

THE MULTI-MODAL TRAFFIC
ASSIGNMENT PROBLEM

by

HEDAYAT ZOKAIE AASHTIANI

S. B. University of Technology
(1973)

S. M. Massachusetts Institute of Technology
(1976)

SUBMITTED IN PARTIAL FULFILLMENT
OF THE REQUIREMENT FOR THE
DEGREE OF

DOCTOR OF PHILOSOPHY

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1979

Signature of Author.....
Alfred P. Sloan School of Management, 1979

Certified by.....
Thesis Supervisor

Accepted by.....
Chairman, Department Committee



JUN 11 1979

LIBRARIES

THE MULTI-MODAL TRAFFIC
ASSIGNMENT PROBLEM

by

HEDAYAT ZOKAIE AASHTIANI

Submitted to the Alfred P. Sloan School of Management
on May 5, 1979 in partial fulfillment of the requirements
for the Degree of Doctor of Philosophy

ABSTRACT

Traffic equilibrium analysis has provided useful insight into the transportation planning process. Even for deterministic demand models, though, the state-of-the-art does not include any efficient approach that is applicable for general equilibrium models. Existing convex programming approaches, which are efficient and guarantee convergence, are restricted to single commodity (mode, user class) flow problems with invertible demand functions. In this thesis, we first show that convex programming approaches cannot be generalized to broader, and yet still realistic, settings. Secondly, we introduce a new approach that can be applied to multi-commodity flow problems (including multi-class, multi-modal, and destination choice user equilibrium models) with arbitrary deterministic demand functions.

The approach consists in formulating the traffic equilibrium problem as a nonlinear complementarity problem. Based upon this formulation, we propose and prove general existence and uniqueness theorems, and we develop a linearization algorithm. We also present computational results on a variety of test problems to illustrate the generality and the efficiency of the algorithm.

Thesis Supervisor: Thomas L. Magnanti

Title: Associate Professor

ACKNOWLEDGEMENTS

I am indebted and very grateful to my thesis advisor, Professor Thomas L. Magnanti, who introduced this problem to me. He has always encouraged me during the past two years, giving of his valuable time and providing new ideas and comments to make this thesis more clear.

My thanks also to my committee members, Jeremy F. Shapiro and Robert L. Simpson. Special thanks to my parents and to my people in Iran who sent me here to continue my education.

I wish to acknowledge the financial support provided by the University of Technology, Isfahan, Iran, and the U.S. Department of Transportation under Contract DOT-TSC-1058.

Finally, I would like to thank Marion Brink and Kathy Sumera for their excellent typing, and Martha Heigham for her valuable assistance.

TABLE OF CONTENTS

	<u>Page</u>
ABSTRACT	2
ACKNOWLEDGEMENT	3
TABLE OF CONTENTS	4
LIST OF FIGURES	7
LIST OF TABLES	8
CHAPTER 1: <u>INTRODUCTION</u>	9
CHAPTER 2: <u>TRANSPORTATION MODELING</u>	14
2.1 INTRODUCTION	14
2.2 COMPONENTS OF TRANSPORTATION SYSTEMS	15
2.2.1 Transportation Technology	15
2.2.2 Transportation Demand	16
2.3 MODELING TRANSPORTATION SYSTEMS	17
2.3.1 Aggregation	17
2.3.2 Deterministic and Stochastic Models	19
2.3.3 Simultaneous and Sequential Models	19
2.3.4 Route Choice	23
2.3.5 General Route Choice	24
2.3.6 Level of Service	27
2.3.7 Volume Delay Function	28
2.3.8 Examples of Volume Delay Functions	30
2.3.9 Nature of Volume Delay Functions	33
CHAPTER 3: <u>TRAFFIC EQUILIBRIUM PROBLEMS</u>	35
3.1 EQUILIBRIUM CONCEPTS	35

	<u>Page</u>
3.2 PROBLEM FORMULATION	41
3.3 EQUIVALENT NONLINEAR COMPLEMENTARITY PROBLEM	45
CHAPTER 4: <u>EXISTENCE AND UNIQUENESS OF AN EQUILIBRIUM</u>	50
4.1 INTRODUCTION	50
4.2 EXISTENCE	50
4.3 UNIQUENESS	61
CHAPTER 5: <u>COMPUTING AN EQUILIBRIUM</u>	70
5.1 INTRODUCTION	70
5.2 HEURISTIC TECHNIQUES	70
5.3 MATHEMATICAL PROGRAMMING TECHNIQUES	71
5.3.1 Fixed-point Techniques	72
5.3.2 Optimization Techniques	74
5.4 A LINEARIZATION TECHNIQUE	90
CHAPTER 6: <u>LINEARIZATION ALGORITHM AND COMPUTATIONAL RESULTS</u>	100
6.1 INTRODUCTION	100
6.2 LINEARIZATION ALGORITHM	100
6.2.1 ϵ -Approximation Solution	100
6.2.2 Starting Solution	102
6.2.3 Path Generation	103
6.2.4 Decomposition	104
6.2.5 Algorithm	105
6.2.6 Assumptions	111
6.3 COMPUTATIONAL RESULTS	111
6.4 STORAGE REQUIREMENT AND DATA STRUCTURES	133

	<u>Page</u>
6.4.1 Modified Data Structure	135
6.4.2 Out-of-Core Storage	137
REFERENCES	141
APPENDIX A	151
APPENDIX B	158

LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
2.1 Equilibrium	17
3.1 Network Configuration for Example 3.1	38
3.2 Continuous Volume Delay Functions	38
3.3 Non-continuous Volume Delay Functions	39
3.4 Decreasing Volume Delay Function	40
3.5 Negative Demand Function	48
3.6 Negative Volume Delay Function	49
5.1 Modified Network Configuration for Example 5.2	88
5.2 Linearization Scheme	97
6.1 Steps of the Linearization Algorithm	107
6.2 Network Configuration for Example 6.1	113
6.3 Auto Distribution Among the Paths for First O-D Pair	118
6.4 Network Configuration for Example 6.2	119
6.5 Network Configuration for Example 6.3	123
6.6 Sioux Falls Network Configuration	128
A.1 The Linearization Algorithm	152
A.2 The Linearization Algorithm Might Not Converge	156

LIST OF TABLES

<u>Table</u>	<u>Page</u>
6.1 Computational Results for Example 6.1	116
6.2 Equilibrium Solution for Example 6.1	116
6.3 Path Flows for Example 6.1	117
6.4 Link Flow for Example 6.2	120
6.5 Parameters θ_{ij} of Demand Function for Destination Choice Model	121
6.6 Computational Results for Destination Choice Model	122
6.7 Link Flows for Destination Choice Model	122
6.8 Trip Table for Example 6.3	123
6.9 Link Flow for Example 6.3	124
6.10 Total Travel Time for Example 6.3	126
6.11 Computational Results for Example 6.4	129
6.12 Computational Results for Example 6.5 with Fixed Demand	131
6.13 Computational Results for Example 6.5 with Elastic Demand	132
6.14 Memory Requirement to Store the Problem Data	134
6.15 Computational Results for the City of Hull with Elastic Demand for Modified Data Structure and Using Out-of-Core Storage	139

CHAPTER 1

INTRODUCTION

Traffic equilibrium models have recently become useful tools for predicting vehicular flow in congested urban areas. They can be used for planning purposes, for managing transportation systems and for improving transportation technologies to achieve better system performance.

In general, there are two directions of research related to traffic equilibrium analysis. The first direction, which is called demand prediction, is an attempt to capture the users' behavioral patterns to understand how they make decisions within the framework of existing technology and to predict their responses to future technology. Although a great deal of research of this nature has been conducted, it is still the weakest link in transportation modeling.

The second direction of this research, which is also essential for transportation planning, is predicting vehicular flow in a congested network, given the users' behavioral patterns. One model of this type now forms part of the UMTA (Urban Mass Transit Authority) Transportation Planning System [T-U1]. In their study of traffic patterns in the city of Winnipeg, Canada, Florian and Nguyen [T-F8] have shown that equilibrium models can predict link flow and traffic impedances accurately, particularly for high volume links and routes. Our purpose in this dissertation is to contribute to the second stage in equilibrium analysis which we refer to as the traffic equilibrium problem.

At present, efficient algorithms are available for traffic equilibrium

models involving:

- 1) a single mode (private vehicle traffic has been the primary application);
- 2) (elastic) demand functions between every origin-destination (O-D) pair that depend only upon the impedance or shortest travel time between that origin-destination pair;
- 3) volume delay functions for each link that depends only upon the total volume of traffic flow on that link.

Initially, Wardrop [T-W1] introduced the notion of user equilibrium for modeling urban traffic. Beckman, McGuire and Winsten [T-B1] showed that assumptions (1), (2) and (3) produce an equilibrium model that can be converted into an equivalent convex programming problem. Samuelson [O-S1] had earlier proposed a similar transformation in the context of spatially separated economic markets. Since then, several researchers have proposed algorithms for solving this convex problem (Bruynooghe, Gibert and Sakarovitch [T-B11], Bertsekas [T-B6], Defermos [T-D1-3], Dembo and Klinckemicz [T-D5], Leventhal, Nemhauser and Trotter [T-L4], Leblanc [T-L2-3], Nguyen [T-N2-6], Golden [T-G4], Florian and Nguyen [T-F6-8]).

There are a number of ways in which these models might be extended. Modeling multi-modal (for example, private vehicle and a transit mode) and multi-class user equilibrium would be the first extension. Incorporating demand functions for an O-D pair that depend upon impedance between other O-D pairs would permit destination choice to be modeled. Another extension would permit volume delay on a link to depend upon volume flow

on other links. This later extension permits modeling of traffic equilibrium with two-way traffic in one link, traffic equilibrium with right and left turn penalties, and the like.

Assumptions (1), (2) and (3) are the key to solving the traffic equilibrium problems. Some attempts have been made to generalize the convex programming approaches to solve the extended models (i.e., Defermos [T-D1] for multi-classes of users and Florian [T-F4-5] for the multi-modal case). We show that these assumptions are strong and that the convex programming approach is not, in general, applicable to the extended equilibrium model.

The goal of this research is:

- i) to formulate mathematically a general traffic equilibrium model that captures each of these modeling extensions;
- ii) to determine when the equilibrium problem can be modeled as an optimization model;
- iii) to determine conditions on the problem data that will insure that an equilibrium exists and is unique; and
- iv) to develop computational procedures for finding an equilibrium to the extended model.

To resolve these issues we formulate the problem as a nonlinear complementarity problem. By imposing very mild restrictions on the problem structure (that are always met in practice), we show that the traffic equilibrium problem always has an equilibrium solution. We introduce an algorithm, called the *linearization algorithm*, to solve the problem efficiently. Although we do not have any formal proof for the convergence of the algorithm, computational results on a variety of problems are promising.

For example, we have been able to use the algorithm to solve problems with 376 links, 155 nodes, 702 O-D pairs and elastic demand in less than 12 seconds on an IBM 370/168 to achieve 5% accuracy. We have also been able to use it to solve problems with link interactions and with complex demand relationships; problems that cannot be solved as equivalent convex optimization problems.

