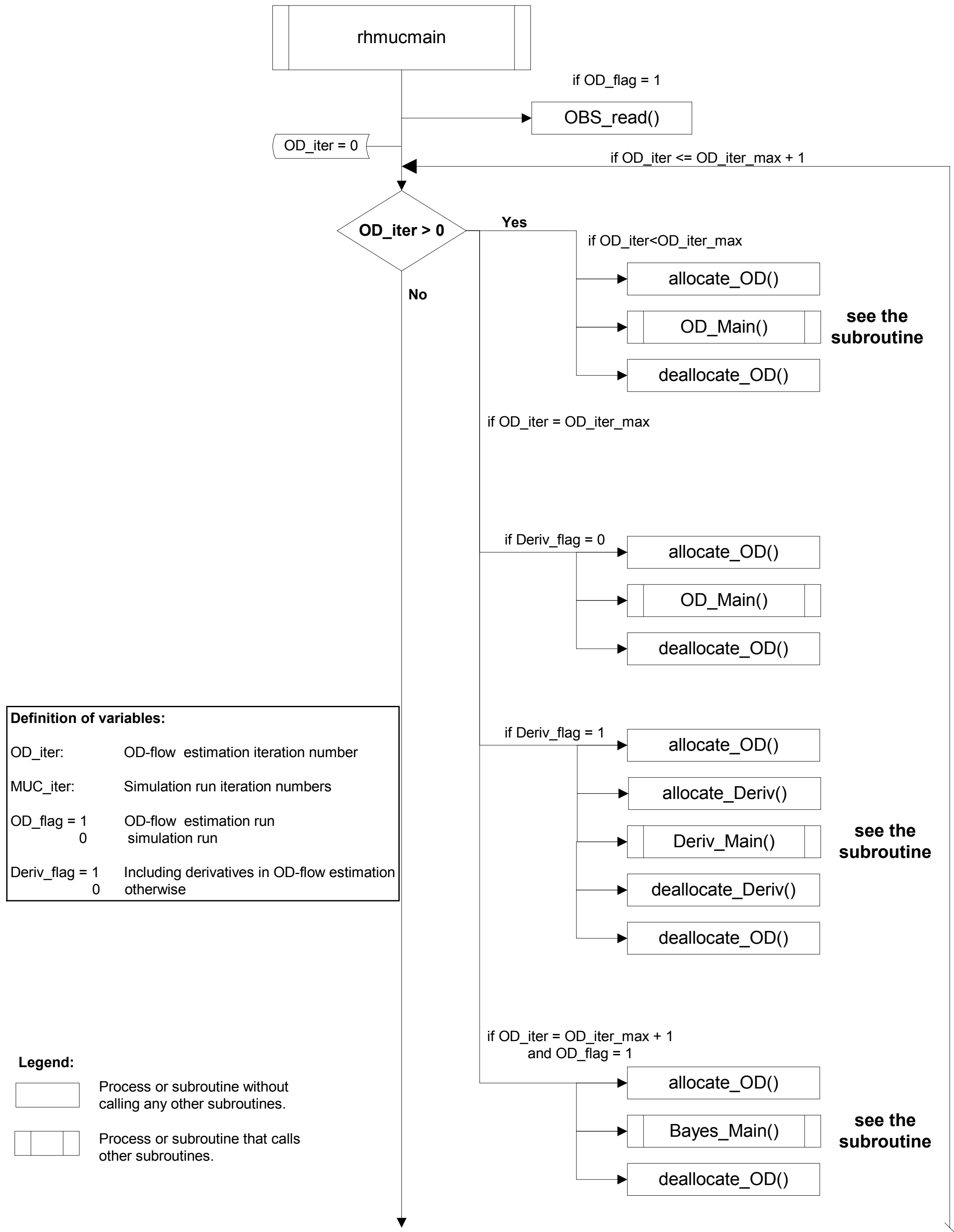
```
/* Start of simulator procedures */
muc_iter=0    /* Iteration counter for multiple user class dynamic
                 traffic assignment*/
Call dynasmart (maxintervals)       /* To set up the initial conditions
                                       and find the initial shortest paths
                                       */
Do while muc_iter < muc_iter_max
        Call dynasmart (maxintervals) /* Run the simulator */
        muc_iter = muc_iter + 1
Enddo
/* End of simulation run (rhmuc) */

Call STAT_OD_main ()       /* To compute the statistics of the OD-
                              flow estimation */
Enddo           /* End of OD-flow estimation loop*/

END
```

Figure A.1. Continued form previous page

Figure A.2 illustrates the organization of different modules and subroutines of the DYNASMART-P simulation program after integration of OD-flow estimation modules. For the description of the modules used in the simulator/assignment program, readers are referred to the DYNASMART-P user's guide (Mahmassani *et al*., 2000). The added modules for OD-flow estimation are explained in more detail in Appendices B to E.