Recently, Hearn and Kuhn [T-H2] and Asmuth [T-A4] have made similar attempts to formulate the general traffic equilibrium problem as a fixed-point problem. Asmuth presented results similar to ours concerning existence and uniqueness issues. Kuhn illustrated the initial steps in applying fixed-point algorithms to the equilibrium problem. But no computational results have been presented to demonstrate the efficiency of the algorithms for realistically sized transportation problems.

We conclude this introduction by briefly outlining the rest of this research. Chapter 2 reviews transportation modeling in general, and summarizes the characteristics of major components of the model effort. In particular, Chapter 2 discusses issues related to the demand function for transportation services and to the volume delay function or congestion that vehicular flow imposes on a transportation system. The concept of a user-equilibrium, which is introduced in Chapter 2, is explored in more detail in Chapter 3. In Chapter 3, we formulate the equilibrium problem mathematically and introduce an equivalent nonlinear complementarity formulation for the problem.

Chapter 4 contains our main results concerning the existence and uniqueness of an equilibrium solution. After briefly reviewing existing

algorithms for the traffic equilibrium problem and their limitations, Chapter 5 contains a skeletal introduction to a new linearization algorithm.

Chapter 6 studies the linearization algorithm in more detail. This chapter illustrates the generality of the algorithm and its convergence properties by presenting computational results on a variety of small examples modeling different aspects of traffic equilibrium. Finally, Chapter 6 contains computational results for some larger examples to illustrate the efficiency of the algorithm.

CHAPTER 2

TRANSPORTATION MODELING

2.1 INTRODUCTION

Transportation modeling aims to answer the following types of questions:

- i) How do users respond to the transportation technology available to them (e.g., what is their utilization of transportation facilities, what is their movement pattern)?
- ii) How do users' utilization of the transportation system change over time (in terms of location, activity, awareness, social-economic change, and so forth)?
- iii) How do users respond to changes in the transportation system (changes in system configuration or in quality, introduction of new facilities, and so forth)?
- iv) How can planners improve an existing transportation system to capture the users' future responses to system changes?

Regardless of the kind of model that might be used to answer any of the above questions, resolution of the first question is crucial for answering any of the others. In the literature, the first type of question is referred to as *short-run-equilibrium* and the other questions are referred to as *long-run-equilibrium*. In this research we are focusing only on short-run-equilibrium models.

Let us begin by reviewing in rather general terms the essential ingredients of most transportation systems and the interactions between these ingredients.

2.2 COMPONENTS OF TRANSPORTATION SYSTEMS.

2.2.1 Transportation Technology

The transportation technology denoted by τ , determines the network structure (e.g., set of nodes, arcs, origins, destinations and modes) available to the users. Suppose that we are given a performance function \mathbb{P} that measures the performance of the system for any traffic volume V . In the transportation literature, the performance function associated with an arc is sometimes called a *volume delay function*, and the measure of performance, denoted by L , is called the *level of service*. The level of service, which might be travel time, travel cost, safety, or some function of all of these, is usually expressed in terms of disutility. L as a function of τ and V can be written as:

$$L = \mathbb{P}(\tau, V). \quad (2.1)$$

Frequently, the performance function has been referred to as a "supply" function. This terminology seems inappropriate in this context and might be misleading because of the usual economic connotation of "supply" as a response of the producers to the market. In transportation, the producers are providing transportation technology τ , though, and this is fixed for short-run equilibrium; the level of service is a measure of how the pro-

duction facilities are being used, rather than their supply. Florian [T-F5] has emphasized this distinction; see also Sheffi [T-S1].

2.2.2 Transportation Demand

The other component of transportation systems, and models that represent them, are the users who utilize the transportation technology τ by making trips. Each user in the system must choose from among a set of available alternatives (this is called decision-making process). The main components of each alternative are trip frequency (to make a trip or not), destination choice, mode choice and route choice.

Suppose that, with *perfect* communication and information, for each user i in the system we are given a function d_i that specifies the alternative choice of that user for any technology τ and level of service L , given the user's utility u_i . Also suppose that the function D specifies the traffic volume in the system: that is,

$$V = D(d_i(\tau, L | u_i) \quad \text{for all users } i). \quad (2.2)$$

This expression is called a demand relationship.

Substituting (2.1) in (2.2) we obtain

$$V = D(d_i(\tau, P(\tau, V) | u_i) \quad \text{for all } i). \quad (2.3)$$

Obviously, (2.3) can be interpreted as a *fixed point* problem with variable V in the sense that with fixed technology τ , and with given utility func-

tions u_i and decision function d_i , the right-hand side of (2.3) for any given $V = V^0$ predicts traffic volume as $V' = D$. Conditions (2.1) and (2.2) are satisfied only if $V' = V^0$.

DEFINITION 2.1: Given a transportation technology τ , the performance function \mathbb{P} , and decision function d_i for all users, any traffic volume V_E which satisfies the fixed point problem (2.3) is called an *equilibrium solution*.

Equivalently, any pair (V_E, L_E) that satisfies equations (2.1) and (2.2) is called an *equilibrium point*. Figure 2.1 illustrates this concept.

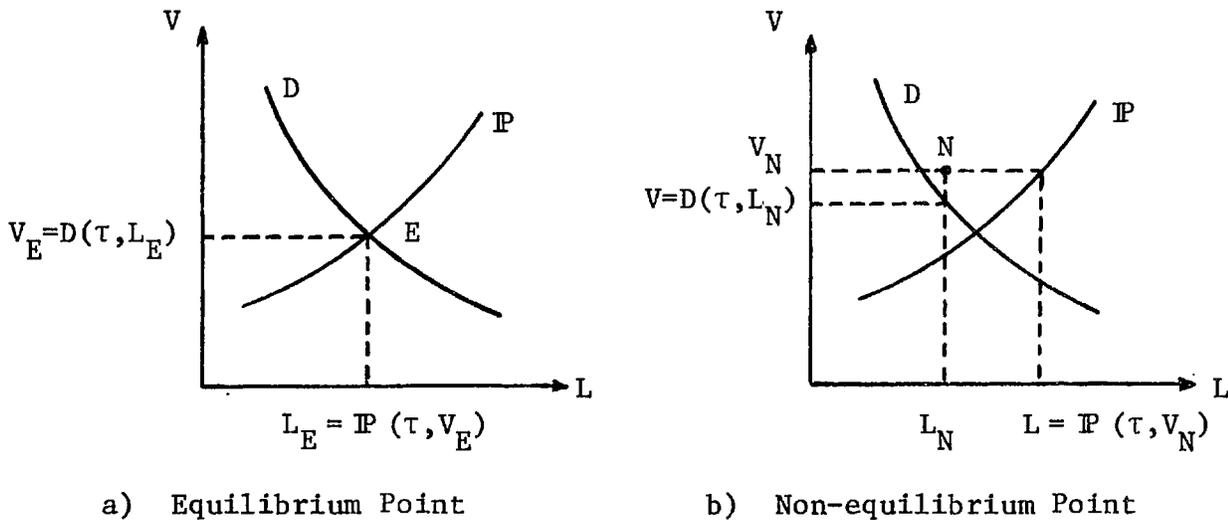


Figure 2.1 Equilibrium

2.3 MODELING TRANSPORTATION SYSTEMS

2.3.1 Aggregation

One of the most important and difficult tasks in transportation modeling is the calibration of choice functions d_i for all of the users. For any real-life problem, the enormous number of users in the system and lack

of perfect information makes it almost impossible to carry out this kind of analysis. Therefore, some assumptions are required about how a chosen user makes decisions.

In the transportation literature, many types of assumptions have been used for different research purposes and for different steps of the user decision-making process, mainly to determine frequency of trips, destination choice, mode choice, and route choice [T-A5, T-M2]. To simplify the problem, the first attempt in almost all previous work has been the classification of the users into homogeneous groups, a process sometimes called aggregation. Aggregation can be in terms of level of income, family size, residential location, job classification, and so forth [T-M2, T-M6]. We assume that all of the users in a group respond similarly to any given situation and we do not distinguish among users within a group. In other words, we do not care who within a group makes the trip; we just care that some user does.

The process of calibrating the demand function is complicated not only by the difficulty in calibrating any particular demand function, but by the enormous number of points where trips originate, which makes the size of the problem too large to be manageable. In practice, planners overcome this difficulty by introducing a spatial aggregation that represents the homogeneous population of each zone as a point called *centroid*. Daganzo [T-D4] discusses this spatial aggregation and the distribution of the population in each zone.

Another type of aggregation that can be used to reduce the size of the problem is aggregation in the structure of the network itself; for

example, aggregation of the nodes, links or even centroids [T-G2, T-H2, T-Z1]. This type of aggregation reduces the size of the problem so that the problem becomes manageable, in terms of both the computational time and storage requirement, although it causes new errors.

In this report, we will assume that the aggregation process has already been carried out and that we are given aggregate user demand functions in an aggregate network.

2.3.2 Deterministic and Stochastic Models

There are two completely different approaches for modeling how users within a group behave. The first approach is to assume that the user's response is a random phenomenon with a given density function describing each group. This approach is called stochastic (non-deterministic) or disaggregate modeling [T-A3, T-M2, T-M6, T-S1], although the term disaggregate seems inappropriate. The main task in this modeling approach is calibrating the parameters of the density function. In this research we are not considering this type of model; instead, we focus on deterministic models [T-M2].

For the deterministic model, we assume that an analytical function can be established that specifies the number of users within any group who select each available alternative or, in other words, it gives the distribution of the flow among alternatives.

2.3.3 Simultaneous and Sequential Models

As we have stated previously, each alternative is composed of a set of components, mainly, trip frequency (make a trip or not), destination choice,

mode choice, and route choice. Depending upon the nature of the trip, some of the components might be fixed. For example, usually in work-trips the frequency of trips and the destinations are fixed, while for shopping-trips, all of the components vary, especially the destination choice.

In reality, the components of each alternative are not independent of one another, and each user usually makes his decision *simultaneously* considering all the components together.

The simultaneous models assume that, for any group of users, any origin-destination pair and any mode, we are given a function $D(A,L)$ that specifies the total number of trips to be made with the current system activities, A , and the current level of service, L .

An example of the simultaneous model is the one developed by Kraft for intercity passenger travel demand. For the case of three modes, the model assumes the following functional format:

$$D_{k\ell m}(t,c) = \phi_0^m (P_k \cdot P_\ell)^{\phi_1^m} (Y_k \cdot Y_\ell)^{\phi_2^m} \cdot \prod_{m'=1}^3 (\alpha_{k\ell m'}^m \cdot c_{k\ell m'}^\beta)$$

where

$D_{k\ell m}$ = demand between k and ℓ by mode m

P_k = population in zone k

Y_k = median income in zone k

$t_{k\ell m}$ = travel time between k and ℓ by mode m

$c_{k\ell m}$ = travel cost between k and ℓ by mode m

ϕ, α, β = parameters of the model (subscripts indicate mode dependency).

For more details about this model and other models presented by McLynn [T-M7] and Baumol-Quandt [T-Q1], see Manheim [T-M2].

Unfortunately, in practice, calibrating a simultaneous demand model is not an easy job, and might even be impossible for the general case [T-M6].

An alternative to simultaneous models is a sequential approach. In this model we assume that some of the components of the decision-making process are independent, that they can be ordered in a hierarchy of steps of decision-making, and that they can be modeled separately [T-M2]. One of the common hierarchy orderings that has been used by a number of transportation planners for both deterministic and stochastic models is as follows:

- i) Frequency of Trips or Trip Generation
- ii) Destination Choice
- iii) Mode Choice
- iv) Route Choice.

In this hierarchy, the first model is used to determine the number of trips generated at each zone. Given the number of trips generated at the zones, a destination choice model is applied to distribute the trips among possible destinations. Given the number of trips between an O-D pair, the mode choice model is used to split the total trips among all available modes. Finally, the route choice model is used for each mode to distribute the trips among all existing paths between an O-D pair.

Different researchers have proposed a number of models for each step, such as a linear model [T-M2] for trip generation, a gravity model [T-M2,

T-U1] and opportunity model [T-M2] for the destination choice step, and a table look-up model [T-C1] and a binary choice logit model [T-F2] for the mode choice step.

One of the most common class of models which can generally be used in any choice situation (both the deterministic and stochastic cases) for choosing among a set of alternatives is the logit model. Suppose we are given A alternatives and u^a represent some characteristic of the alternative a. The choice of alternative a is given as:

$$d^a = d \frac{e^{f^a(u^a)}}{\sum_{a' \in A} e^{f^{a'}(u^{a'})}}$$

where d is some constant and f^a is some function of u^a .

Florian in [T-F4] used the following logit model for the mode choice step:

$$D^m(u) = d \frac{e^{\theta u^m}}{\sum_{\substack{\text{all modes} \\ m'}} e^{\theta u^{m'}}}$$

where d is the total number of trips by all modes between a given O-D pair, u^m is the travel time by mode m and θ is some constant. Dial [T-D7] has proposed an extension of this model for making both destination choice and mode choice simultaneously, as follows:

$$D_{pq}^m(u) = d_p \frac{r_q e^{\theta_{pq}^m}}{\sum_{q'} \sum_{m'} r_{q'} e^{\theta_{pq'}^{m'}}}$$

where d_p is the total number of trips generated at origin p , and r_q is an index of the attraction of destination q .

2.3.4 Route Choice

One of the most traditional assumptions that has been used indirectly in almost all past work is that the route choice step is independent of the other steps and is the last step in the hierarchy sequence. Furthermore, the distribution of the flow among the available paths is such that all of the used paths have equal travel time, which is less than or equal to the travel time for non-used paths. Wardrop was first to state this law of the distribution and it later became known as *Wardrop's first principle* or as *the user-equilibrium law*. As we mentioned previously, the user-equilibrium notion has much broader meaning than this special case. This is only one possible type of assumption that we might make for the distribution of path flow.

A definition analogous to Wardrop's first principle that has been used is:

"At equilibrium no user can improve his travel time by unilaterally changing paths."

Although most papers in the literature [T-A4, T-F6, T-M2] have explicitly assumed that these definitions are equivalent, there is no formal proof.

In section 3.1 we show by example that these definitions are not always equivalent.

The second definition says that the users are in a competitive market and that each user tries to improve his own travel time. For this reason it is sometimes called a *user-optimized* formulation. In contrast with this formulation, there is *system-optimized* (Wardrop's second principle) formulation wherein all the used paths have equal *marginal* travel times (or the average travel time is minimum), as compared with the user-optimized formulation wherein all used paths have the same travel times.

2.3.5 General Route Choice

At least theoretically, we can use any other type of function, besides the traditional ones, for the distribution of the flow among the paths. Relaxing the restrictions in the traditional model permit us to have more flexible models that include directly attributes like travel cost, safety, and convenience, as well as travel time, which was the only attribute of the level of service for Wardrop's first principle. Also, in reality, it is not true that all used paths have equal travel times. This might be because of the lack of user awareness as to their route choice possibilities or to travel times; or, it might be because other attributes are important to the users in their route choice.

Sheffi in his thesis [T-S1] introduced a type of probability distribution function for the path flow distribution of a stochastic model. This model permits a small percentage of flow in paths with the higher travel times, which is more realistic.

Generally, for the deterministic case, the route choice step can be modeled like any other step of the decision-making process and in a variety of ways, like all or nothing routing, logit distribution, and so forth. For example, consider a logit model. Let \bar{d} be the total number of users who are going to travel through k available paths and let L_k denote the level of service for path k . Then the number of users who travel through path k is given by:

$$d_k = \bar{d} \frac{e^{-\theta_k L_k}}{\sum_j e^{-\theta_j L_j}} \quad \theta_j > 0.$$

Appropriate choices of θ_k permit us to include implicitly other attributes besides travel time.

Also, the traditional path flow distribution satisfying Wardrop's first principle can be written in the form of the following analytical function:

$$d_k = \bar{d} \alpha_k \quad \text{for all } k.$$

where α satisfies:

$$\left\{ \begin{array}{l} \alpha_k \geq 0 \\ \sum \alpha_k = 1 \\ \alpha_k = 0 \end{array} \right. \quad \text{if } L_k^* > L_{\min}^*$$

where L_{\min}^* is the minimum level of service among all paths at equilibrium.

As we mentioned before, the first advantage of this general route choice model is that we have more flexibility to assign the flow among the paths, considering the other components of the level of service beside the travel time, which are closer to the real-life distribution of the flow. The other advantage is that the route choice can be modeled similarly to mode choice, destination choice, or even trip frequency. In fact, by introducing new nodes and arcs with appropriate volume delay functions, all the components of the decision-making process can be induced in the route choice step in a new network, which is called the *hypernetwork* [T-S2].

Considering the enormous number of paths in the network, it is almost impossible to calibrate a model like a logit directly for the general route choice model. However, in reality, only a small number of all available paths will have positive flow and their choice depends upon the level of congestion. Therefore, if somehow we could enumerate the possible paths with positive flow, then we could use any functional form, such as the logit model, for the path flow distribution.

On the other hand, there are some existing efficient techniques (i.e., a shortest path algorithm) to assign the flow among the paths satisfying Wardrop's principle, when the level of service consists of travel time only, without considering all the existing paths explicitly. This fact makes the use of the flow distribution satisfying Wardrop's principle more attractive.

For the above reason, in this thesis, we only work with the traditional route choice model in terms of computational results, although, it seems that the theoretical developments are valid for the general route choice model and for the hypernetwork.

2.3.6 Level of Service

As we have mentioned several times previously, the level of service vector is composed of several components including travel time (in-vehicle and out-of-vehicle times), travel cost, safety, and convenience. Although all of the components greatly effect a user's choice of alternatives, it is difficult to incorporate some of these components, like safety and convenience, in a mathematical model for the equilibrium problem. It is hard enough to include other variables that are even easier to measure. Thus, practically, only travel time and travel cost have been considered as components for the level of service.

Usually for the short-run equilibrium, the travel cost does not change with the volume of the traffic in the network, or it is assumed to be proportional to travel time as perhaps when gas consumption increases as the in-vehicle travel time increases. (Proportionality may not always be a good assumption, though. For example, in case of high speed, the travel cost increases while travel time decreases.) Travel cost does depend strongly on traffic volume, though. For this reason, most traffic equilibrium models do not consider travel costs explicitly in the demand functions, whereas the travel time usually is considered explicitly as a variable. Thus the travel times are needed for the demand model.

Also, almost all previous work uses Wardrop's principle for path flow distribution and does not include travel cost in the route choice step. Although there has been some attempt to use a generalized travel time (a function, usually linear, of travel time and travel cost), there are no computational results for these types of models.

In this research we also consider the travel time as the only measure for the route choice step, and we use constant travel cost for the other parts of the demand model. However, it might be possible to extend all of the results, especially those that are theoretical in nature, to consider generalized travel time.

2.3.7 Volume Delay Function

Traveling through any path in the network involves delay time associated with both nodes and arcs in the path. Delay time at a node refers to waiting time for transfer to another mode, waiting time for service, waiting time at intersections and so forth. Delay time at an arc refers to the actual time required for physical movement and, possibly wait time. Since the delay time at a node can be represented by the delay time at an arc in a suitably modified model of the transportation network (for example, representing an intersection by a set of arcs [T-F7, T-F8]), we can assume that there is no delay at nodes, and by an arc we mean a generalized arc.

In general, the travel time in a path depends on the volume of traffic in the whole network. However, to be able to model the problem that can be solved, some assumptions are needed. The first "natural" assumption is that the travel time for a path is the sum of the travel times of the arcs in the path. This assumption might not be true. For example, even if two arcs have equal travel times, they might have different disutilities. Consider traveling in an attractive neighborhood as compared to another unattractive neighborhood, or walking compared to riding in a luxurious car. Since, in the route choice model the travel time is the only measure, to model

differences between arcs, we might use some scaling factor in the volume delay function for each arc. We have to notice that the above assumption makes the computation of the finding of the shortest path much easier, as compared to the case of travel times on arcs, are not additive, because there are very efficient algorithms available for the additive case.

The fact that each user affects the delay time for each arc differently, forces us to consider each user individually. But this is not feasible because the number of users is enormous. One way to solve this problem is to classify the users into homogeneous groups and assume that all of the users in the group have similar affects upon the delay time. The classification might be in terms of transportation mode (i.e., auto and bus), vehicle size (i.e., private auto and truck), drivers (i.e., slow drivers and fast drivers), and so forth.

Suppose that, for every arc $a \in A$, E_a denotes the set of groups who are sharing arc a and v_a^e denotes the volume of traffic on arc a for group e . Then the volume delay function for group e on arc a can be represented as $t_a^e(v)$, where v is the vector of traffic volume by all groups and all arcs. Notice that, for this general type of volume delay function, theoretically, we can assume that each arc is used by only one group, simply by duplicating the whole network by the number of groups. The new network would be much larger; however, from the computational point of view this duplication might be made only implicitly.

Most existing models have assumed that the delay on an arc depends only on its own volume, i.e., $t_a^e(v) = t_a^e(v_a)$, where v_a is the vector of the volumes by all groups using arc a . Although it is true that the delay

on an arc usually does not depend on the arcs far from that arc, it may depend upon the flow on arcs close to it. For example, the delay on an arc corresponding to a left turn strongly depends on the volume of the flow on the arc representing the cross street and vice-versa. Another example is two-way streets: the delay for each direction depends on the volume of traffic in both directions.

Another assumption that has been made for simplicity in most models is that the effect of different groups on the delay time can be captured by assigning constant weight factors to each group. In other words, t_a^e is a function of $\sum_{e \in E_a} \alpha_e v_a^e$, where α_e is a constant. For example, a bus is equivalent to 5 autos.

2.3.8 Examples of Volume Delay Functions

Here we review some of the volume delay functions proposed most frequently in the literature. Constant functions have been used for arcs representing walking distances, waiting times (i.e., half of the head-way per bus), free-way flow time (i.e., uncongested highways, flight times, in vehicle transit time, and so forth), and so on [T-M2]. There are a few models for the delay at intersections, especially to represent traffic lights (see [T-F8], for example).

For congested street arcs, a variety of models have been used. We mention only a few of them. In most of the models, the delay time has been given only as a function of total volume on that arc. Let c denote the arc capacity,[†] let v denote the total volume, and let t_0 denote the travel

[†] Different definitions have been used for the arc capacity, mainly the "steady state" capacity used in Overgaard's model and the "practical" capacity used in BPR model. For more details, see [T-B12].

time at zero flow for an arc.