Continued on next page

Figure A.2. Flow of DYNASMART-P simulation program

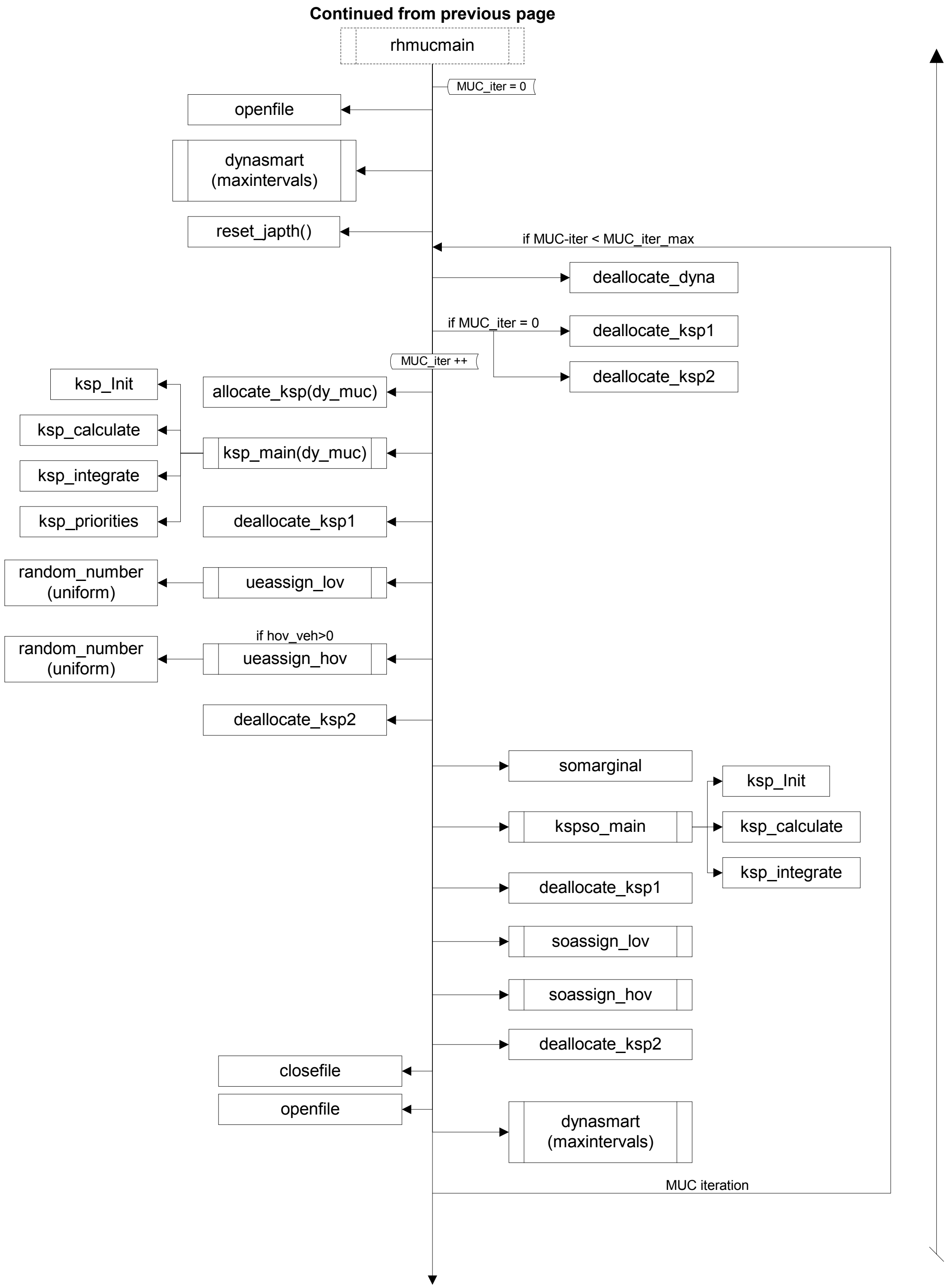


Figure A.2. continued

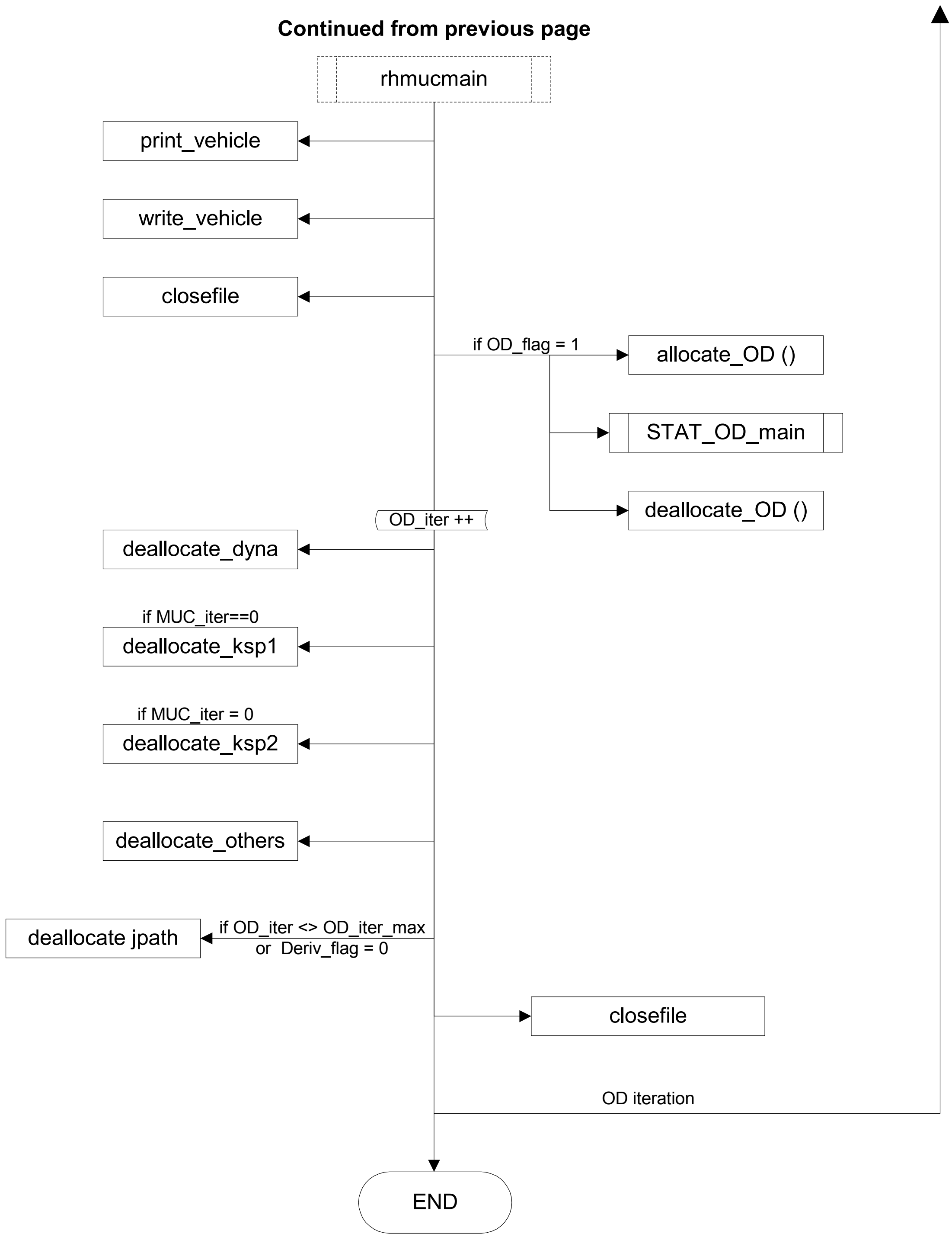


Figure A.2. continued

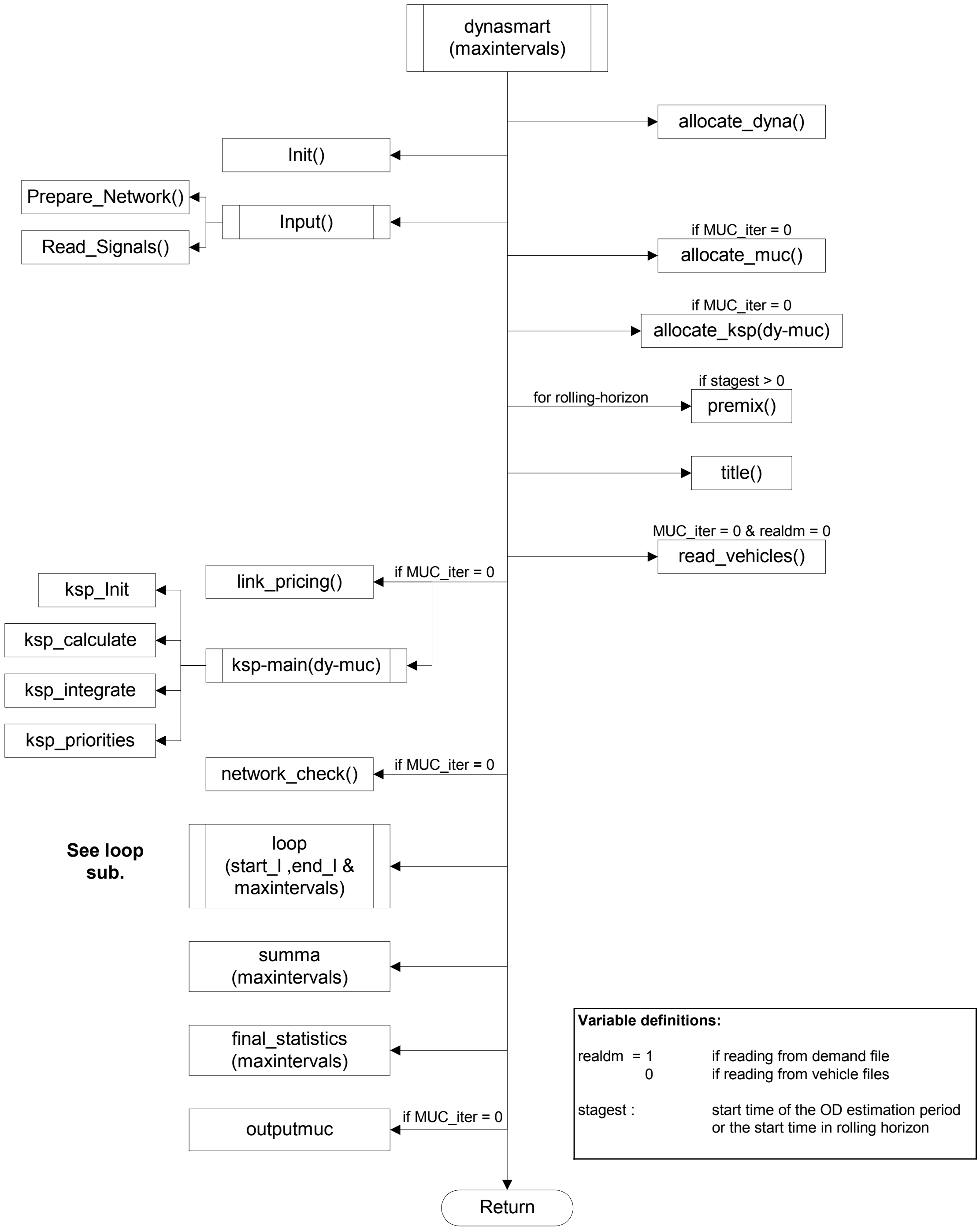


Figure A.2. continued

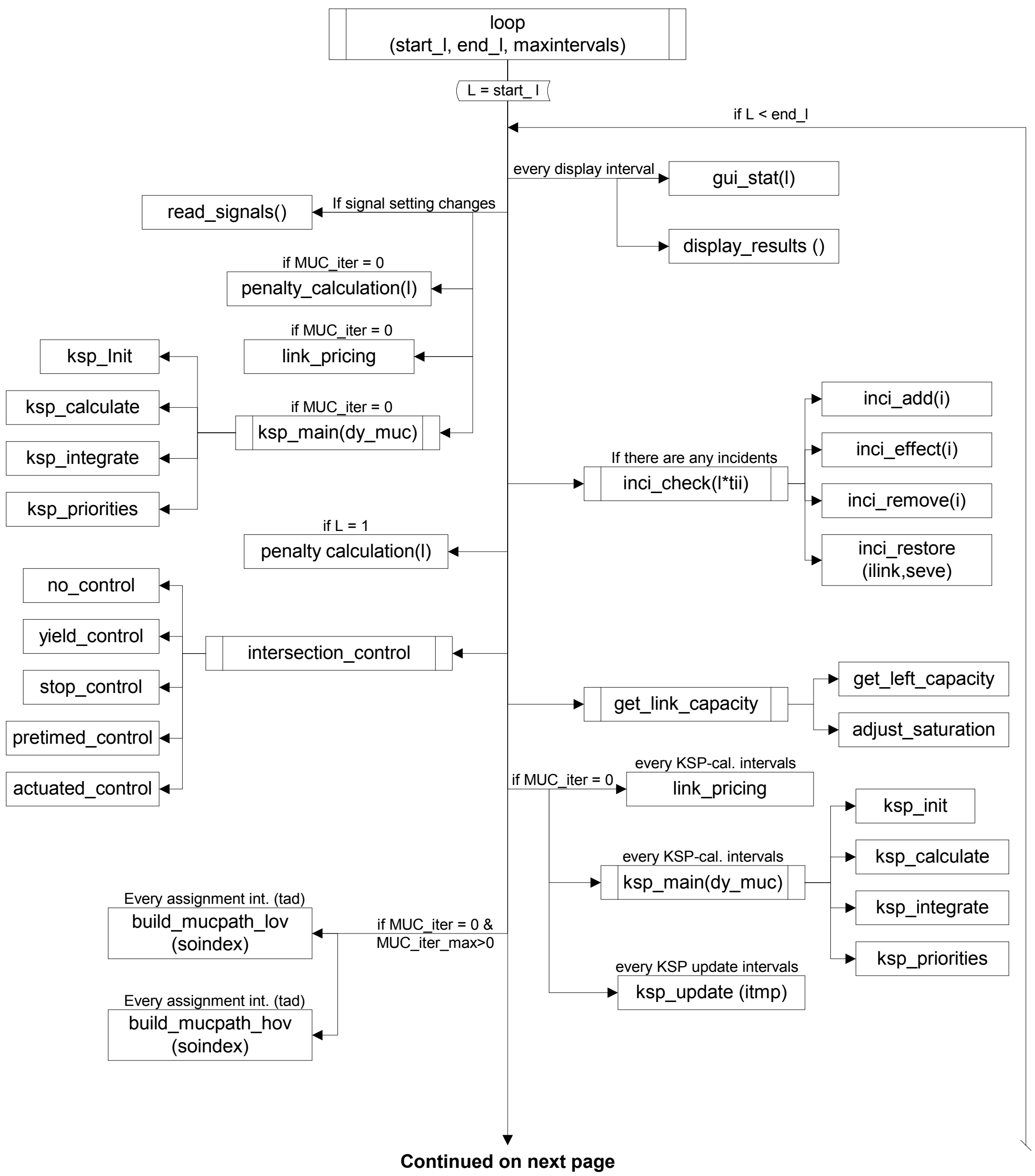


Figure A.2. continued

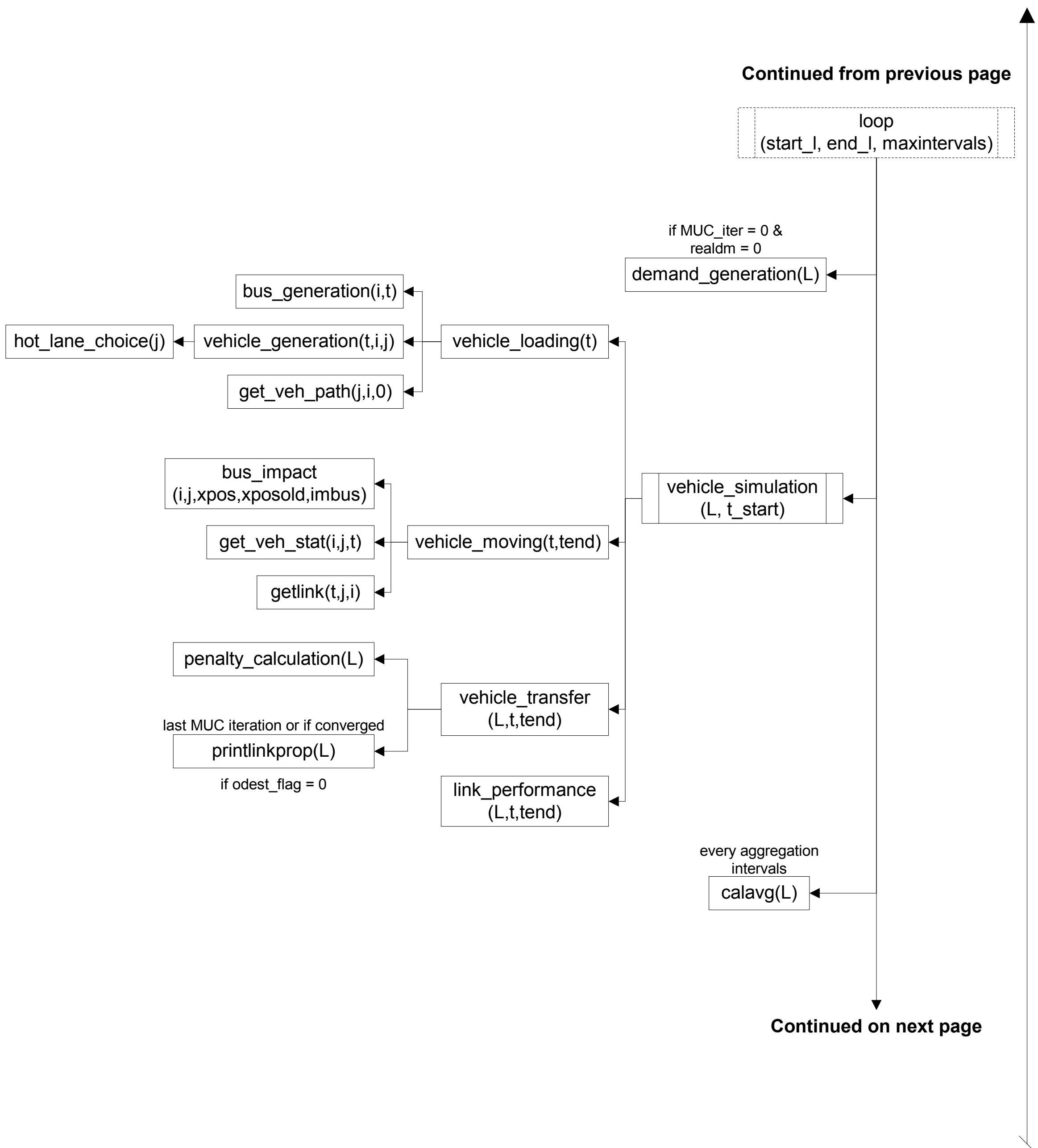


Figure A.2. continued

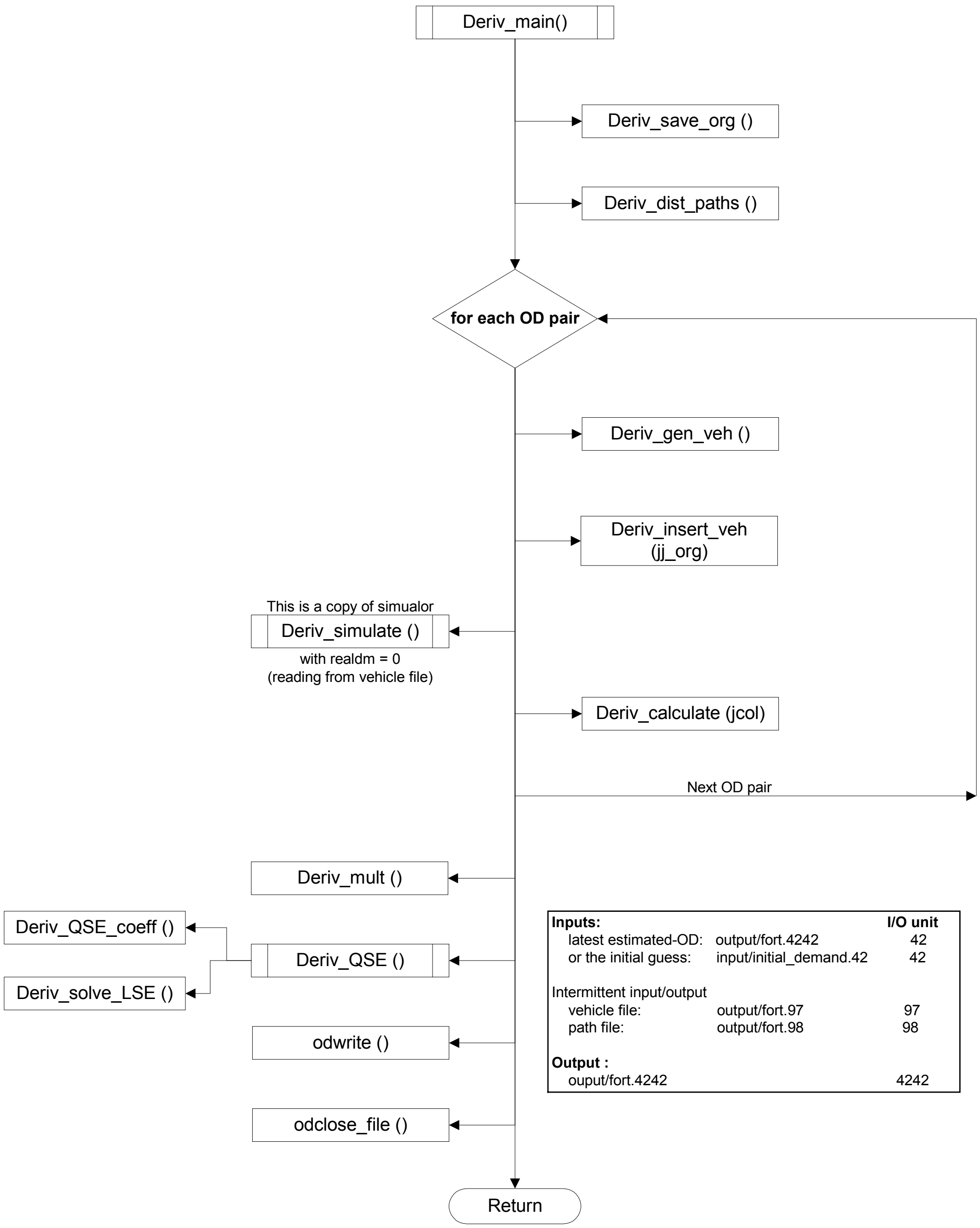
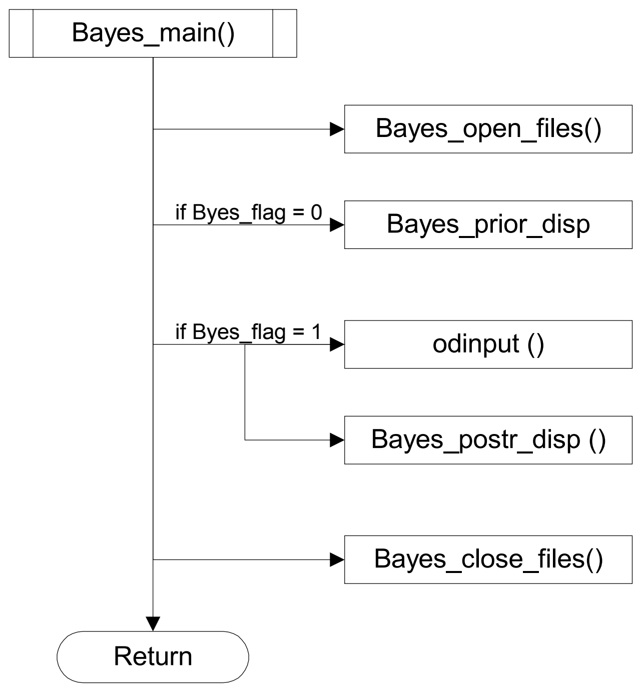


Figure A.2. continued



Inputs:		**I/O unit**
	input/prior_demand.42	42
	input/prior_disp.dat	601
	input/prior_abg.dat	602
Output :		
posterior OD:	output/fort.4242	4242
post. dispersion matrix:	input/posterior_disp.dat	701
post. alpha, beta, gamma	input/posterior_abg.dat	702

Figure A.2. continued

A.2. Input file "System.dat"

The control parameters of the simulation program and the OD-flow estimation procedures are included in the "System.dat". A sample content of the file is shown in Figure A.3. The first three lines contain the parameters used in the rolling horizon implementation of the DTA system. For description of these variables, readers are referred to DYNASMART-P users' manual (Mahmassani *et al*, 2000). The main variables added for OD-flow estimation are as follows:

odest_flag:	1 if OD-flow estimation run, 0 otherwise (simulation or assignment run only).
iter_od_max:	The required number of iterations (in the outer loop) for OD-flow estimation.
msa_flag:	1 if method of successive averages is used to average the estimated OD-flows in successive iterations; 0 otherwise.
rload_period:	The period used for loading of the network before the start of the estimation stage (rolling horizon implementation).
w_r:	(window_ratio) the fraction of stage-length that OD estimation is considered final (used in calculating the estimation statistics).
deriv_flag:	1 if the derivatives are included in the formulation (non-linear optimization problem); 0 otherwise (GLS OD-flow estimation).

iter_deriv_sim: number of simulation runs in OD-flow estimation when derivatives are included (to calculate the derivatives).

nveh_add_path: number of vehicles added per distinct path in calculation of derivatives of link-flow proportions with respect to demand.

precision: the required precision in terms of RMSR (Root Mean Squares of Residuals) in consecutive iterations in solving the set of quadratic simultaneous equations in non-linear OD-flow estimation.

max_deriv_iter: maximum number of iterations in solving the set of simultaneous quadratic equations if precision is not achieved.

od_min: The minimum number of vehicular trips between any OD pairs during each aggregate departure interval. It is recommended that this parameter be set to a positive number (at least one), thus the OD estimation procedure can update the pertinent estimates in the following iteration.

od_max: The maximum number of vehicular trips between any OD pairs. This value can be chosen subjectively based on the length of the aggregate departure intervals. This value is useful especially if the link-flow proportion matrix becomes singular (due to the uncompleted trips initiated toward the end of the estimation period).

obs_interval: the length of link-flow observation intervals in minutes.

od_interval: the length of aggregate departure intervals in minutes. This variable is specified here for the cases where realdm=0 (reading from the vehicle file). The value of od_interval should be consistent with the values specified in demand files, if provided (i.e. when realdm=1).

Bayes_flag: 1 if Bayes inference posterior variables should be computed;

0 otherwise (if odest_flag is ON, the prior variables for future estimation runs will be calculated).

```
  30.0   30.0   30.0   0.0
   0      0      1     3     50
  10     50      0.5   150
   0      5      1     0.0   1.0
   0      0      2     0.5   100
   1     300
  10.0   10.0
  1      0

roll horizon stagelength stagest
ireroute itedex realdm nu_ksp ksp_agg_cap
ftr tad muc_diff no_via
odest_flag  iter_od_max  msa_flag load_period  w_r
deriv_flag iter_deriv_sim nveh_add_path  precision
max_deriv_iter
od_min  od_max
obs_interval  od_interval
Bayes_prior_flag  Bayes_postr_flag
```

Figure A.3. A sample content of file "System.dat"

The OD-estimation procedures are added to the outer loop of the main simulation and assignment program. The process is designed such that in an OD-flow estimation run the GLS estimation procedure is iterated for 'iter-od-max' number of times.

If non-linear OD-flow estimation is set active (deriv_flag = 1), the process will be executed once at the end of GLS OD-flow estimation. In this case the vehicle files (fort.97 and fort.98 resulting from the simulation) generated in the last run of GLS estimation will be used as the vehicular demand input files (instead of OD table files). To find the derivatives of link-flow proportions with respect to demand flows, the estimated time-dependent OD-flow values in the last GLS OD-flow estimation run, is augmented incrementally one cell (OD pair) at a time. The number of added vehicles depends on the number of distinct paths between the pertaining time-dependent OD pairs and it is equal to the number of existing distinct paths in the simulation multiplied by the variable 'nveh_add_path', as specified in "System.dat" file (recommended to be a small number in the range of two to five). The number of simulation runs to find the equilibrium assignment solution with augmented vehicle files is controlled by variable 'iter_deriv_sim' in "System.dat" file. To avoid jumping from one solution region to another, a value of zero or one is recommended for this variable.

To control the lower and upper bound of the estimated demand values, particularly when some of the vehicles cannot reach their destinations by the end of the estimation period (causing singularity in the link-flow proportion matrix), two variables, 'od_min' and 'od_max', are specified in the "System.dat" file. The units of these variables are the number of vehicles per departure interval.

The non-negativity condition insinuates the use of a value of zero for 'od_min'. However, setting 'od_min' to zero, will cause all the entries in the pertaining column of the link-flow proportion matrix to become zero, generating a singular matrix. Moreover, the OD-flow values of zero could not be updated in the consecutive iterations.

Based on the above discussion, a small positive value is suggested for 'od_min'. In the experiments, a value of one is chosen for 'od_min'. The value of 'od_max' should be chosen based on the prevailing network characteristics. Unless otherwise stated, a value of 100 is used.

In the experiments, since real-world link-flow observations do not exist, a time-dependent OD-flow table was presumed and DYNASMART-P was run to find the 'ground-truth' link flows. In this case, the variable 'obs_interval' in the "System.dat" determines the length of the time interval during which the vehicle flow volume is accumulated. The variable 'od_interval' is the aggregate departure interval length and should be equal to the departure time intervals specified in the demand input files.

The variable "Bayes_flag" specifies if Bayesian inference process should be activated at the end of the OD-flow estimation process. If the value of this parameter is set to one, the Bayesian inference will be run and the posterior demand values and the pertaining parameter values will be calculated. However, if in an OD-flow estimation run (odest_flag=1), Bayes_flag is set to zero, the Bayes_main procedure will be invoked but only the prior Bayesian parameters (alpha, beta and gamma variables and dispersion matrix) are calculated.

To make the OD-flow estimation procedure more stable and prevent from jumping from one local optimal region to another, the method of successive averages (MSA) is used in consecutive iterations (see Section 5.4). The MSA method can be activated by setting the value of variable 'msa_flag' in the file "System.dat" to one. In this case, the MSA method as shown in equation (5.3) will be used. However, by setting this variable to zero, the estimated OD-flow values in each iteration are treated independently from the values obtained in the previous iterations. This is achieved by setting the value of i (the variable 'weight' in the code) to zero.

A.3. Multiplication of link-flow proportion matrices

The characteristics of link-flow proportion matrix are utilized to speed up matrix multiplication in the code. That is, all the entries in the matrix, $\boldsymbol{p}_{\tau,k}^{l,t}$, are zero if the observation interval t is prior to the start of the aggregate departure interval τ. That is, vehicles departing at any time interval τ cannot contribute to flows observed before the start of τ.

The above feature of the link-flow proportion matrix causes all non-zero cells to reside in a step-like part of the link-flow proportion matrix as shown in Figure A.4. In the code, the number of non-zero rows for each departure interval is calculated and used in the multiplication of matrices when appropriate.

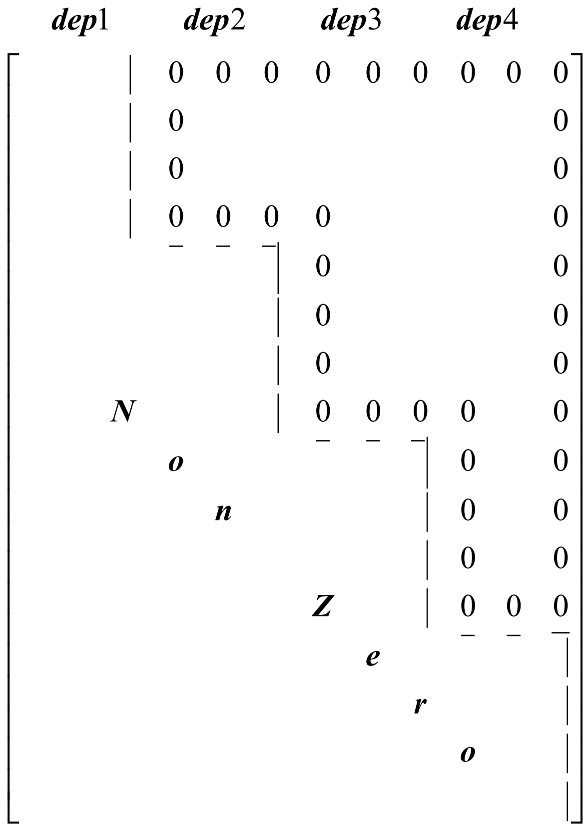


Figure A.4. Non-zero entries in link-flow proportion matrix

A.4. Finding derivatives of link-flow proportions with respect to OD flows

DYNASMART-P simulator program is used to find the derivatives of link-flow proportions. To find the derivatives of link-flow proportions with respect to any OD flow element of the time-dependent OD-flow table, an incremental number of vehicles are added to that particular demand flow element. For each obtained augmented OD-flow table, the simulation program is run and a new matrix of link-flow proportions are calculated. The derivatives of link-flow proportions are then computed as follows:

$$\nabla \boldsymbol{P}_{(\nu,\boldsymbol{o},\boldsymbol{d})} = \frac{\partial \boldsymbol{P}}{\partial \boldsymbol{d}_{(\nu,\boldsymbol{o},\boldsymbol{d})}} \approx \frac{\boldsymbol{P}_2 - \boldsymbol{P}_1}{\Delta \boldsymbol{d}_{(\nu,\boldsymbol{o},\boldsymbol{d})}} \tag{A.1}$$

where $\boldsymbol{P}$ is the matrix of link-flow proportions and $\boldsymbol{\Delta d}$ is the induced increment in the OD flow $\boldsymbol{d}_{(\tau,\boldsymbol{o},\boldsymbol{d})}$.