In 1962, Irwin and Von Cube [T-I2] introduced a piece-wise linear function. In 1967, Overgaard [T-O1] proposed the following exponential function:

$$t(v) = t_0 \alpha (v/c)^\beta$$

where α and β are constant parameters.

In 1963, Mosher [T-M10] suggested logarithmic and hyperbolic functions, e.g.,

$$t(v) = t_0 + \ln \frac{\alpha}{\alpha - v} \quad \text{for } v < \alpha \quad (\text{logarithmic})$$

$$t(v) = \beta + \frac{\alpha(t_0 - \beta)}{\alpha - v} \quad \text{for } v < \alpha \quad (\text{hyperbolic})$$

where α and β are constants. Although these functions are not defined for $v \geq \alpha$, by changing the function for $v \geq \alpha_s$, where $\alpha_s < \alpha$, we can construct a well-defined function for all $v \geq 0$ (see [T-M10]).

One of the best known and most widely-used volume delay functions is that often referred to as the BPR (Bureau of Public Roads) [T-B12] function:

$$t(v) = t_0 [1 + \alpha (v/c)^\beta]$$

where α and β are constant parameters. The BPR engineers suggested values of 0.15 and 4 for α and β , respectively. However, generally for all of the above models, the value of the parameters depends upon the structure of the arcs. For example, they depend upon the number of lanes, the speed limit, vehicle type, and so forth. Usually there are some tables available to calibrate these functions. For more details concerning these models see Branston [T-B10].

The transportation literature does not contain many models to represent the volume delay function for a link that is used by more than one class of users. In [T-F4], Florian uses a special model for the links that are used by two modes of transportation, namely private auto and transit bus. By using some conversion factor, $\bar{\alpha}$, he assumed that the flow by auto, v^{au} , and the flow by bus, v^b are additive, i.e.:

$$v = v^{au} + \bar{\alpha}v^b .$$

A bus is assumed to be equivalent to a multiple of private cars (in the traffic engineering study [T-H3], $\bar{\alpha}$ is 3 or 4). Florian uses the BPR function for the volume delay functions, as follows:

$$t^{au}(v^{au}, v^b) = t_0 [1 + \alpha(v/c)^\beta]$$

$$t^b(v^{au}, v^b) = \gamma \cdot t^{au}(v^{au}, v^b) + \delta$$

where δ is a constant penalty per mile to allow for stopped time for buses, and γ is another constant to allow for speed differences between two modes.

2.3.9 Nature of Volume Delay Functions

It is natural to assume that the volume delay function is a continuous function. This assumption might not be valid in some special instances, as when modeling delay at a traffic light [T-G1, T-M8]. When the flow arriving at a traffic light increases more than the number of vehicles that can pass in one cycle, the delay time will increase by another cycle time. Thus, the delay function would have a step-wise character.

The second type of assumption which seems natural is that the volume delay function is positive and monotone. If t denotes the vector of volume delay functions (i.e., $t(v) = \{t_a^e(v)\}$ for all $e \in E_a$ and $a \in A$) and v denotes the vector of volumes, then t is called monotone if,

$$(v - v') \cdot (t(v) - t(v')) \geq 0 .$$

When t_a^e is only a function of v_a^e then this property says that t_a^e is non-decreasing, which is what we expect for the transportation applications. Furthermore, in this case, for congested arcs, we can assume that t_a^e is strictly increasing, even though the slope of the function may be close to zero (see all the above examples). Note though, that if the transportation technology is permitted to vary, then this assumption might not be valid. For example, the delay in waiting for a bus might decrease with increased user demand, as when more frequent bus service is provided at

rush hours. For the general case, t_a^e might not be strictly monotone, especially when t_a^e is a function of $\sum_{e \in E_a} \alpha_e v_a^e$. For example, it is easy to see that the volume delay function used by Florian in [T-F4] is monotone, but it is not strictly monotone.

CHAPTER 3

TRAFFIC EQUILIBRIUM PROBLEM

3.1 EQUILIBRIUM CONCEPTS

In the transportation literature, the term "equilibrium" has been used in a number of different ways. In this section we attempt to unify and clarify this term and to state what we mean by an equilibrium in this report.

In general, we refer to any fixed-point solution for the system (2.3) as an *equilibrium* point, as stated in definition (2.1). This general definition is valid for both short-run equilibrium (when the transportation technology τ is fixed) and for long-run equilibrium (when τ is not fixed). Also, it is valid for both deterministic and probabilistic demand models.

In the case of the short-run equilibrium, the users are the only decision makers in the system. This is the reason for referring to the equilibrium as a *user-equilibrium*. In this case, if we assume that each user tries to optimize his own utility independent of the other users, then at equilibrium the following condition prevails:

User-Equilibrium Law:

"At equilibrium no user perceives a possible increase of his utility by unilaterally changing alternatives."

This is a generalization of the Wardrop's user-equilibrium law (see Sheffi [T-S1]) for both deterministic and probabilistic demand functions.

For the special case when the travel time is the only attribute of the level of service in the performance function, when the route choice is

the only decision for each user to make (i.e., when the number of trips between each O-D pair, which includes trip generation, destination choice and mode choice, is prescribed by a given function), and finally, when each user's decision is based upon minimizing his own travel time, then the above user-equilibrium law becomes:

Special User-Equilibrium Law:

"At equilibrium no user can improve his travel time by unilaterally changing routes."[†]

This law was originally stated by Wardrop [T-W1] and later has been known, at least intuitively, as the definition of user-equilibrium. At the same time, in practice, to introduce this law into a mathematical formulation of the problem, Wardrop proposed an analogous law (known as Wardrop's first principle) stated as follows:

Traffic-Equilibrium Law:

"At equilibrium, for each O-D pair the travel time on all the routes actually used are equal, and less than the travel times on non-used routes."

We used the name *traffic equilibrium* for this law to distinguish between it and the special user equilibrium law, although in the literature the same name has been used for both laws.

It is important to note what we mean by the term user in the definition of a user equilibrium. If we view the transportation system as being composed of a finite number of individuals who make trips or not, then each "user" provides an integral unit of flow from its origin to its destination.

[†] Here we assume that each user has knowledge about the effect that his transfer onto a new route has upon travel time.

In this case the flow variables must be restricted to assume integral values. Rosenthal [T-R1] models the equilibrium problem from this point of view.

On the other hand, if we view the demand between each origin and destination pair as a collection of a large number of individuals who are making trips, then we might view the flow as being decomposable into a large number of smaller units or users. A limiting assumption would be that flow is infinitely divisible; that is, the flow variables are continuous and each user is an infinitesimal unit of flow. The relationship between the continuous model and limiting behavior of the integral model seems not to be well understood. See Weintraub [T-W3] though, for results of this nature.

By imposing implicit assumptions (such as continuous variables, continuous volume delay functions and non-decreasing volume delay functions) transportation analysts have assumed that these two laws are equivalent. Since these assumptions might not be true in general, and also since it is not clear exactly when these laws are equivalent, we refer to any equilibrium point that satisfies the traffic-equilibrium law as the *traffic equilibrium* solution. In the next section, when we formulate the problem, we introduce equivalent mathematical equations that characterize the traffic-equilibrium law.

EXAMPLE 3.1: To see the differences between these two definitions, when any of the implicit assumptions are relaxed, we consider a single O-D pair example with two units of flow and two paths:

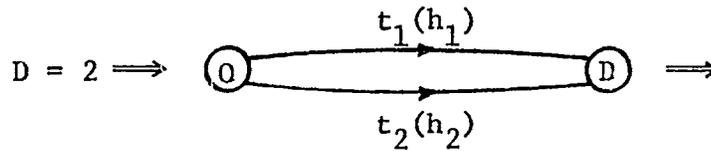


Figure 3.1 Network Configuration for Example 3.1

h_1 and h_2 represent the path flows and $t_1(h_1)$ and $t_2(h_2)$ are the corresponding volume delay functions. We consider the following cases:

Case I: Integral Variables.

Consider the following continuous volume delay functions:

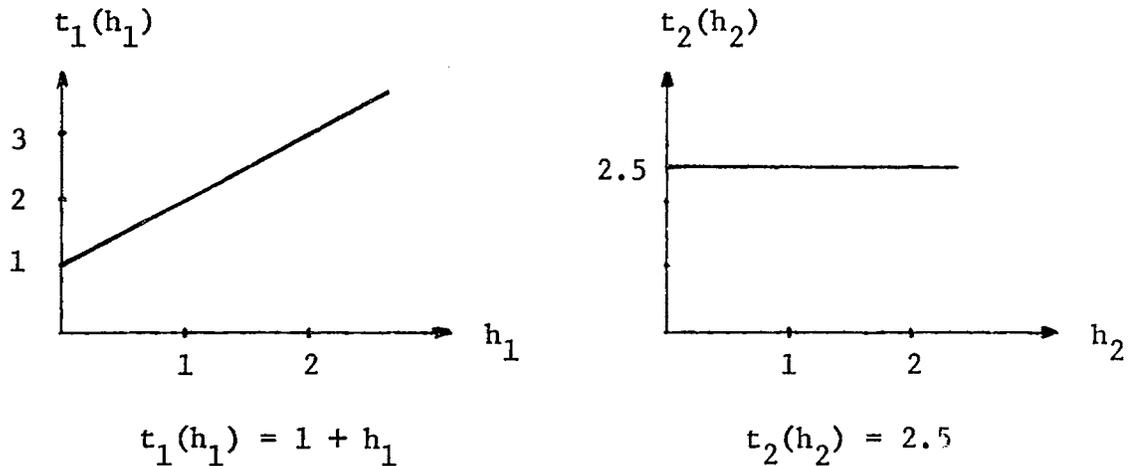


Figure 3.2 Continuous Volume Delay Functions

For this example with continuous and non-decreasing volume delay functions, the equilibrium problem with a user-equilibrium law has a unique solution; namely $h_1 = h_2 = 1$ with perceived times $t_1(h_1) = 2$ and $t_2(h_2) = 2.5$. With

the traffic-equilibrium law the problem has no equilibrium solution, because $h_1 = 0$ or 1 implies that $t_1(h_1) < 2.5 = t_2(h_2)$ and $h_1 = 2$ implies that $t_1(h_1) = 3 > 2.5 = t_2(h_2)$. However, in the case of continuous variables, the problem has a unique equilibrium solution for both laws; namely $h_1 = 1.5$ and $h_2 = 0.5$ with the perceived travel times equal to 2.5.

Case II: Non-Continuous Volume Delay Functions.