To minimize the effect of inherent randomness in the simulation, the algorithm keeps track of the vehicles' paths, stores it in a vehicle file and in the subsequent runs of the simulation, it assigns the vehicles based on vehicles' initial paths stored in that file (the vehicle files fort.97 and fort.98 in DYNASMART-P) . In DYNASMART-P, vehicles are generated on links and are simulated from origin *nodes* to destination *zones* (as opposed to origin-destination zones). Therefore, to compute the total number of incremental vehicles that should be added to each OD pair, the number of distinct paths between each origin *node* and each destination *zone* in the simulation is counted. For each distinct path to a destination, a set of new vehicles are generated and augmented to the vehicle file. The number of generated vehicles is obtained according to the following equation (the variable nveh_add_path is specified in the System.dat file).

$$\Delta \boldsymbol{d}_{(\nu,o,d)} = (\text{nveh\_add\_path}) \times (\text{no. of distinct paths})_{(\nu,o,d)} \tag{A.2}$$

The process is repeated for each entry in the time-dependent OD table to compute the derivatives of link-flow proportion matrix with respect to all OD flows.

APPENDIX B. PROGRAMMER'S GUIDE FOR OD ESTIMATION USING GENERALIZED LEAST-SQUARES METHOD

B.1. List of subroutines

Main subroutine **OD_main ( )**
Subroutine **odinput** ( )
Subroutine **odread** ( )
Subroutine **odwrite** ( )
Subroutine **odclose_files** ( )

The following subroutines are used as auxiliary functions:

Subroutine **od_convert** (jcol, nod, noz, idep, norg, ndest)
Subroutine **od_convert_rev** (jcol, nod, noz, idep, norg, ndest)

B.2. List of input files:

STATE_OBS.888:	File containing the real-world traffic flow observations.
System.DAT:	File containing parameters controlling the execution of the program.
Actual_demand.42	File containing the actual demand, in case it exists. In the experiments, this file will be used to find the measure of performance of the OD-estimation method.
Initial_demand.42	Initial guess of time-dependent OD table.
Fort.4242	Intermittent values of estimated OD table, used for successive averaging.

(Link-flow proportion values are calculated in the simulator and passed as an array)

B.3. List of output files:

Fort.4242	File containing the final values of the estimated time-dependent OD flow values.

B.4. Parameters:

no_of_origins:	Number of origin zones
no_of_destinations:	Number of destination zones
nod:	Number of OD pairs in the network.

depart_interval:	The length of aggregate departure intervals in minutes
obs_interval:	The length of observation intervals during which the traffic volume on links are accumulated and reported
no_obs_dep:	Number of observation intervals in each aggregate departure interval
nth_od_max:	Number of aggregate departure time intervals in an OD-estimation period
niobs_max:	Number of observation intervals in an OD-estimation period
nlink_w_detector:	Number of links which have flow measurements
igammaj:	Total number of OD flow entries that should be estimated (nth_od_max * nod)
nints_load:	Number of aggregate intervals used for loading the network before the start of the estimation period
startload:	Starting time of the loading period (before the estimation period). This variable is equal to 'stagest' in the rolling horizon implementation of the simulator.
starttime:	The starting time of the estimation period.
endtime:	The end of the estimation period.
rload_length:	The length of the loading period
w_r:	Window ratio, the fraction of the estimation period that the estimated OD is deemed to be final. It is only used in computing the statistics of the estimation.
od_min:	The minimum number of vehicular trips between any OD pairs during each aggregate departure interval.
od_max:	The maximum number of vehicular trips between any OD pairs.

odest_flag: 1 if the run is for OD estimation,
0 otherwise (simulation or planning run).

deriv_flag: 1 if non-linear OD flow estimation
0 otherwise.

bayes_flag: 1 if Bayesian inference is used to incorporate the *a priori* information,
0 otherwise.

itedex: The maximum number of simulation iteration to find the assignment (UE, SO, etc.) solution

iter_od_max: The required number of iterations for OD estimation.

B.5. Primary arrays:

B.5.1. Allocatable arrays

link_prop_lt (i, j): Two dimensional link-flow proportion matrix, that is the proportion of demand flows between OD pair (o, d) which start their trip at departure interval τ which contributes to flows on link l, during observation interval t. The row i represents (l,t) combination and the column j represents the (τ, o, d) combination.

link_prop_load (i, j): The link-flow proportion matrix associated with flows departing in τ in the loading period but observed on links during the observation interval t in the estimation period.

odvec(i): A column vector of time-dependent OD flows (to be estimated).

odvec_load(i): A column vector of time-dependent OD flows pertinent to the loading period.

odarr(i, j, k): The estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination zone.

odarr_old(i, j, k): The previous estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination zone.

odarr_load(i, j, k): The estimated time-dependent OD flows pertaining to the loading period (for rolling-horizon implementation or non-zero initial conditions), i is the departure interval number, j is the origin zone and k is the destination zone.

flow_diff(i): Difference between the link-flow observations and estimated link flow volumes obtained from the simulation based on the estimated OD flows.

B.5.2. Permanent arrays:

observation_org(i): Vector of observed time-dependent link flows.

observation(i): Vector of estimated time-dependent *net* link flows that is due to OD flows departing during the estimation period.

jlink(irow): The link number associated with row number irow in the observation vector.

jobst(irow): The observation interval associated with row number irow in the observation vector.

The two vectors **jlink** and **jobst** are used for bookkeeping to keep track of the links and observation intervals with available flow measurements. Therefore, the links and observation intervals with flow measurements are not required to be sequential.

link_flow(i): Estimated link flow volumes based on the estimated OD flows obtained from the simulation.

flow_minus(i): The estimate of the portion of link volume that is due to the OD flows initiated during the loading period. This is deducted from the observed volumes to estimate the portion of flow due to the OD flows departing during the estimation period.

B.6. Main subroutine OD_Main ( ):

Function: This main subroutine calls other subroutines in order to estimate the time dependent OD flow from the real-world traffic flow measurements using the Generalized Least-Squares method. The real-world observations are read separately in the OBS_read subroutine or are fed to the system by a data broker.

Pseudo Code:

Start of OD_main

Call **odopen_files**	/* to open the files */
Call **odinput**	/* to calculate the required variables */
Call **odread**	/* to read the arrays from the pertaining files */
Call **odcal**	/* to calculate the OD demand flows */
Call **odwrite**	/* to write the output arrays to output files */
Call **odclose_files**	/*to close the files */

End of OD_main

B.7. Subroutine odopen_files ( )

Function: To open the input data files. In the first iteration, it reads the given input/initial_demand.42 file. After the first iteration, it opens the intermittent estimated OD flow table from output/fort.4242 file.

Pseudo Code:

```
Start of odopen_files
        Check if output/fort.4242 exists,
                If yes, open it as unit #4242,
                otherwise, open input/initial_demand.42 as unit #4242
End of odopen_files
```

B.8. Subroutine odinput()

Function: To calculate and assign the variables used in the OD-estimation procedures.

Pseudo Code:

Start of odinput

Read the length of aggregate departure intervals from I/O unit 4242 (initial or estimated OD file).

Compute and store the variables.

End of odinput

B.9. Subroutine odread()

Function: This subroutine reads and stores the estimated OD flows in the last iteration of the estimation process (or their initial guess) in a vector.

Pseudo Code:

Start of odread

Read the constants of demand file I/O unit 4242 (initial or estimated OD file).

Read OD flows pertaining to the loading period.

Read the initial or the OD demand flows estimated in the last iteration (**odarr_old)**.

Multiply the demand values by the multiplication factor (multi).

Convert the loading OD matrix into a vector (**od_load**).

End of odread

B.10. Subroutine odcal()

Function: This subroutine finds the (estimated) net flow to be generated by the OD flows during the estimation period, checks if two columns of link-flow proportion matrix are identical (a cause of singular matrix), and finds the least-squares estimate of OD flows in the equation **LP*OD=C** by solving the set of simultaneous equations ($\mathbf{LP^T*LP)*OD=LP^T*C}$

Where

LP	is the link-flow proportion matrix
$\mathbf{LP^T}$	is the transpose of the link-flow proportion matrix.
C	is the vector of link-flow observations
OD	is the vector of time-dependent OD demand flows.

It stores the estimated demand flows in **odvec** vector.

Pseudo Code:

Start of odcal

Compute the portion of the flow on links that are associated with flows initiated during the loading period (**flow_minus**) based on the estimate of link-flow proportions from the simulator (**link_prop_load**).

Find (the estimate of) the net flow (**observation)** that is associated with the OD demand flow in the estimation period.

Calculate the sum of the elements in each column (OD pair) of link-flow proportion matrix.

Do icol = 1, total number of columns in the link-flow proportion matrix (igammaj)

If sum of elements in a column is zero, write the error prompt and go to the next column.

Do icomp = icol+1 , igammaj

Check if all entries in column icol and icomp of link-flow proportion are equal.

If they are all equal, prompt the user, exit the *do* loop and continue the process.

Enddo

Enddo

Multiply the link-flow proportion matrix by its transpose and store the results in **c_prime** matrix.

Multiply the link-flow proportion matrix by **observation** vector and store the results in **q_prime** matrix.

Write some temporary test files (optional).

Solve the set of simultaneous equations **c_prime*OD = q_prime** for OD using elimination method.

Store the results in **odvec** vector.

Write **odvec** in "output/od_vector.temp" file

End of odcal

B.11. Subroutine odwrite()

Function: This subroutine averages the successive estimates of OD flow values, checks them against the pre-specified minimum and maximum values and writes the results to output files.

Pseudo Code:

Start of odwrite

> Find origin, destination and departure interval for each row of the **odvec** vector.
>
> Store the **odvec** vector in the **odarr** matrix.
>
> Set the intra-zonal OD flows to zero.
>
> Take the weighted average of the estimated OD flows with the estimated OD flow in the last iteration (or the initial guess of OD flows) using the method of successive averages.
>
> Increment the iteration counter and write the constants of the demand file to the demand intermittent file ("outpur/fort.4242).
>
> Check the averaged OD flow values against minimum and maximum acceptable demand flows. If it is out of bound, set it to the boundary value and make a note in the error file.
>
> Write the OD flow values pertaining to the loading period and the Estimated values of the estimation period to the output file "output/fort.4242."

End of odwrite

B.12. Subroutine odread()

Function: This subroutine reads and stores the initial guess of OD demand flows or the estimates of OD flows obtained in the last iteration in a vector.

Pseudo Code:

Start of odread

Read the constants of demand file I/O unit 4242 (initial or estimated OD file).

Read loading period OD demand flows.

Read the initial or the OD flows estimated in the last iteration (**odarr_old)**.

Multiply the demand values by the multiplication factor (multi).

Convert the loading OD flow matrix into a vector (**od_load**).

End of odread

B.13. Subroutine odclose_files ( )

Function: To close all the files associated with the OD-flow estimation modules.

B.14. Subroutine od_convert (jcol, nod, noz, idep, norg, ndest)

Function: This is an auxiliary function called from different subroutines to calculate the departure interval, origin and destination zone numbers given the consecutive time-dependent OD pair number, that is:

Given jcol compute (idep, norg, ndest)

jcol:	is the consecutive time-dependent OD pair number
nod:	number of OD pairs in the network
noz:	number of origin zones in the network
idep:	departure time interval
norg:	origin zone number
ndest:	destination zone number

B.15. Subroutine od_convert_rev (jcol, nod, noz, idep, norg, ndest)

Function: This is an auxiliary function called from different subroutines. This subroutine does the reverse function of **od_convert,** that is it calculates the consecutive time-dependent OD pair number given the departure interval, the origin and destination numbers:

Given (idep, norg, ndest) compute jcol

The set of input variables are the same as **od_convert**.

APPENDIX C. ESTIMATION USING NON-LINEAR OPTIMIZATION METHOD

The set of **Deriv** subroutines are developed to estimate the time-dependent OD demand flows when the derivatives of the link-flow proportions are included in the formulation (the bi-level non-linear optimization method, Bi-NLP). This process is activated when 'deriv_flag' in the 'System.dat' file is set to one and is run after the last iterative run of OD- flow estimation using the bi-level Generalized Least-Squares (Bi-GLS) method.

Several of the I/O files and arrays and variables are used jointly in GLS and NLP OD-flow estimation.

C.1. List of subroutines

Main subroutine **Deriv_main ( )**
Subroutine **Deriv_save_org** ( )
Subroutine **Deriv_dist_path** ( )
Subroutine **Deriv_gen_veh** (jcol)
Subroutine **Deriv_insert_veh** (jj_org)
Subroutine **Deriv_simulate** ( )
Subroutine **Deriv_calculate** (jcol)
Subroutine **Deriv_mult** ( )

Subroutine **Deriv_QSE** ( )

Subroutine **Deriv_QSE_coeff** (a, b, g, d, f, df, isize, IGJ)

Subroutine **Deriv_solve_LSE** (a, b, x, isize, IGJ)

Subroutine **odwrite** ( )

Subroutine **odclose_files** ( )

For the description of **odwrite** and **odclose_file** see the description of Bi-GLS OD-flow estimation modules in Appendix B.

C.2. List of input files:

STATE_OBS.888:	File containing the real-world traffic flow observations.
System.DAT:	File containing parameters controlling the execution of the program and the OD-flow estimation procedure.
Fort.4242	The last estimated time-dependent OD demand table by the Bi-GLS estimation method.

(Link-flow proportion values are calculated in the simulator and passed as an array)

C.3. List of output files:

Fort.4242 File containing the final values of the estimated time-dependent OD flow values.

C.4. List of intermittent input/output files:

These files function both as input and as output in the process.

Fort.97 Vehicles trip attributes file used as the input to the simulator.

Fort.98 Vehicles path information file used as the input to the simulator.

C.5. Parameters:

nveh_add_path: Incremental number of vehicles added to any distinct path between every time-dependent OD pair (TD-OD pair) for calculating the derivative of link-flow proportions with respect to demand. This variable is set in System.dat file.

nveh_add_od: The number of vehicles added to each TD-OD pair and is equal to nveh_add_path multiplied by the number of distinct paths between that TD-OD pair.

itedex_org: The original number of iterations in the simulator (rhmuc).

iter_deriv_sim:	Number of iterative simulation runs with augmented vehicle files to calculate the derivatives of link-flow proportions with respect to demand. This variable is set in the System.dat file.
precision:	The accuracy required for convergence in solving the set of quadratic simultaneous equations. This variable is set in the System.dat file.
max_iter:	Maximum number of iterations in solving the set of quadratic simultaneous equations if precision is not achieved earlier. This variable is set in the System.dat file.
no_of_origins:	Number of origin zones.
no_of_destinations:	Number of destination zones.
nod:	Number of OD pairs in the network.
depart_interval:	The length of aggregate departure intervals in minutes.
obs_interval:	The length of the observation intervals during which the traffic volumes on links are accumulated and reported.
no_obs_dep:	Number of observation intervals in each aggregate departure interval.
nth_od_max:	Number of aggregate departure time intervals in an OD-flow estimation period.
niobs_max:	Number of observation intervals in an OD-flow estimation period.
nlink_w_detector:	Number of links which have detectors (with flow measurements).
igammaj:	Total number of OD flow elements that should be estimated (nth_od_max * nod).

nints_load:	Number of loading period aggregate departure intervals before the start of the estimation period.
startload:	Starting time of the loading period before the estimation period. This variable is equal to stagest in the rolling horizon implementation of the simulator.
starttime:	Starting time of the estimation period.
endtime:	End of the estimation period.
rload_length:	Length of the loading period.
od_min:	Minimum number of vehicular trips between any OD pairs during each aggregate departure interval.
od_max:	The maximum number of vehicular trips between any OD pairs during each aggregate departure interval.
deriv_flag:	1 if the link-flow proportion derivatives w.r.t. demand are included in the OD estimation formulation (bi-level NLP estimation), 0 otherwise.

C.6. Primary arrays:

C.6.1. Allocatable arrays

deriv_link_prop (i, j, k): Derivative of link-flow proportion with respect to demand flows. The index i is the consecutive time-dependent OD pair with respect to which the derivate is obtained. The other indices represent the element at row j and column k of the link-flow proportion matrix.

no_dist_path (j): Number of distinct paths between each time-dependent OD (TD-OD) pair j.

tot_vol (j): Total number of vehicular flows between each TD-OD pair j.

vol_path_dist (j, k): Vehicular volume between each TD-OD pair j, using the k^{th} distinct path.

rep_veh (j, k): The representative vehicle (in the vehicle file) taking the k^{th} distinct path between TD-OD pair j.

cum_freq (k): Cumulative frequency distribution of number of vehicles traversing each distinct path k between any TD-OD pair.

rank (i): The rank of the newly generated vehicle i among the vehicles sorted based on their trip start time and generation link (the sorting is required to add the vehicle to the vehicle files).

isec_org (j): Matrix to save the original **isec** matrix from the last GLS OD-flow estimation run before incremental change of demand flows for calculating the derivatives. This vector represents the link number on which vehicle j starts its trip.

veh_class_org (j): The original class of vehicle j in the last GLS OD-estimation run.

jdest_org (j): Original destination of vehicle j.

stime_org (j): Trip start time of vehicle j.

jpath_org (j): The original path of vehicle j.

Deriv_A (i, j, k): An auxiliary matrix to store the results of matrix multiplication.

Deriv_B (i, j): An auxiliary matrix to store the results of matrix multiplication.

Deriv_g (i): An auxiliary matrix to store the results of matrix multiplication.

link_prop_lt (i, j): Two dimensional link-flow proportion matrix, that is the proportion of demand flow between OD pair (o, d) starting its trip at departure interval τ which contribute to flows on link l, during observation interval t. The row i represents (l,t) combination and the column j represents the (τ, o, d) combination.

link_prop_org (i, j): The matrix to save the original link-flow proportion matrix from the last GLS OD-estimation run before incremental change of demand flows to find the derivatives.

link_prop_load (i, j): The link-flow proportion matrix associated with flows observed on links during the estimation period but having departed during loading period.

odvec(i): A column vector of time-dependent OD flows (to be estimated).

odvec_load(i): A column vector of time-dependent OD flows pertaining to the loading period.

odarr(i, j, k): The estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination zone.

odarr_old(i, j, k): The previous estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination zone.

odarr_load(i, j, k): The estimated time-dependent OD flows pertaining to the loading period (for rolling-horizon implementation or non-zero initial conditions), i is the departure interval number, j is the origin zone and k is the destination zone.

flow_diff(i): Difference between the link-flow observations and the estimated link flow volumes from simulations based on the estimated OD flows.

C.6.2. Permanent arrays:

observation_org(i): Vector of observed time-dependent link flow volumes.

observation(i): Vector of estimated time-dependent link flow observations that is due to the OD flows departing within the estimation period.

jlink(irow): The link number associated with row number irow in the observation vector.

jobst(irow): The observation interval associated with row number irow in the observation vector.

The two vectors **jlink** and **jobst** are used for bookkeeping to keep track of the links and observation intervals with available flow measurements. Therefore, the links and observation intervals with flow measurements are not required to be sequential.

link_flow(i): Estimated link flow volumes based on the estimated OD flows resulting from the simulation run.

flow_minus(i): The estimate of the portion of link volume that is due to the OD flows initiated during the loading period. This is deducted from the observed volumes to estimate the portion of flow due to the OD flows departing during the estimation period.

C.7. Main subroutine Deriv_Main ( ):

Function: This subroutine calls the pertaining subroutines for OD-flow estimation when the derivatives of link-flow proportion with respect to OD flows are included in the formulation of the problem (bi-level non-linear, Bi-NLP, estimation).

Pseudo Code:

Start of Deriv_main

Store the original number of vehicles in the network (from the last GLS OD-estimation run) in jj_org.

Store the original number of iteration runs for MUC simulation in itedex_org

Set itedex equal to iter_deriv_sim (the number of iteration of simulation in Bi-NLP estimation).

Set realdm equal to zero (reading from vehicle files in the simulator)

Call Deriv_save_org ( )

Call Deriv_dist_paths ( )

Do jcol = 1, igammaj (total number of Time-Dependent OD pairs)

Call Deriv_gen_veh (jcol)

If no vehicle is added (there was no distinct path between the two TD-OD pairs), *next* TD-OD pair

else

Call Deriv_insert_veh (jj_org)

Call Deriv_simulate ( )

Call Deriv_calculate (jcol)

Endif

Enddo

Call **Deriv_mult** ( )

Call **Deriv_QSE** ( )

Call **odwrite** ( )

Call **odclose_files** ( )

Set realdm back to 1 (reading from OD demand file in the simulator)

Set itedex (the number of iteration in the simulator) back to its original value.

End of Deriv_main

C.8. Subroutine Deriv_save_org ()

Function: This subroutine saves the original matrices and vectors from the last GLS OD-flow estimation run and reads the last estimated time-dependent OD flow values.

Pseudo Code:

Start of Deriv_save_org

- Initialize **Deriv_link_prop** matrix.
- Save the latest link-flow proportions in **link_prop_org**.
- For all vehicles in the network, save the original starting link, destination zone, starting time and path of the vehicle.
- De-allocate **jpath** (path of vehicles) from memory to be able to use it in the simulation runs.
- Open the estimated OD flow file in the last iteration (or the initial demand file, if it is the first run).
- Compute the variables pertaining to the estimation period.
- Read the OD flows of the loading period and the last estimated OD flows of the estimation period.
- Convert the time-dependent OD matrix into a vector and save it in **odvec** and **odvec_load**.

End of Deriv_save_org

C.9. Subroutine Deriv_dist_paths ()

Function: This subroutine finds the number of existing distinct paths between each time-dependent OD (TD-OD) pair and computes the number of vehicles traversing each path and the total number of vehicles between each TD-OD pair.

Local arrays:

id_path (j): The ID of the path of vehicle j. The path ID is calculated by:

$$id\_path = \sum_{k=1}^{no.\ of\ nodes\ in\ the\ path} k \times node(k, path)$$

id_path_dist(i, k): The ID of k^{th} distinct path between TD-OD pair i

Pseudo Code:

Start of Deriv_dist_paths

Initialize the arrays for all TD-OD pairs

Do i = 1 *To* number of all vehicles in the network (in the last run of GLS estimation)

Find vehicle i's consecutive TD-OD pair number according to its departure time, origin and destination

Find path ID of vehicle i using the relation

path_ID = SUM (k $\times$ number of k^{th} node along the vehicle path)

If similar path for this TD-OD pair is already marked (stored in the **id_path_dist** array) *then*

Increment the traffic volume between the TD-OD pair which uses this distinct path by one

Increment the total traffic volume between the TD-OD pair by one check the next vehicle (in the *do loop*)

otherwise

Increment the number of distinct paths between the TD-OD pair, **no_dist_path**, by one

Increment the total traffic volume between the TD-OD pairs, **tot_vol**, by one.

Increment the traffic volume along the distinct path, **vol_path_dist**, by one

Store the path's ID number in the list of distinct paths' ID numbers for this TD-OD pair

Record the representative vehicle number for this distinct path

endif

enddo (next vehicle)

End of Deriv_dist_paths

C.10. Subroutine Deriv_gen_veh (jcol)

Function: This subroutine is called within a loop for each TD-OD pair. It assigns the attributes of the newly generated vehicles randomly (proportional to the frequency of utilization of each path) and prepares them to be inserted into the vehicle files by sorting them according to their trip starting time and generation link. The newly generated vehicles are used to find the derivatives of link-flow proportions with respect to demand.

Pseudo Code:

Start of Deriv_gen_veh

> Initialize **jpath_new**, the paths for newly generated vehicles.
>
> Set total number of vehicles that should be added to the flow between the TD-OD pair.
>
> If there was no distinct path between this TD-OD pair (volume equal to zero), exit and assume the derivatives are equal to zero.
>
> Set the cumulative frequency distribution, **cum_freq**, of number of vehicles traversing each path.
>
> *Do* jveh = 1, number of added vehicles
>
>> Assign the path of newly generated vehicle (jveh) randomly and proportionally to the frequency of usage of each distinct path.
>>
>> Set the generation link of the vehicle the same as the generation link of the vehicle representing the path for this TD-OD pair.
>>
>> Set the destination zone, and vehicle class and path of the vehicle the same as the representative vehicle.