Consider the following non-decreasing volume delay functions:

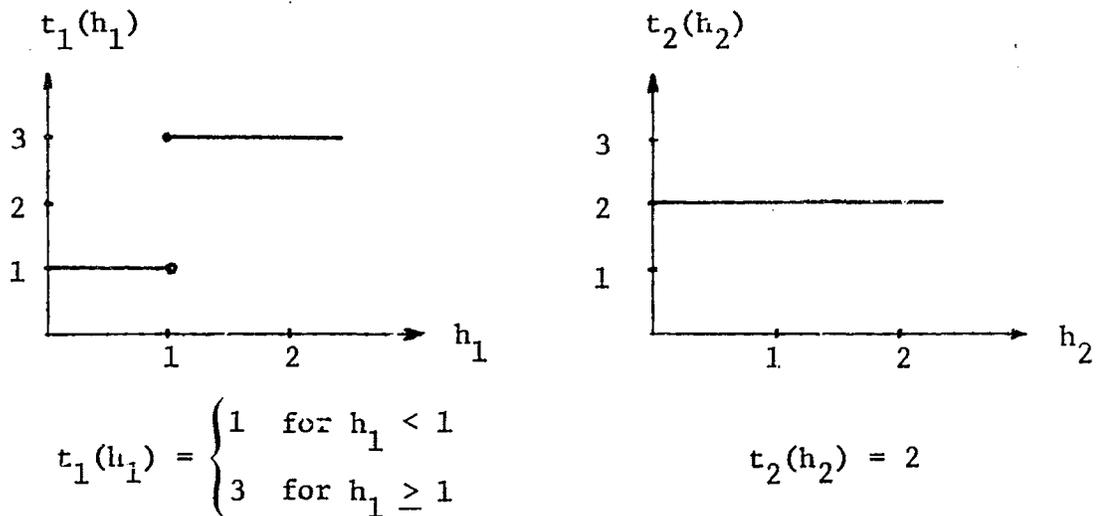


Figure 3.3 Non-Continuous Volume Delay Functions

In the case of integral variables, this example with the user-equilibrium law has a unique equilibrium solution, $h_1 = 0$ and $h_2 = 2$. But with the traffic equilibrium law it has no solution, because for $h_1 < 1$ we have $t_1(h_1) = 1 < 2 = t_2(h_2)$ and for $h_1 \geq 1$ we have $t_1(h_1) = 3 > 2 = t_2(h_2)$.

In the case of continuous variables, this problem has no equilibrium solution even with the user-equilibrium law, because if $h_1 < 1$ then some

users, Δh for $0 < \Delta h < 1 - h_1$, can improve their travel times by transferring from the second route to the first route. For a similar reason, we cannot have $h_1 \geq 1$.

Case III: Decreasing Volume Delay Function.

Consider the volume delay functions given as follows:

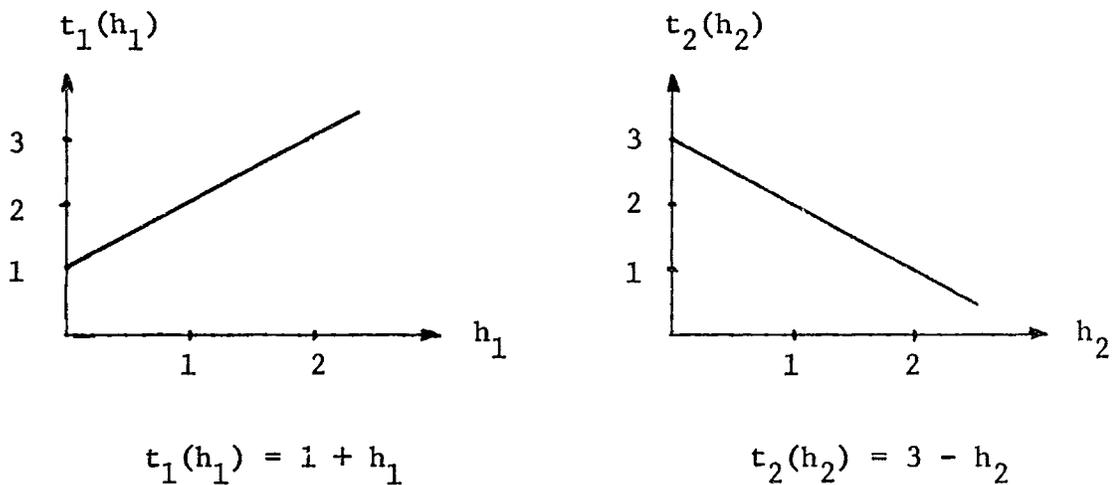


Figure 3.4 Decreasing Volume Delay Function

For continuous variables, this problem with the user-equilibrium law has a unique solution, that is, $h_1 = 0$ and $h_2 = 2$. It has infinitely many solutions with the traffic-equilibrium law; those are $h_1 = b$ and $h_2 = 2 - b$ for all $0 \leq b \leq 2$ with travel times equal to $t_1(h_1) = t_2(h_2) = 1 + b$. Among all these solutions, the one with $b = 0$, which is the solution with the user-equilibrium law, has the minimum travel time. But minimization of the travel time is not part of the traffic equilibrium law. ■

These examples show that the continuity assumption for both variables and volume delay functions, and also a non-decreasing assumption for the volume delay functions are necessary conditions for the user-equilibrium law and the traffic-equilibrium law to be equivalent. Even under these assumptions, there is no formal proof that the two laws are equivalent. Because of these differences, in this report we assume that the traffic equilibrium law governs the distribution of flow among the existing paths. From this position, by traffic-equilibrium, traffic-assignment, user-equilibrium, user-optimized, or an equilibrium problem, we mean the fixed-point problem discussed in Section 2.2 with the traffic-equilibrium law governing the distribution of flow among the paths.

Also, these examples show that the continuity assumption for both variables and volume delay functions is a necessary assumption for the traffic-equilibrium problem to have a solution, while a non-decreasing assumption on the volume delay function is not required. We prove this claim in Chapter 3. Although it might seem that the volume delay function for transportation applications is non-decreasing, this is not always true, especially when the transportation technology is not fixed. For example, consider a shuttle bus system in which the number of buses depends on the number of passengers. In this case the delay time (waiting time) may decrease as the number of passengers increases and new buses are added to the system.

3.2 PROBLEM FORMULATION

The Transportation Science and Operations Research literatures contain

a number of different representations of the traffic assignment problem, modeling a variety of features such as multiple origins and destinations, multi-modal routing, and multiple classes of users. Although these specializations facilitate understanding of problems and provide intuition and guidance for developing solution techniques, they lead to somewhat fragmented views that inhibit investigations that might apply across a wide range of applications. The area has now matured to the extent that a broader perspective is possible. In this section we formulate a rather general version of the traffic assignment problem that can be specialized to any of these previously considered cases by defining appropriately the network structure, the problem variables, and the functional forms for vehicle delay and origin destination demand. Later in this section we give an example to make this point clear.

This general formulation reduces notational difficulties enormously for the theoretical investigations to be pursued in the next chapters.

We formulate the problem as a multiple origin destination traffic assignment problem. For a given network $[N,A]$ where N is the set of nodes and A is the set of (directed) arcs, the user-equilibrium traffic assignment problem can be formulated as:

$$(3.1) \left\{ \begin{array}{ll} (T_p(h) - u_i)h_p = 0 & \text{for all } p \in P_i \text{ and } i \in I \quad (3.1a) \\ T_p(h) - u_i \geq 0 & \text{for all } p \in P_i \text{ and } i \in I \quad (3.1b) \\ T_p(h) = \sum_{a \in A} \delta_{ap} \cdot t_a^i(h) & \text{for all } p \in P_i \text{ and } i \in I \quad (3.1c) \\ \sum_{p \in P_i} h_p - D_i(u) = 0 & \text{for all } i \in I \quad (3.1d) \\ h \geq 0 & (3.1e) \\ u \geq 0 & (3.1f) \end{array} \right.$$

where,

the index a denotes an arc, $a \in A$,

I is the set of O-D pairs,

the index i denotes an O-D pair, $i \in I$,

the index p denotes a path, $p \in P_i$,

P_i is the set of "available" paths for flow for O-D pair i

(which might, but need not, be all paths joining the O-D pair),

h_p is the flow on path p ,

h is the vector of $\{h_p\}$ with dimension $n_1 = \sum_{i \in I} |P_i|$ equal to the total number of O-D pairs and path combinations,

u_i is an accessibility variable, shortest travel time, for O-D pair i ,

u is the vector of $\{u_i\}$, with dimension $n_2 = |I|$,

$$\delta_{ap} = \begin{cases} 1 & \text{if link } a \text{ is in path } p \\ 0 & \text{otherwise} \end{cases}$$

$t_a^i(h)$ is the volume delay function for arc a and O-D pair i ,

$$t_a^i : R_+^{n_1} \rightarrow R_+^1,$$

$D_i(u)$ is the demand function for O-D pair i , $D_i : R_+^{n_2} \rightarrow R_+^1$

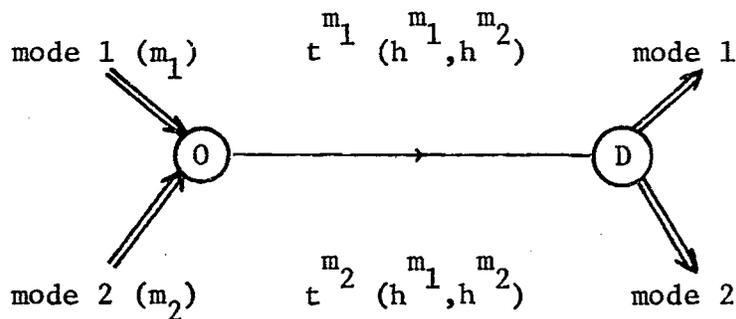
$T(h)$ is the volume delay function for path p , which is that sum of that volume delay function of the arcs in path p (a more general formulation would relax this additivity assumption).

We assume that the network is strongly connected, i.e., for any O-D pair i with positive demand there is at least one path joining the origin to the destination; i.e., $|P_i| \geq 1$. The first three equations require that for any O-D pair i , the travel time (generalized travel time) for all paths,

$p \in P_i$, with positive flow, $h_p > 0$, is the same and equal to u_i , which is less than or equal to the travel time for any path with zero flow. Equation (3.1d) requires that the total flow among different paths between any O-D pair i equals the total demand, $D_i(u)$, which in turn depends upon the congestion in the network through the shortest path variable u . Condition (3.1e) and (3.1f) state that both flow on paths and minimum travel times should be non-negative.

Up to this point, we have not imposed any restrictions on the volume delay function. It is a function of all the flows in the network; by defining the structure of the network appropriately this formulation can model a wide range of equilibrium applications, for instance situations with multi-modal and multi-class of users with mixed type of flow in an arc (such as bus and auto), with separate type of flow in arcs (such as subway and auto), or even two way traffic in one arc and, right and left turn penalties.

For example, consider a single link network with two modes of transportation (i.e., auto and bus) and one O-D pair, and suppose that the volume delay function for each mode depends upon the flows by both modes.



To formulate the problem as a single mode problem with multiple O-D pairs, we make duplicate copies of the networks, one for each mode. Define the volume delay functions as follows:

$$\begin{array}{l}
 t_1(h) = t^{m_1}(h^{m_1}, h^{m_2}) \implies \text{mode 1} \\
 t_2(h) = t^{m_2}(h^{m_1}, h^{m_2}) \implies \text{mode 2}
 \end{array}$$

By this device of duplicating the network and letting $t_a(h)$ be a function of the vector of h in the generalized network, we can assume that there is only one type of commodity (user class, mode, and so forth) flowing in each arc. Thus in the generalized network we can omit index i from $t_a^i(h)$. In the theoretical part of this report, we will work with this generalized network.

3.3 EQUIVALENT NON-LINEAR COMPLEMENTARITY PROBLEM (NCP)

Let $F(x) = (f_1(x), \dots, f_n(x))$ be a vector-valued function from a n -dimensional space R^n into itself. Then a vector $x \in R^n$ is called a complementarity solution if it satisfies the following conditions [see C-K2]:

$$\begin{cases}
 x \cdot F(x) = 0 \\
 F(x) \geq 0 \\
 x \geq 0.
 \end{cases}$$

This formulation can also be generalized for the point-to set mapping [see C-E2].

In this section we show that the traffic equilibrium problem (3.1) has a complementarity nature. It is clear that equations (3.1a), (3.1b) and (3.1e) are complementary in nature. To show that the rest of the equations can be expressed in a complementarity form requires some mild assumptions that we would expect to be met always in practice.

First, some simplification in the formulation helps to clarify the transformation. Let $x = (h, u) \in R^n$ where $n = n_1 + n_2$ and furthermore, let

$$f_p(x) = T_p(h) - u_i \quad \text{for all } p \in P_i \text{ and } i \in I$$

and

$$g_i(x) = \sum_{p \in P_i} h_p - D_i(u) \quad \text{for all } i \in I.$$

Also, let

$$F(x) = (f_p(x) \text{ for all } p \in P_i \text{ and } i \in I, g_i(x) \text{ for all } i \in I) \in R^n$$

then F would be a vector-valued function from a n -dimensional space R^n into itself. Now consider the following nonlinear complementarity system:

$$(3.2) \quad \left\{ \begin{array}{ll} f_p(x) h_p = 0 & \text{for all } p \in P_i \text{ and } i \in I \\ f_p(x) \geq 0 & \text{for all } p \in P_i \text{ and } i \in I \\ g_i(x) u_i = 0 & \text{for all } i \in I \\ g_i(x) \geq 0 & \text{for all } i \in I \\ x \geq 0 & \end{array} \right.$$

which can be written as the following compact form:

$$(3.3) \quad \begin{cases} F(x) \cdot x = 0 \\ F(x) \geq 0 \\ x \geq 0 \end{cases}$$

PROPOSITION 3.1: Suppose that $t_a: R_+^{n_1} \rightarrow R^1$ for all $a \in A$. Also, suppose that $D_i: R_+^{n_2} \rightarrow R^1$ for all $i \in I$. Then any solution to the user-equilibrium system (3.1) is a solution to the nonlinear complementarity system (3.3).

PROOF: Obvious, since $g_i(x) = 0$ in the user equilibrium conditions (3.1). ■

PROPOSITION 3.2: Suppose that for all $a \in A$ that $t_a: R_+^{n_1} \rightarrow R_+^1$ is a positive function. Also, suppose that for all $i \in I$ that $D_i: R_+^{n_2} \rightarrow R_+^1$ is a non-negative function. Then the user-equilibrium system (3.1) is equivalent to the nonlinear complementarity system (3.3).

PROOF: To prove the theorem, it is enough to show that any solution to (3.3) is a solution to (3.1). Suppose to the contrary that there is a $x = (h, u)$ satisfying (3.3), but that $g_i(x) = \sum_{p \in P_i} h_p - D_i(u) > 0$. Then $g_i(x)u_i = 0$ implies that $u_i = 0$. Also, since D_i is non-negative

$\sum_{p \in P_i} h_p > D_i(u) \geq 0$ which implies that $h_p > 0$ for some $p \in P_i$. But, for this particular p , equation $f_p(x)h_p = 0$ implies that:

$$f_p(x) = T_p(h) - u_i = 0$$

or

$$T_p(h) = u_i.$$

But since $u_i = 0$, $T_p(h) = \sum_{a \in A} \delta_{ap} \cdot t_a(h) = 0$ which contradicts the assumption $t_a(h) > 0$. ■

REMARK 1: The user-equilibrium system (3.1) need not be equivalent to the nonlinear complementarity system (3.3) if the assumption $D_i(u) \geq 0$ is dropped from this proposition. For example, consider the following network with a single link and a single O-D pair.

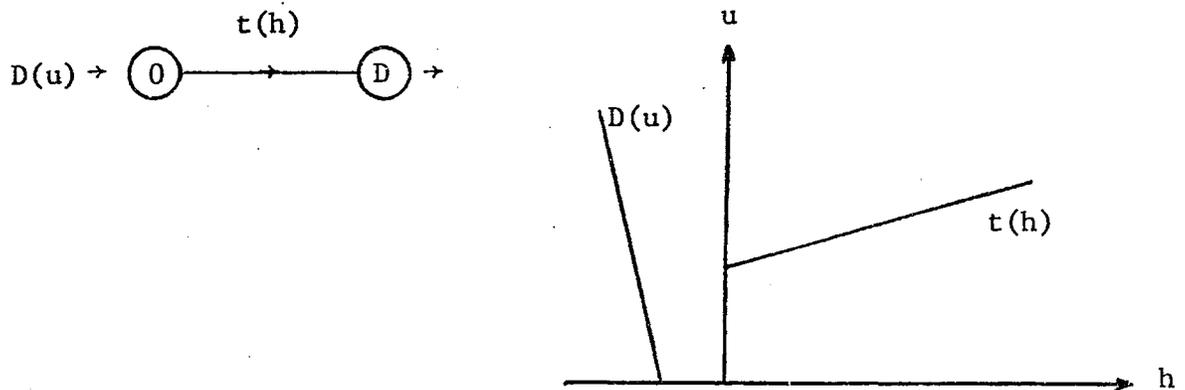


Figure 3.5 Negative Demand Function

In this example, $(h,u) = (0,0)$ is a solution to the nonlinear complementarity system, while the user-equilibrium system does not have any solution. ■

REMARK 2: The user-equilibrium system (3.1) need not be equivalent to the nonlinear complementarity system (3.3) if the assumption $t_a(h) > 0$ is dropped from proposition 3.2. For example, consider the following network with a single link and a single O-D pair:

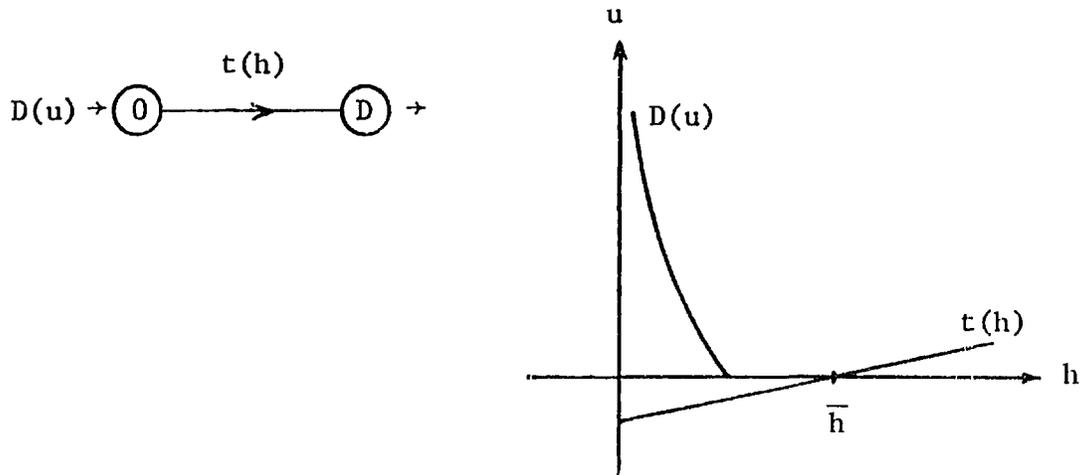


Figure 3.6 Negative Volume Delay Function

For this example, $(h,u) = (\bar{h},0)$ is a solution to the nonlinear complementarity system, but not to the user equilibrium system. ■

CHAPTER 4

EXISTENCE AND UNIQUENESS OF AN EQUILIBRIUM

4.1 INTRODUCTION

In this chapter we prove both the existence and uniqueness of the solution to any traffic equilibrium model (3.1) that satisfies assumptions that are not very restrictive for transportation applications and do not limit, in any essential way, the generality of the model. Although the literature contains some proofs of existence and uniqueness for special cases when the problem can be formulated as an equivalent optimization problem (see [T-D2-3], [T-F6] or [T-S4]), this approach seems to require strong assumptions that make it difficult, if not impossible, to extend the formulation and the proofs to more general settings (see Section 5.3.2).

The nonlinear complementarity formulation provides us with a stronger tool to generalize the formulation of the user-equilibrium, to extend the existence and uniqueness theorems, and even to introduce new solution techniques. This might be because user-equilibrium is essentially complementary in nature.

4.2 EXISTENCE

Several researchers [C-K1-4] have developed theorems that provide necessary conditions for the existence of a solution to the nonlinear complementarity problem. Unfortunately, most of the conditions are too strong to be applied directly to the user-equilibrium problem. To illustrate this situation we introduce a prototype of this theory, by considering results

due to Karamardian. Later in Section 4.3 when we discuss uniqueness we utilize some of the concepts introduced at this point. Before starting the theorem, we require some definitions:

DEFINITION 4.1: Let $F : D \rightarrow E^n$, $D \subset E^n$. The function F is said to be *monotone* on D if, for every pair $x \in D$ and $y \in D$, we have

$$(x-y)(F(x) - F(y)) \geq 0.$$

F is said to be *strictly monotone* on D if, for every pair $x \in D$, $y \in D$ with $x \neq y$, we have

$$(x-y)(F(x) - F(y)) > 0.$$

It is said to be *strongly monotone* on D if there is a scalar $k > 0$ such that, for every pair $x \in D$, $y \in D$, we have

$$(x-y)(F(x) - F(y)) \geq k \|x-y\|^2$$

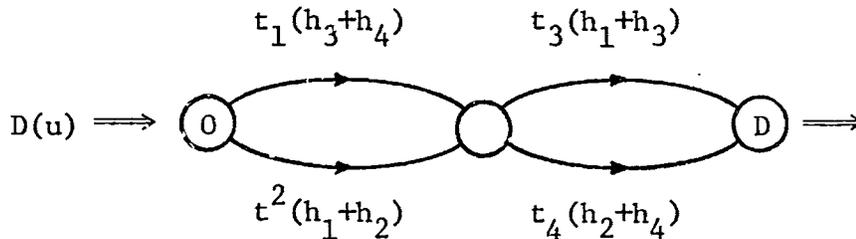
where $\|\cdot\|$ denotes the usual Euclidean norm.

THEOREM 4.1: (Karamardian [C-K2]) If $F : E_+^n \rightarrow E^n$ is continuous and strongly monotone on E_+^n , then the nonlinear complementarity system has a unique solution.

THEOREM 4.2: (Karamardian [C-K2]) *If $F : E_+^n \rightarrow E^n$ is strictly monotone on E_+^n , then the nonlinear complementarity system has at most one solution.*

Notice that, for traffic equilibrium problems, these theorems require that $F(x) = (\sum_{a \in A} \delta_{ap} t_a(h) - u_i$ for all $p \in P_i$ and $i \in I$, $\sum_{p \in P_i} h_p - D_i(u)$ for all $i \in I$) and necessarily $t_a(h)$ be strictly or strongly monotone in terms of *path flows*. But this is not usually true since most of the time the volume delay function t_a is a function of the sum of the flow on different paths corresponding to the same O-D pair.