Set the trip start time of the vehicle randomly. (The starting time should be within the departure time of the TD-OD pair, but should be specified in finer units of the simulation interval, say 0.1 minute.)

Sort the newly generated vehicles based on their starting time and starting link number using the modified bubble sort algorithm. (Instead of swapping all the attributes of the vehicles, the **rank**s of the vehicles are swapped.)

End of Deriv_gen_veh

C.11. Subroutine Deriv_insert_veh (jj_org)

Function: This subroutine is called for each TD-OD pair within the main loop of **Deriv_main** subroutine. It inserts the newly generated vehicles into the array of original vehicles, obtained in the last GLS estimation run according to their trip start time and generation link and writes the augmented vehicle attributes to the vehicle and path files.

Pseudo Code:

Start of Deriv_insert_veh

    Do nveh = 1, total number of added vehicles

        Scan through the start time array, **stime_org**, of original vehicles (from the last GLS estimation run) and their generation link array, **start_link**.

        Insert the new vehicle based on its start time and generation link in the array of vehicles attributes: **isec, vehclass, jdest, stime, xpar, jpath_tmp**.

        For other vehicles, copy the attributes from the original arrays of attributes.

    Enddo

    Write the augmented vehicles' attributes to vehicle and path files, output/fort.97 and output/fort.98.

End of Deriv_insert_veh

C.12. Subroutine Deriv_simulate ()

Function: This subroutine simulates the vehicles in the network and is called within the main loop of **Deriv_main** for each TD-OD pair. By setting realdm equal to zero, in the **Deriv_main** subroutine, the simulator reads the information from the augmented vehicle and path files with the newly generated vehicles.

The output of the simulator is a new set of link-flow proportion matrix that is used to calculate the derivative of link-flow proportions with respect to demand.

Pseudo Code:

> Pseudo code of this subroutine is identical to the main rhmuc_main subroutine explained in DYNASMART-P user's guide.

C.13. Subroutine Deriv_calculate ()

Function: This subroutine is called from **Deriv_main** for each TD-OD pair and computes the derivative of link-flow proportion with respect to the change in the pertaining TD-OD pair flow.

Pseudo Code:

Start of Deriv_calculate

Do i = 1, last row of link-flow proportion matrix (no. of obs. × no. of links)

Do j = 1, last column of link-flow proportion matrix (no. of TD-OD pairs)

Subtract the original link-flow proportion from the new link-flow proportion (resulting from simulating the augmented vehicle file).

Divide the difference in link-flow proportion by the total number of augmented vehicles and store the results in **Deriv_link_prop** array.

Enddo

Enddo

End of Deriv_calculate

C.14. Subroutine Deriv_mult ()

Function: This subroutine is used for matrix multiplication by preparing a set of auxiliary matrices used in Bi-NLP estimation.

Pseudo Code:

Start of Deriv_mult

> Find the estimate of traffic flows due to the OD flows departing during the loading period and store it in the **flow_minus** array.
>
> Find the net flow by deducting the **flow_minus** from the observed flow on the links and store the results in **observation** vector.
>
> Initialize the auxiliary arrays **Deriv_A**, **Deriv_B** and **Deriv_g**.
>
> Compute Deriv_A as $((\text{Deriv\_link\_prop})^T \times \text{link\_prop\_lt}))$
>
> Compute Deriv_B as $[((\text{observation})^T \times (\text{Deriv\_link\_prop}) - (\text{link\_prop\_lt})^T \times (\text{link\_prop\_lt}))]$
>
> Compute Deriv_g as $[(\text{link\_prop\_lt})^T \times (\text{observation})]$

End of Deriv_mult

C.15. Subroutine Deriv_QSE ()

Function: This subroutine is the main module for solving the set of simultaneous quadratic equations. It calls two subroutines, **Deriv_QSE_coeff** and **Deriv_solve_LSE**, and saves the best results based on the minimum value of the root mean square of corrections in consecutive runs.

Pseudo Code:

Start of Deriv_QSE

Save the last vector of estimated OD by GLS method in the **odvec_org** vector.

Do icounter = 1, max_deriv_iter (max. number of iterations to solve the QSE)

Call **Deriv_QSE_coeff**. This subroutine takes **Deriv_A**, **Deriv_B** and **Deriv_g** as inputs and results in **q_prime** and **c_prime** (the coefficients of a set of linearized simultaneous equations) as output.

Call **Deriv_solve_LSE**. This subroutine solves a set of linear simultaneous equations. It solves for the amount of adjustment to the estimated values of unknowns and stores them in **tmp_res**.

For all TD-OD pairs, make the adjustment to the assumed values as:

(**od_vec** = **od_vec** + **tmp_res**).

Compute the total root mean square of corrections (RMSC1) made in solving the linearized set of quadratic simultaneous equations.

Compute RMSC2, which is the difference in the computed OD flow values in this iteration from its initial values when starting the Bi-NLP method (the last run of GLS estimation method).

Find the best solution in terms of the minimum achieved RMSC1 and RMSC2.

If RMSC1 or RMSC2 is less than the required 'precision' exit the *do* loop.

Enddo

Substitute the results in **odvec** vector.

End of Deriv_QSE

C.16. Subroutine Deriv_QSE_coeff (a, b, g, d, f, df, isize, IGJ)

Function: This subroutine computes the RHS coefficients and the LHS constants of a set of quadratic simultaneous equations that are linearized using Taylor series expansion around the OD-flow values in the current iteration.

The set of simultaneous quadratic equations to be solved is:

$$\boldsymbol{f}(\boldsymbol{d}) = \sum_{\boldsymbol{m}=1}^{\boldsymbol{\Pi J}} \sum_{\boldsymbol{n}=1}^{\boldsymbol{\Pi J}} \boldsymbol{a}_{\boldsymbol{mn}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} \boldsymbol{d}_{\boldsymbol{m}} \boldsymbol{d}_{\boldsymbol{n}} - \sum_{\boldsymbol{m}=1}^{\boldsymbol{\Pi J}} \boldsymbol{b}_{\boldsymbol{m}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} \boldsymbol{d}_{\boldsymbol{m}} - \boldsymbol{g}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} = 0 \qquad \forall (\nu, o, d) \in (\Gamma, \mathrm{I}, \mathrm{J})$$

Using the Taylor series expansion, the above set of quadratic equations is converted to a set of linearized simultaneous equations with unknowns as the deviation from the current solution.

The negative of ***f(d)*** at any given demand vector comprises the right-hand side (RHS) constants of the set of linearized simultaneous equations.

The partial derivative of ***f(d)*** with respect to demand flows, ***d***, comprises the left-hand side (LHS) coefficients of the set of linearized simultaneous equations:

$$\boldsymbol{df}[(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d}),\boldsymbol{m}] = \frac{\partial \boldsymbol{f}(\boldsymbol{d})}{\partial \boldsymbol{d}_{\boldsymbol{m}}} = \sum_{\boldsymbol{n}=1}^{\boldsymbol{\Pi J}} (\boldsymbol{a}_{\boldsymbol{mn}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})} + \boldsymbol{a}_{\boldsymbol{nm}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}) \boldsymbol{d}_{\boldsymbol{n}} - \boldsymbol{b}_{\boldsymbol{m}}^{(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d})}$$

$$\forall [(\boldsymbol{\nu},\boldsymbol{o},\boldsymbol{d}),\boldsymbol{m}] \in [(\boldsymbol{\Gamma},\boldsymbol{I},\boldsymbol{J}),(\boldsymbol{\Gamma},\boldsymbol{I},\boldsymbol{J})]$$

where

a is the auxiliary **Deriv_A** matrix

b is the auxiliary **Deriv_B** matrix

g is the auxiliary **Deriv_g** vector

d is the given OD flow values in this iteration (**odvec**)

f is the vector of RHS values of the set of linearized simultaneous equations

df is the matrix of LHS coefficients of the set of linearized simultaneous equations

isize is the maximum number of TD-OD pairs

IGJ is the number of TD-OD pairs in the problem (the same as igammaj)

Pseudo Code:

Start of Deriv_QSE_coeff

Compute the RHS coefficients of the linearized set of simultaneous equations, **df**(m,i) for all m and i as the combinations of all TD-OD pairs.

Compute the LHS constants of the linearized set of simultaneous equations, **f**(m) for all TD-OD pairs (m).

End of Deriv_QSE_coeff

C.17. Subroutine Deriv_solve_LSE (a, b, x, isize, IGJ)

Function: This subroutine solves a set of linear simultaneous equations using the Gaussian elimination method. The result is the last column of the transformed RHS matrix.

The arguments of the subroutine are defined below:

a The vector of RHS coefficients to the set of linearized simultaneous equations

b The matrix of LHS coefficients of the set of linearized simultaneous coefficients

x The solution to the set of linearized simultaneous equations, which consists of the vector of deviation from the current estimate of OD flows or the amount of adjustment that should be applied to the current solution.

isize The maximum number of TD-OD pairs used to size the arrays

IGJ The number of TD-OD pairs in the problem (the same as igammaj)

Pseudo Code:

Start of Deriv_solve_LSE

Augment vector **b** to the last column of matrix **a**.

Use the Gauss method to inverse the matrix **a** (making the diagonal elements equal to one).

The solution to the linearized set of equations, **x**, is the last column of the transformed matrix **a**.

End of Deriv_solve_LSE

APPENDIX D. BAYESIAN INFERENCE

The Bayesian inference is used to incorporate the estimated OD flows from traffic counts with the available historical OD tables or OD flows information obtained from other sources such as household surveys. The structure of the module is as follows.

D.1. List of subroutines

Main subroutine **Bayes_main** ( )
Subroutine **Bayes_open_files** ( )
Subroutine **odinput** ( )
Subroutine **odread** ( )
Subroutine **Bayes_prior_disp** ( )
Subroutine **Bayes_postr_disp** ( )
Subroutine **Bayes_close_files** ( )

For the description of subroutines **odinput, odread, od_conver** and **od_convert_rev** see the GLS estimation module.

D.2. List of input files:

SYSTEM.DAT:	File containing parameters controlling the execution of the program.
Output/fort.4242	The last estimated OD demand table, by GLS or non-linear OD-estimation module.
Input/prior_demand.42	The *a priori* information on OD demand table.
Input/prior_disp.dat	The *a priori* dispersion matrix of OD values.
Input/prior_abg.dat	The *a priori* distribution parameters: alpha, beta and gamma.

The traffic flow observations (***C***) are read from the memory.

D.3. List of output files:

Fort.4242	Posterior OD demand flow values.
Input/prior_disp.dat	The *a priori* dispersion matrix, if this is the first run and *a priori* information does not exist.
Input/prior_abg.dat	The *a priori* distribution matrix, if this is the first run and *a priori* information does not exist.
Ouput/posterior_disp.701	The posterior dispersion matrix.
Output/posterior_abg.702	The posterior distribution parameters: alpha, beta and gamma.

D.4. Parameters:

alpha:	*A priori* parameter of OD-flow distribution
betta:	*A priori* parameter of OD-flow distribution
gamma:	*A priori* parameter of OD-flow distribution
alpha_postr:	Posterior alpha
beta_postr:	Posterior beta
gamma_postr:	Posterior gamma
no_of_origins:	Number of origin zones
no_of_destinations:	Number of destination zones
nod:	Number of OD pairs in the network (I × J).
depart_interval:	The length of aggregate departure intervals in minutes
obs_interval:	The length of observation intervals during which the flow on links are accumulated and reported
no_obs_dep:	Number of observation intervals in each aggregate departure interval
nth_od_max:	Number of aggregate departure intervals in an OD-estimation period (Γ)
niobs_max:	Number of observation intervals in an estimation period
igammaj:	Total number of OD flow elements that should be estimated (nth_od_max × nod or Γ×I×J)
nints_load:	Number of aggregate departure intervals used for loading the network before the start of the estimation period
startload:	Start time of the loading period before the estimation period. This variable is equal to stagest in rolling horizon implementation of the simulator.

starttime: Start time of the estimation period.

endtime: End of the estimation period.

od_min: The minimum number of vehicular trips between any OD pairs during each aggregate departure interval.

od_max: The maximum number of vehicular trips between any OD pairs during each aggregate departure interval.

bayes_flag: 1 if Bayesian inference method is used, 0 otherwise.

D.5. Primary arrays:

D.5.1. Allocatable arrays

PSI (i, j): Prior dispersion matrix (Ψ).