EXAMPLE 4.1: Consider the following single O-D pair network with 4 possible paths:



Suppose that $\bar{x} = (\bar{h}_1, \bar{h}_2, \bar{h}_3, \bar{h}_4, \bar{u})$ is a solution to the corresponding nonlinear complementarity problem. Then clearly $\bar{y} = (\bar{h}_1 + \theta, \bar{h}_2 - \theta, \bar{h}_3 - \theta, \bar{h}_4 + \theta, \bar{u})$, which has the same total O-D flow and same link flows as \bar{x} , is another solution as long as $\bar{y} \geq 0$. But

$$(\bar{x} - \bar{y})(F(\bar{x}) - F(\bar{y})) = (\theta, -\theta, -\theta, +\theta, 0) \cdot \underline{0} = 0$$

implying that F is not strictly monotone or strongly monotone. ■

However, for transportation applications, the volume delay functions are usually monotone, or even strictly monotone, in terms of *link volumes*. Later we use this property to show the uniqueness of the solution in terms of link flows. In Theorem 4.4 to follow, we show that no monotonicity assumption is required for the existence of the solution.

Before stating this theorem we recall another existence result for nonlinear complementarity problems.

DEFINITION 4.2: A bounded set $B \subseteq \mathbb{R}_+^n - D$ separates D from ∞ , if each unbounded closed connected set in \mathbb{R}_+^n that meets D also meets B .

THEOREM 4.3: (Kojimas [C-K5]) Let d be a positive vector in \mathbb{R}^n . Suppose that f is continuous, and that $B \subset \mathbb{R}_+^n - \{0\}$ separates the origin $\{0\}$ from ∞ , and that for each $\bar{x} \in B$ there is an $x' \in \mathbb{R}_+^n$ for which $(x' - \bar{x})d < 0$ and $(x' - \bar{x}) f(\bar{x}) \leq 0$. Then (3.3) has a solution.

THEOREM 4.4: Suppose (N, A) is a strongly connected network. Suppose that $t_a : \mathbb{R}_+^{n_1} \rightarrow \mathbb{R}^1$ is a non-negative continuous function for all $a \in A$. Also suppose that for all $i \in I$, $D_i : \mathbb{R}_+^{n_2} \rightarrow \mathbb{R}^1$ is a continuous function that is bounded from above. Then the nonlinear complementarity system (3.3) has a solution.

PROOF: Let d be a vector with components d^i such that

$$\infty > d_i > \text{Max} \{0, \text{Max}_{u \geq 0} D_i(u)\} \geq 0 \quad \text{for all } i \in I.$$

This maximum always exists because D_i is bounded from above. Also, let

$$\tau_i > \text{Max}\{0, \text{Max}_{p \in P_i} \text{Max}_{0 \leq h \leq d_p} T_p(h)\} \geq 0 \quad \text{for all } i \in I.$$

This maximum always exists because $T_p(h) = \sum_{a \in A} \delta_{ap} \cdot t_a(h)$ and t_a is continuous. Notice that there is at least one path for each $i \in I$.

Now, let $1 < \gamma < \infty$ and define

$$\delta = \gamma \left(\sum_{i \in I} d_i \cdot |P_i| \right) + \sum_{i \in I} \tau_i$$

where $|P_i|$ is the number of paths between O-D pair i . Also, suppose that

$$\bar{\Delta} = \{ \bar{x} = (\bar{h}, \bar{u}) \in R_+^n : \gamma \left(\sum_{i \in I} \sum_{p \in P_i} \bar{h}_p \right) + \sum_{i \in I} \bar{u}_i = \delta \}$$

and

$$\Delta' = \{ x' = (h', u') \in R_+^n : \gamma \left(\sum_{i \in I} \sum_{p \in P_i} h'_p \right) + \sum_{i \in I} u'_i < \delta \}.$$

Clearly, $\bar{\Delta}$ separates the origin $\{0\}$ from ∞ . Thus by Kojimas' theorem, it is enough to show that for any $\bar{x} \in \bar{\Delta}$ there exists an $x' \in \Delta'$ such that $(x' - \bar{x}) \cdot f(\bar{x}) \leq 0$ and $x' - \bar{x} < 0$. To prove this we distinguish two cases:

Case 1: $\bar{x} \in \bar{\Delta}$ and for some $p \in P_i$ and $i \in I$ we have $\bar{h}_p \geq d_i$. Then

$$g_i(\bar{x}) = \sum_{p \in P_i} \bar{h}_p - D_i(\bar{u}) > \sum_{p \in P_i} \bar{h}_p - d_i \geq 0.$$

Now, if $\bar{u}_i > 0$ by taking $h'_j = \bar{h}_j$, $u'_j = \bar{u}_j$ for $j \neq i$, $j \in I$ and $u'_i = \bar{u}_i - \alpha$ for some $0 < \alpha < \bar{u}_i$, we complete the proof since $(x' - \bar{x})f(\bar{x}) = (u'_i - \bar{u}_i) f_i(\bar{x}) = -\alpha g_i(\bar{x}) \leq 0$.

If $\bar{u}_i = 0$ then we have:

$$f_p(\bar{x}) = T_p(\bar{h}) - u_i = T_p(\bar{h}) \geq 0.$$

Again, since $\bar{h}_p \geq d_i > 0$, by taking $u'_q = \bar{u}_q$, $h'_q = \bar{h}_q$ for $q \neq p$, $q \in P_j$, $j \in I$ and $h'_p = \bar{h}_p - \alpha$ for some $0 < \alpha < \bar{h}_p$, we complete the proof since $(x' - \bar{x})f(\bar{x}) = (h'_p - \bar{h}_p) f_p(\bar{x}) = -\alpha f_p(\bar{x}) \leq 0$.

Case 2: $\bar{x} \in \bar{\Delta}$ and $\bar{h}_p < d_i$ for all $p \in P_i$ and $i \in I$, which implies that

$$\sum_{i \in I} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I} |P_i| d_i < 0.$$

Also, $\bar{x} \in \bar{\Delta}$ therefore:

$$\sum_{i \in I} (\tau_i - \bar{u}_i) = \gamma \left(\sum_{i \in I} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I} |P_i| d_i \right) < 0 \quad (4.1)$$

which implies

$$\sum_{i \in I} \bar{u}_i > \sum_{i \in I} \tau_i \geq 0. \quad (4.2)$$

We partition the set I into two disjoint sets I_1 and I_2 such that

$$I_1 = \{i \in I: \bar{u}_i > \tau_i\}$$

$$I_2 = \{i \in I: \bar{u}_i \leq \tau_i\}.$$

Clearly, $I_1 \cup I_2 = I$, $I_1 \cap I_2 = \emptyset$ and $I_1 \neq \emptyset$ because of (4.2). Now (4.1) can be written as:

$$\sum_{i \in I_1} (\tau_i - \bar{u}_i) + \sum_{i \in I_2} (\tau_i - \bar{u}_i) = \gamma \left(\sum_{i \in I_1} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I_1} |P_i| \cdot d_i \right) + \gamma \left(\sum_{i \in I_2} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I_2} |P_i| \cdot d_i \right)$$

But, in this equation all terms except $\sum_{i \in I_2} (\tau_i - \bar{u}_i)$ are negative and $\gamma > 1$, thus:

$$\sum_{i \in I_1} (\tau_i - \bar{u}_i) < \gamma \left(\sum_{i \in I_1} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I_1} |P_i| \cdot d_i \right) < 0 \quad (4.3)$$

On the other hand, for any $i \in I_1$ we have:

$$f_p(\bar{x}) = T_p(\bar{h}) - \bar{u}_i \leq \tau_i - \bar{u}_i < 0 \quad \text{for all } p \in P_i$$

implying that

$$\sum_{p \in P_i} f_p(\bar{x}) \leq \sum_{p \in P_i} (\tau_i - \bar{u}_i) = |P_i| (\tau_i - \bar{u}_i) < 0$$

and that

$$\sum_{i \in I_1} (1/|P_i|) \sum_{p \in P_i} f_p(\bar{x}) \leq \sum_{i \in I_1} (\tau_i - \bar{u}_i) < 0. \quad (4.4)$$

Also, for any $i \in I_1$ we have:

$$g_i(\bar{x}) = \sum_{p \in P_i} \bar{h}_p - D_i(\bar{u}) > \sum_{p \in P_i} \bar{h}_p - d_i$$

and, since $|P_i| \geq 1$,

$$g_i(\bar{x}) > \sum_{p \in P_i} \bar{h}_p - |P_i|d_i \quad (4.5)$$

Now, if $g_i(\bar{x}) \geq 0$ for some $i \in I_1$ then the proof is clear. To see this, take $h' = \bar{h}$, $u'_j = \bar{u}_j$ for $j \neq i$ and $j \in I_1$, and $u'_i = \bar{u}_i - \alpha$ for some $0 < \alpha < \bar{u}_i$. Therefore, suppose that $g_i(\bar{x}) < 0$ for all $i \in I_1$. Adding (4.5) for all $i \in I_1$ gives:

$$0 > \sum_{i \in I_1} g_i(\bar{x}) > \sum_{i \in I_1} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I_1} |P_i|d_i \quad (4.6)$$

Combining (4.3), (4.4) and (4.6) implies that

$$\frac{\sum_{i \in I_1} (1/|P_i|) \sum_{p \in P_i} f_p(\bar{x})}{\sum_{i \in I_1} g_i(\bar{x})} > \frac{\sum_{i \in I_1} (\tau_i - \bar{u}_i)}{\sum_{i \in I_1} \sum_{p \in P_i} \bar{h}_p - \sum_{i \in I_1} |P_i| d_i} \geq \gamma > 1$$

Now, define $x' = (h', u')$ as follows:

$$\left\{ \begin{array}{ll} h'_p = \bar{h}_p + \alpha/|P_i| & \text{for } p \in P_i \text{ and } i \in I_1 \\ h'_p = \bar{h}_p & \text{for } p \in P_i \text{ and } i \in I_2 \\ u'_i = \bar{u}_i - \alpha \beta & \text{for } i \in I_1 \\ u'_i = \bar{u}_i & \text{for } i \in I_2 \end{array} \right.$$

where α and β are constants satisfying:

$$\left\{ \begin{array}{l} \gamma < \beta \leq \left(\sum_{i \in I_1} (1/|P_i|) \sum_{p \in P_i} f_p(\bar{x}) \right) / \left(\sum_{i \in I_1} g_i(\bar{x}) \right) \\ 0 < \alpha \leq \text{Min}_{i \in I_1} (\bar{u}_i / \beta). \end{array} \right.$$

To complete the proof, it is enough to show that x' satisfies Kajimas' conditions. First, it is clear that $x' \geq 0$ and $x' \in \Delta'$ because:

$$\begin{aligned}
\gamma \left(\sum_{i \in I} \sum_{p \in P_i} h'_p \right) + \sum_{i \in I} u'_i &= \gamma \left(\sum_{i \in I} \sum_{p \in P_i} \bar{h}_p \right) + \sum_{i \in I} \bar{u}_i \\
&+ \gamma \left(\sum_{i \in I_1} \sum_{p \in P_i} \frac{\alpha}{|P_i|} \right) - \sum_{i \in I_1} \alpha \beta \\
&= \delta + \gamma(\alpha \cdot |I_1|) - \alpha \cdot \beta \cdot |I_1| \\
&= \delta + \alpha |I_1| (\gamma - \beta) < \delta .
\end{aligned}$$

Also, $(x' - \bar{x}) \cdot e < 0$ where e is a vector of n ones because:

$$\begin{aligned}
(x' - \bar{x}) \cdot e &= \sum_{i \in I_1} \sum_{p \in P_i} (\alpha / |P_i|) - \sum_{i \in I_1} \alpha \beta \\
&= \alpha |I_1| - \alpha \beta \cdot |I_1| \\
&= \alpha |I_1| (1 - \beta) < 0 .
\end{aligned}$$

The last inequality is valid because $\beta > \gamma > 1$.