PSI_postr (i, j): Posterior dispersion matrix (Ψ^*).

link_prop_lt (i, j): Two dimensional link-flow proportion matrix, that is the proportion of demand flow between OD pair (o, d) that start their trip at departure interval τ which contribute to flows on link l, during observation interval t. The row i represents (l,t) combination and the column j represents the (τ, o, d) combination ($\boldsymbol{P}$).

link_prop_load (i, j): The link-flow proportion matrix associated with flows departing during loading period but observed on links during the estimation period.

odvec(i): A column vector of time-dependent OD flows (μ).

odvec_load(i): A column vector of time-dependent OD flows pertaining to the loading period.

odarr(i, j, k): The estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination zone.

odarr_old(i, j, k): The previous estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destinationzone.

odarr_load(i, j, k): The estimated time-dependent OD flows pertaining to the loading period (for rolling-horizon implementation or non-zero initial conditions), i is the departure interval number, j is the origin zone and k is the destination zone.

D.5.2. Permanent arrays:

no_row_dep (j): Number of observations (rows) in aggregate departure interval j. (due to detector failure, they might be different in different departure intervals.)

observation_org(i): Vector of observed time-dependent link flow volumes.

observation(i): Vector of estimated time-dependent link flow observations that is due to OD demand flows departing during estimation period (***C***).

jlink(irow): Link number associated with row number irow in the observation vector.

jobst(irow): The observation interval associated with row number irow in the observation vector.

The two vectors **jlink** and **jobst** are used for bookkeeping to keep track of the links and observation intervals with available flow measurements. Therefore, the links and observation intervals with flow measurements are not required to be sequential.

link_flow(i): Estimated link flow volumes given the estimated OD flows obtained from the simulation run.

flow_minus(i): The estimate of the portion of link volume that is due to the OD flows initiated during the loading period. This is deducted from the observed volumes to estimate the portion of flow due to the OD flows departing during the estimation period.

D.6. Main subroutine Bayes_main ( )

Function: This is the main subroutine of Bayesian inference module. It calls different subroutines according to the control settings in the system.dat file.

Pseudo Code:

Start of Bayes_main

Call **Bayes_open_files** ( ) to open needed files.

Call **odinput** ( ) to read and calculate the constant values.

Call **odread** ( ) to read **odvec_load** with **link_prop_load** already in memory

Compute the estimate of the net traffic volume of the observed flow that are supposedly due to the OD flows during the estimation period.

If Bayes_flag = 0, Call **Bayes_prior_disp** ( ). That is, if it is not a Bayesian inference run, calculate the *a priori* parameters and matrices.

If Bayes_flag = 1, Call **Bayes_postr_disp** ( ). That is, if it is a Bayesian inference run, update the *a priori* information based on the sample information from the link observation.

Call **Bayes_close_files** ( ) to close the open files.

End of Bayes_main

For description of **odinput** and **odread** refer to GLS estimation module.

D.7. Subroutine Bayes_open_files ( )

Function: This subroutine opens the required input and output files based on the settings in system.dat file.

Pseudo Code:

Start of Bayes_open_files

Open output/fort.4242 as I/O unit 4242 for input and output

If Bayesian inference run *then*

Open the historical data (*a priori* information file) "input/prior_demand.42" as I/O unit number 42 as input.

Open "output/posterior_disp.701" as unit number 701 and "output/posterior_abg" as unit number 702 as the output files.

Open "input/prior_disp.dat" as I/O unit number 601 containing the *a priori* dispersion matrix as input.

Open "input/prior_abg.dat" as I/O unit number 602 containing the *a priori* distribution parameters as input.

Else (that is if not a Bayesian inference run)

Open "input/prior_disp.dat" as I/O unit number 601 as output.

Open "input/prior_abg.dat" as I/O unit number 601 as output.

(If it is not a Bayesian inference run, the above files will be used as output to write the prior parameters that can be used in subsequent Bayesian inference runs.)

Endif

End of Bayes_open_files

D.8. Subroutine Bayes_prior_disp ( )

Function: This subroutine computes dispersion matrix, alpha, beta and gamma as the *a priori* parameters when *a priori* information does not exist.

Pseudo Code:

Start of Bayes_prior_disp

Compute the number of zero rows in link-flow proportion matrix for each departure interval (link-flow proportion is zero where observation time is before the departure time).

Multiply link-flow proportion matrix and its transpose. Save in **PT_P** matrix.

Invert **PT_P** matrix.

Save prior dispersion matrix (inverted **PT_P**) in "input/prior_disp.dat" file.

Compute inverse of gamma as

$$(\gamma^{-1})^{*} = \frac{\boldsymbol{C}^{*T}\boldsymbol{C}^{*} - \boldsymbol{C}^{*T}\boldsymbol{P}^{*}(\boldsymbol{P}^{*T}\boldsymbol{P}^{*})^{-1}\boldsymbol{P}^{*T}\boldsymbol{C}}{(\mathrm{L.T} - \Gamma.\mathrm{I.J})}$$

Set alpha as (L.T + 2)/2. (L.T is the total number of observation rows.)

Set beta as (alpha – 1)/gamma

Output alpha, beta and gamma to "input/prior_abg.dat" file.

End of Bayes_prior_disp

D.9. Subroutine Bayes_postr_disp ( )

Function: This subroutine computes the posterior dispersion matrix, alpha, beta and gamma. It also updates the *a priori* OD demand table based on the estimated OD demand flows from traffic flow observations.

Pseudo Code:

Start of Bayes_postr_disp

Read *a priori* demand information (of the loading period and the estimation period) from file "input/prior_demand.42".

Convert the OD demand matrix into OD demand vector (**odvec**).

Read the *a priori* alpha, beta and gamma from "input/prior_abg.dat".

Read from file "input/prior_disp.dat". Determine the rows pertaining to the OD-estimation period and assign it to **PSI** matrix (dispersion matrix).

Set alpha_postr = (total number of observations + 2)/2

Compute posterior dispersion matrix as

$\boldsymbol{\Psi}^* = (\boldsymbol{P}^T \boldsymbol{P} + \boldsymbol{\Psi})\gamma$

Find the inverse of $\boldsymbol{P}^T \boldsymbol{P} + \boldsymbol{\Psi}$.

Find the posterior OD demand flows ($\boldsymbol{\mu}^*$) as

$$\mu^* = (P^T P + \Psi)^{-1} (P^T C + \Psi \mu).$$

Write the posterior dispersion matrix to "output/posterior_disp.701".

Write the posterior alpha, beta and gamma to "output/posterior_abg.702".

Convert the **odvec** array to a three-dimensional TD-OD array.

Open the output /fort.4242 file (since it is closed in **odread** subroutine, in Bayes_main)

Check the posterior OD flow values against the maximum and minimum bounds.

Write the TD-OD pairs to “output/fort.4242” file.

End of Bayes_postr_disp

APPENDIX E. FINDING THE STATISTICS OF OD-FLOWS ESTIMATION

E.1. List of subroutines

Main subroutine **STAT_OD_main ( )**

Subroutine **STAT_open_files** ( )

Subroutine **STAT_input** ( )

Subroutine **STAT_read** ( )

Subroutine **STAT_cal** ( )

Subroutine **STAT_close_files** ( )

E.2. List of input files:

STATE_OBS.888:	File containing the real world traffic flow observations.
Actual_demand.42	File containing the actual demand, if it exists. It will be used to find the measure of performance of OD-estimation method.
Initial_demand.42	Initial guess of time-dependent OD table (for the first iteration of OD-flow estimation).
Fort.4242	The estimated OD table in each iteration.

E.3. List of output files:

Fort.555	File containing the statistics of OD-flow estimation of each iteration.

E.4. Parameters:

no_of_origins:	Number of origin zones.
no_of_destinations:	Number of destination zones.
nod:	Number of OD pairs in the network.
depart_interval:	Length of aggregate departure intervals in minutes.

obs_interval: Length of observation intervals during which the flow on links are accumulated and reported.

no_obs_dep: Number of observation intervals in each aggregate departure interval.

nth_od_max: Number of aggregate departure intervals in an estimation period.

niobs_max: Number of observation intervals in an estimation period.

nlink_w_detector: Number of links which have detectors (with link flow measurements).

igammaj: Total number of OD flow cells that should be estimated (nth_od_max × nod).

nints_load: Number of aggregate departure intervals used for loading the network before the start of the estimation period.

startload: Start of the loading period before the estimation period. This variable is equal to stagest in rolling horizon implementation of the simulator.

starttime: Start time of the estimation period.

endtime: End of the estimation period.

rload_length: The length of the loading period.

w_r: Window ratio, the fraction of estimation period that the estimated OD is deemed to be final, recommended to be one.

odest_flag: 1 if the run is for OD estimation,
0 otherwise (simulation or planning run).

amsq_od_n: Root mean square error in the estimated OD-flows for each aggregate departure interval (if an assumed actual time-dependent demand table exists).

fmsq_od_n: Root mean square error in the estimated time-varying flows for each aggregate departure interval.

E.5. Primary arrays:

E.5.1. Allocatable arrays

odvec_load(i): A column vector of time-dependent OD flows pertaining to the loading period.

odarr(i, j, k): The estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination.

odarr_old(i, j, k): The previous estimated time-dependent OD flows, i is the departure interval number, j is the origin zone and k is the destination zone.

odarr_load(i, j, k): The estimated time-dependent OD flows pertaining to the loading period (for rolling-horizon implementation or non-zero initial conditions), i is the departure interval number, j is the origin zone and k is the destination zone.

flow_diff(i): Difference between the link-flow observations and the estimated link flow volumes obtained from the simulation given the estimated OD flows.

odo (i, j, k): The (assumed) actual OD flow for departure time i, from origin j to destination k. This matrix is only used for hypothetical cases and for testing of the OD-flow estimation method.

anumber (i, j, k): The difference in the actual and estimated demand value for departure interval i, origin j and destination k.

amsq(j): Sum square-error of OD flow values.

rn_ck(i): Sum of estimated OD flow values.

rn_od(i): Sum of actual OD flow values.

no_row_dep(i) Number of flow observations (number of rows in **observation** matrix) in each aggregate departure interval.

rn_obs(j): Total sum of observed (measured and reported) flows on the links.

rn_flow(j): Total sum of the estimated link-flows on the links that have flow measurement sensors.

fmsq(j): Sum square-error of link-flow estimation.

E.5.2. Permanent arrays:

observation_org(i): Vector of observed time-dependent link flow volumes.

observation(i): Vector of estimated time-dependent link flow observations that is due to OD flows departing during the estimation period.

jlink(irow): The link number associated with row number irow in the observation vector.

jobst(irow): The observation interval associated with row number irow in the observation vector.

The two vectors **jlink** and **jobst** are used for bookkeeping to keep track of the links and observation intervals with available flow measurements. Therefore, the links and observation intervals with flow measurements are not required to be sequential.

link_flow(i): Estimated link flow volumes given the estimated OD flows.

flow_minus(i): The estimate of the portion of link volume that is due to the OD flows initiated during the loading period. This is deducted from the observed volumes to estimate the portion of flow due to the OD flows departing during the estimation period.

E.6. Main subroutine STAT_OD_main ( ):

Function: This subroutine calls the pertaining subroutines for calculating the statistics of OD-flow estimation in the Bi-GLS, Bi-NLP and Bayesian inference method.

Pseudo Code:

Start of STAT_OD_main

Call **STAT_open_files** ( ) to open files needed for OD-flow estimation statistics calculation.

Call **STAT_input** ( ) to calculate the basic constants of the network and the OD- flow estimation period.

Call **STAT_read** ( ) to read the required information from the pertaining files.

Call **STAT_cal** ( ) to calculate the statistics of the OD-flow estimation.

Call **STAT_close_files** ( ) to close the files.

End of STAT_OD_main

E.7. Subroutine STAT_open_files ( ):

Function: This subroutine opens the pertaining input and output files.

Pseudo Code:

Start of STAT_open_files

> Check if an assumed actual OD-flow file ("input/actual_demand.42") exists. If it does, open it as I/O unit number 424.
>
> Check if the estimated OD-flow file ("output/fort.42") exists. If it does, open it. Otherwise, it is the first run and open the file containing the initial guess of OD-flow table ("input/initial_demand.42"). Open either one as I/O unit number 42.
>
> Open "output/fort.555", which is the file containing the statistics of the estimation in each iteration, as I/O unit number 555.
>
> Open the error messages output file ("output/fort.911) as I/O unit number 911.

End of STAT_open_files

E.8. Subroutine STAT_input ( ):

Function: This subroutine calculates the constants of the estimation period and the relevant network characteristics.

Pseudo Code:

Start of STAT_input

Set the values for the start and end of the estimation period.

Read the constants of the estimated or the initial demand file.

Calculate other pertaining constants of the network characteristics and the estimation period.

End of STAT_input

E.9. Subroutine STAT_read ( ):

Function: This subroutine reads the assumed actual-demand OD table and the current estimated OD-flow table.

Pseudo Code:

Start of STAT_read

If the assumed actual demand exists, read the constants and the actual OD-flows for the loading period and the estimation period (**odo**) from the actual demand file.

Read the current estimated OD-flows or, in the first run, the initial guess of demand flows during the loading period (**odarr_load**) or estimation period (**odarr**).

(Optionally, write the observation and the estimated link flows for comparison to a temporary file ("output/link_flow.temp").

End of STAT_read

E.10. Subroutine STAT_cal ( ):

Function: This subroutine calculates the statistics of the OD-flow estimation in terms of the root mean square error in link-flow values and OD-flow values (if an assumed OD flow table exists) and writes the results to the output file "output/fort.555".

Pseudo Code:

Start of STAT_cal

If the actual demand file exists

Find the difference between the actual OD flow and the estimated values and store it in **anumber** array.

Find the sum of squares of errors in demand, sum of actual demand and the estimated demand flow values.

Endif

Store the original link-flow observation vector in **observation**.

Find the sum of link-flow observations, simulated flows and errors in link-flow estimates for each aggregate departure interval and for the whole estimation period.

If the actual demand file exists

Calculate the sum of the actual OD-flow values, the estimated values, and the sum square errors for each departure interval and over the whole estimation period.

Endif

Write the statistics of the OD-flow estimation to the output file ("output/fort.555").

End of STAT_cal

E.11. Subroutine STAT_close_files ( ):

Function: This subroutine closes all pertaining files.

Pseudo Code:

Start of STAT_close_files

    Close all the input and output files.

End of STAT_close_files

REFERENCES

Ashok K. (1996) Estimation and Prediction of Time-Dependent Origin-Destination Flows. *Ph.D. Dissertation,* Massachusetts Institute of Technology.

Bard, J. (1998) *Practical Bilevel Optimization,* Kluwer Academic Publishers, Dordrecht, The Netherlands.

Bell, M.G.H. (1991) The estimation of origin-destination matrices by constrained generalized least squares. *Transportation Res. B* 25B (1), pp 13-22.

Bell, M.G.H. (1984) Log-linear models for the estimation of origin-destination matrices from traffic counts: An approximation. *Proceeding of the Ninth International Symposium on Transportation and Traffic Theory,* The Netherlands, Delft, July, pp 451-470.

Bell, M.G.H. (1983) The estimation of an origin-destination matrix from traffic counts. *Transportation Science* 17(2), pp 198-217.

Bhat, C.R. (1997) An endogenous segmentation mode choice model with an application to intercity travel. *Transportation Science* Vol. 31, No. 1, pp 34-48.

Boyles, R.A. (1983) On the convergence of the EM algorithm. *Journal of the Royal Statistical Society*, B45, No. 1, pp 11-38.

Broemeling, L.D. (1985) *Bayesian Analysis of Linear models*, Marcel Dekker, Inc., New York.

Cascetta E., D. Inaudi and G. Marquis (1993) Dynamic estimators of origin-destination matrices using traffic counts. *Transportation Science* 27(4), pp 363-373.

Cascetta E. (1984) Estimation of trip matrices from traffic counts and survey data: A generalized least squares estimator. *Transportation Res. B* 18(4/5), pp 289-299.

Chang G.L. and X. Tao (1995) An advanced computing architecture for large-scale network O-D estimation. *Vehicle Navigation and Information Systems Conference,* Proceedings of the 6th annual meeting.

Cochran W.G. (1963) *Sampling Techniques*, 2nd Ed. Wiley, New York.

Cremer M. (1983) Determining the time dependent trip distribution in a complex intersection for traffic responsive control. *Preprint of the 4th Int. IFAC/IFIP/IFORS Conf. On Control in Transp. Systems*, Baden-Baden, Germany.

Cremer M. and H. Keller (1987) A new class of dynamic methods for identification of origin-destination flows. *Transportation Res.* 21B, pp 117-132.

Cremer M. and H. Keller (1984) A systems dynamics approach to the estimation of entry and exit O-D flows. *Proc. 9th Int. Symp. on Transp. and Traffic Theory. J. Volmuller and R. Hamerslag eds.*, VUN Science Press, Utrecht, The Netherlands.

Cremer M. and H. Keller (1981) Dynamic identification of flows from traffic counts at complex intersections. *Proc. 8th Int. Symp. Transportation and Traffic Theory,* Toronto University.

Dempster, A.P., N.M. Laird, and D.B. Rubin (1977) Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, B39, No.1, pp 1-38.

Ding Y., P.B. Mirchandani and S. Nobe (1997) Prediction of network loads based on origin-destination synthesis from observed link volumes. *Transportation Research Record,* 1607, National Research Council, Washington, D.C. pp 95-104.

Dixon M.P., L.R. Rilett (2000) Real-time origin-destination estimation using automatic vehicle identification data, *proceeding of 79th Annual Meeting Transportation Research Board,* Washington D.C.

Doan D.L., A. Ziliaskopoulos and H.S. Mahmassani (1999). An on-line monitoring system for real-time traffic management applications. *Transportation Research Record,* n1678, Nov. 1999, National Research Council, Washington D.C., USA, pp. 142-149.

Florian M., Y. Chen (1994) A coordinate descent method for the bilevel O-D matrix adjustment problem. *Preprints of the 7th IFAC/IFORS Symposium on Transportation Systems: Theory and Application of Advanced Technology.* Tianjin, china, August 24-26, pp 1029-1034.

Gur J., Hutsebout O., Kurth D., Clark M. and Castilla J. (1980) Estimation of an Origin-Destination trip table based on observed link volumes and turning movements, Volume 1. *Technical Report*, *FHWA, Report No. RD-801035.* U.S. Dept. of Transportation, Washington D.C.

Inaudi D., M.G.H. Bell, M. Maher, G. Ploss, P. Philips and M. Cremer (1991) OD information vs. traffic control, *Drive I v 1047, Final Report.*

Jayakrishnan R., H.S. Mahmassani and T.Y. Hu (1994) An evaluation toll for advance traffic information and management systems in urban networks. *Transportation Research C,* 2(3), pp 129-147.

Jeffreys M. and Norman M. (1977) On finding realistic turning flows at road junctions. *Traffic Engineering Control* 18(1), pp 19-21, 25.

Johnston J. (1972) *Econometric Methods,* McGraw-Hill, New York.

Kang, Y. (1999) Estimation and prediction of dynamic origin-destination (O-D) demand and system consistency for real-time dynamic traffic assignment, *Ph.D.* Dissertation, University of Texas at Austin.

Keller H. and G. Ploss (1987) Real-time identification of O-D network flow from counts for urban traffic control. In N.H.Gartner and N.H.M. Wilson (Eds.), *Transportation and Traffic Theory.* Elsevier Science Publishing Co. Inc., NY. Pp 267-284.

Leblanc L.J. and Farhangian K. (1982) Selection of a trip table which reproduces observed link flows. *Transportation Res.* 16B, pp 83-88.

Maher M.J. (1983) Inferences on trip matrices from observations on link volumes: A Bayesian statistical approach. *Transportation Res. B* 17B(6), pp 435-447.

Mahmassani H.S. *et al*. (2000) DYNASMART-P, Volume II, User's Guide, Technical Report STO67-85-PII, Center for Transportation Research, The University of Texas at Austin.

Mahmassani H.S., T. Hu, S. Peeta and A. Ziliaskopoulos (1993) Development and testing of dynamic traffic assignment and simulation procedure for ATIS/ATMS applications. *Technical Report DTFH61-90-R-00074-FG*, Center for Transportation Research, The University of Texas at Austin.

Mekky A. (1979) On estimating turning flows at road junctions. *Traffic Engineering Control* 20(10), pp 486-487.

Nguyen, S. (1982) Estimating origin-destination matrices from traffic flows. *1st Course in Transportation Planning Model,* ICTS, Amalfi

Nguyen S. (1977) Estimating an OD matrix from network data: A network equilibrium approach, *Publication No. 87, Centre de Recherche sur les Tranports*, Universite de Montreal, Montreal, Quebec.

Oh J. (1992) Estimation of trip matrices from traffic counts: an equilibrium approach, *Mathematics in Transport Planning and Control,* pp 35-44, Oxford Clarendon Press.

Okutani I. (1987) The Kalman filtering approach in some transportation and traffic problems. In *Transportation and Traffic Theory (Gartner N.H. and Wilson N.H.M. eds)*, Elsevier Science Publishing Co., Inc.

Ortuzar J. de D. and L.G. Willumsen (1990) *Modeling Transport.* New York: Wiley.

Peeta, S., Hani S. Mahmassani (1995a) Multiple user classes real-time traffic assignment for online operations: a rolling horizon solution framework, *Transportation Research Part C: Emerging Technologies*, Volume 3, Issue 2, April 1995, pp 83-98.

Peeta S., Hani S. Mahmassani (1995b) System optimal and user equilibrium time-dependent traffic assignment in congested networks, *Annals of Operations Research*, 60, 1995, pp 81-113.

Robillard P. (1975) Estimating the O-D matrix from observed link volumes. *Trans. Res.* 9, pp 123-128.

Sheffi Y. (1985) *Urban Transportation Networks: Equilibrium Analysis with Mathematical Programming Methods*, Prentice Hall, Englewood Cliffs.

Tavana H. and H.S. Mahmassani (2000) Estimation and application of dynamic speed-density relation using transfer functions, *Transportation Research Record*, 1710, pp 47-57.

Tobin R.L. and T.L. Friesz (1988) Sensitivity analysis for equilibrium network flows. *Transportation Science*, 22, pp 242-250.

Turnquist M., Y. Gur (1979) Estimation of trip tables from observed link volumes. *Transportation Research Record* 730, TRB, National Research Council, Washington, D.C., pp 1-6.

Van Zuylen H.J. and Willumsen L.G. (1980) The most likely trip matrix estimated from traffic counts. *Transportation Res.* 14B, pp 281-293.

Van Zuylen H.J. (1979) The estimation of turning flows on junction. *Traffic Engineering Control* 20(11), pp 539-541.

Willumsen L.G. (1984) Estimating time-dependent trip matrices from traffic counts, in *Proceeding of the Ninth International Symposium on Transportation and Traffic Theory,* VUN Science press, pp 397-411.

Xu W. and Y. Chan (1993) Estimating an origin-destination matrix with fuzzy weights. *Transportation Planning and Technology,* Vol. 17, pp 145-163.

Willumsen L.G. (1982) Estimation of trip matrices from volume counts: validation of a model under congested conditions. *PTRC Summer Ann. Meeting,* University of Warwick, July 1982.

Willumsen L.G. (1981) Simplified transport models based on traffic counts. *Transportation Research.* 10, pp 257-278.

Willumsen L.G. (1978) Estimation of an O-D matrix from traffic counts: a review. Institute for Transport Studies, Leeds University.

Wu, C.F.J. (1983) On the convergence properties of the EM algorithm. *The Annals of Statistics*, Vol. 11, No. 1, pp 95-103.

Yang H., Y. Iida, and T. Sasaki (1994) The equilibrium-based origin-destination matrix estimation problem. *Transportation Research,* 28B*,* pp 23-33.

Yang H., T. Sasaki, Y. Iida, and Y. Asakura (1992) Estimation of origin-destination matrices from link traffic counts on congested networks. *Transportation Research*, 26B, pp 417-434.

Yang H. (1995) Heuristic algorithms for the bilevel origin-destination matrix estimation problem. *Transportation Research-B,* 29B(4), pp 231-242.

Yates F. (1981) *Sampling Methods for Censurer and Surveys*, 4th Ed. Griffin, London.

VITA

Hossein Tavana was born in Tehran, Iran on July 6, 1961, the second son of Fatemeh Safari and Hamzeh Tavana. At the age of 16, he completed his studies at Kharazmi High School and entered the University of Tehran. He earned a continuous Master of Science in Civil Engineering in May 1986. From 1982 to 1987, he worked with several local consulting firms as an assistant, and later as a design engineer. In 1987-88, he fulfilled mandatory military service as an officer of the Air Force, working as a design engineer. In February 1988, he co-founded Rahpooyan Transportation Engineering Consultants and served as a Member of the Board. He continued to work there as a transportation modeler and design engineer for seven years. After coming to the United States, he earned his second Master of Science in Civil Engineering (Transportation) from Wayne State University in Detroit, Michigan in August 1996 and began his doctoral studies in Transportation Engineering at the University of Texas at Austin in the fall of the same year. In July 2000, he received his Professional Engineering license in the State of Texas, and in February 2001, he began his work with Continental Airlines in Houston, Texas as a Lead Operations Research Analyst. He has published several papers pertaining to his research and has made several presentations at local and international conferences.

Permanent Address: 3300 Sage Rd. #8303
Houston, Texas 77056

Email: Hossein@Tavana.org

The dissertation was typed by the author.