Finally, $(x' - \bar{x}) \cdot f(\bar{x}) \leq 0$ because:

$$\begin{aligned}
(x' - \bar{x}) f(\bar{x}) &= \sum_{i \in I_1} \sum_{p \in P_i} (\alpha / |P_i|) \cdot f_p(\bar{x}) - \sum_{i \in I_1} \alpha \beta g_i(\bar{x}) \\
&= \alpha \left[\sum_{i \in I_1} (1 / |P_i|) \sum_{p \in P_i} f_p(\bar{x}) - \beta \cdot \sum_{i \in I_1} g_i(\bar{x}) \right] \leq 0
\end{aligned}$$

this inequality is valid by definition of β , and the proof is complete. ■

THEOREM 4.5: (Existence) Suppose (N, A) is a strongly connected network. Suppose that $t_\alpha : R_+^{n_1} \rightarrow R_+^I$ is a positive continuous function for all $\alpha \in A$. Also suppose that for all $i \in I$, $D_i : R_+^{n_2} \rightarrow R_+^I$ is a non-negative continuous function that is bounded from above. Then the user-equilibrium system (3.1) has a solution.

PROOF: Theorems 3.2 and 4.4 immediately imply the proof of this theorem. ■

Recently Asmuth [T-A4] has shown how the user-equilibrium problem can be formulated as a *stationary point* problem, and has given existence and uniqueness proofs for a more general type of volume delay and demand functions.

EXISTENCE THEOREM 4.6: (Asmuth) Suppose (N, A) is a strongly connected network. Suppose

- i) t_α , the volume delay function, is a positive, convex valued and upper semi-continuous point-to-set map on $\{h \mid h \geq 0\}$ and
- ii) D_i , the demand function, is a non-negative, bounded, convex valued and upper semi-continuous point-to-set map on $\{u \mid u \geq 0\}$.

Then a solution to the user-equilibrium problem exists.

Although Asmuth's proof differs from ours, the underlying ideas are the same. He uses Saigal's results [C-S1] to give a necessary condition

for the existence of the solution for the stationary point problem, instead of Kojimas' results which we used.

4.3 UNIQUENESS

Although there is a straightforward method to show the uniqueness of the solution under some assumptions, it is usually difficult to extend the results to more general cases. In this section we use ideas similar to what Asmuth [T-A4] has used for the stationary point problem to show that the nonlinear complementarity problem and user-equilibrium problem has a unique solution under strictly monotonicity assumptions. As we showed previously in Section 4.2, the path flows usually are not unique and only the arc volumes will be unique. Also, in this section we extend the results for situations in which the link flows are not unique, but the path travel times, the accessibility variables u_i , are unique.

To facilitate our study in this section, we represent the traffic equilibrium problem in a matrix form. Let v_a denote the total flow on arc a , that is, $v_a = \sum_{i \in I} \sum_{p \in P_i} \delta_{ap} \cdot h_p$, and let v with dimension $|A|$ denote the vector of arc flows. Then $t_a(h) = t_a(v)$ for all $a \in A$.

Also, let $t(v)$ be the vector of volume delay functions and $D(u)$ be the vector of demand functions. Let $\Delta = (\delta_{ap})$ be the arc-path incidence matrix with dimension $|A| \times n_1$ and let $\Gamma = (\gamma_{pi})$ be the path-O-D pair incidence matrix with dimension $n_1 \times n_2$, i.e., $\gamma_{pi} = 1$ when path p joins O-D pair i and $\gamma_{pi} = 0$ otherwise.

Then the user-equilibrium problem can be written as follows:

$$\left\{ \begin{array}{l} (\Delta^T \cdot t(\Delta h) - \Gamma \cdot u) \cdot h = 0 \\ \Delta^T \cdot t(\Delta h) - \Gamma \cdot u \geq 0 \\ \Gamma^T \cdot h - D(u) = 0 \\ h \geq 0 \quad , \quad u \geq 0 \end{array} \right.$$

Now let $G(x) = (t(\Delta h), -D(u))$ where $x = (h, u)$ and $G: \mathbb{R}_+^n \rightarrow \mathbb{R}^m$ with $n = n_1 + n_2$ and $m = |A| + n_2$. Also let:

$$\bar{\Delta} = \begin{pmatrix} \Delta & 0 \\ 0 & I' \end{pmatrix} \quad \text{and} \quad \bar{\Gamma} = \begin{pmatrix} 0 & -\Gamma \\ \Gamma^T & 0 \end{pmatrix}$$

with dimensions $m \times n$ and $n \times n$ respectively, and I' is the identity matrix with dimension $n_2 \times n_2$.

Then, the corresponding nonlinear complementarity problem can be written as follows:

$$\left\{ \begin{array}{l} (\bar{\Delta}^T G(\bar{\Delta}x) + \bar{\Gamma}x) \cdot x = 0 \\ \bar{\Delta}^T G(\bar{\Delta}x) + \bar{\Gamma}x \geq 0 \\ x \geq 0 \end{array} \right. \quad (4.7)$$

It is easy to show that $\bar{\Delta}^T G(\bar{\Delta}x) + \bar{\Gamma}x = F(x)$ where F has been defined in the Section 3.3. Therefore (4.7) is equivalent to the system 3.3.

The following lemma and its proof is very similar to results of Asmuth [T-A4] concerning a stationary point problem, and is included here for completeness.

LEMMA 4.1: Let $K \subset \mathbb{R}^n$, let B be an $m \times n$ matrix, and let $L = \{Bx \mid x \in K\} \subset \mathbb{R}^m$. Suppose that $g: L \rightarrow \mathbb{R}^m$ is strictly monotone on L . Let A be an $n \times n$ positive semi-definite matrix. Define $f: \mathbb{R}^n \rightarrow \mathbb{R}^n$ by $f(x) = B^T g(Bx) + Ax$. Then the set of solutions $(x, f(x))$ to the complementarity problem $x \geq 0$, $f(x) \geq 0$ and $xf(x) = 0$ is convex and Bx has the same value for all of these solutions.

PROOF: Suppose that x^1 and x^2 , $x^1 \neq x^2$, solve the nonlinear complementarity problem, i.e.

$$x^i \geq 0, f(x^i) \geq 0 \text{ and } x^i f(x^i) = 0 \quad \text{for } i = 1, 2$$

then

$$x^2 f(x^1) \geq 0, x^1 f(x^2) \geq 0$$

and consequently

$$\begin{cases} (x^2 - x^1) f(x^1) \geq 0 \\ (x^1 - x^2) f(x^2) \geq 0 \end{cases}$$

which implies that

$$(x^1 - x^2) (f(x^1) - f(x^2)) \leq 0$$

or

$$(x^1 - x^2) (B^T g(Bx^1) + Ax^1 - B^T g(Bx^2) - Ax^2) \leq 0$$

or

$$(x^1 - x^2) [B^T (g(Bx^1) - g(Bx^2))] + (x^1 - x^2) A (x^1 - x^2) \leq 0.$$

Since A is positive semi-definite, $(x^1 - x^2)A(x^1 - x^2) \geq 0$, implying that

$$(x^1 - x^2)[B^T(g(Bx^1) - g(Bx^2))] \leq 0.$$

or

$$(Bx^1 - Bx^2)(g(Bx^1) - g(Bx^2)) \leq 0. \quad (4.8)$$

But g is strictly monotone on L , therefore $Bx^1 = Bx^2$.

To prove the convexity of the solution set, let $\lambda \in [0,1]$ and let $x = \lambda x^1 + (1-\lambda)x^2$. Then clearly $x \geq 0$ and also, by the first part of the proof $Bx = \lambda Bx^1 + (1-\lambda)Bx^2 = Bx^1 = Bx^2$. Therefore

$$\begin{aligned} f(x) &= B^T g(Bx) + Ax \\ &= \lambda B^T g(Bx) + (1-\lambda)B^T g(Bx) + A(\lambda x^1 + (1-\lambda)x^2) \\ &= \lambda B^T g(Bx^1) + \lambda Ax^1 + (1-\lambda)B^T g(Bx^2) + (1-\lambda)Ax^2 \\ &= \lambda(B^T g(Bx^1) + Ax^1) + (1-\lambda)(B^T g(Bx^2) + Ax^2) \\ &= \lambda f(x^1) + (1-\lambda)f(x^2). \end{aligned}$$

But $\lambda \in [0,1]$ and $f(x^i) \geq 0$ for $i = 1,2$; therefore $f(x) \geq 0$. Also, clearly $xf(x) \geq 0$ and

$$xf(x) = x(B^T g(Bx) + Ax)$$

or

$$\begin{aligned}
f(x) &= xB^T g(Bx) + xAx \\
&= \lambda xB^T g(Bx) + (1-\lambda)xB^T g(Bx) + xAx \\
&= \lambda x^1 B^T g(Bx^1) + (1-\lambda)x^2 B^T g(Bx^2) + xAx.
\end{aligned}$$

Also,

$$\begin{aligned}
xAx &= (\lambda x^1 + (1-\lambda)x^2) A(\lambda x^1 + (1-\lambda)x^2) \\
&= \lambda^2 x^1 A x^1 + \lambda(1-\lambda)x^1 A x^2 + \lambda(1-\lambda)x^2 A x^1 + (1-\lambda)^2 x^2 A x^2 \\
&= \lambda x^1 A x^1 + (1-\lambda)x^2 A x^2 - \lambda(1-\lambda)(x^1 - x^2)A(x^1 - x^2)
\end{aligned}$$

and thus,

$$\begin{aligned}
xf(x) &= \lambda x^1 [B^T g(Bx^1) + Ax^1] + (1-\lambda)x^2 [B^T g(Bx^2) + Ax^2] \\
&\quad - \lambda(1-\lambda)(x^1 - x^2) A(x^1 - x^2) \\
&= \lambda x^1 f(x^1) + (1-\lambda)x^2 f(x^2) - \lambda(1-\lambda)(x^1 - x^2)A(x^1 - x^2)
\end{aligned}$$

But, by assumption, $x^1 f(x^1) = 0$ and $x^2 f(x^2) = 0$; therefore,

$$xf(x) = -\lambda(1-\lambda)(x^1 - x^2)A(x^1 - x^2).$$

This implies that $xf(x) \leq 0$ because A is positive semi-definite. Also, we showed previously that $xf(x) \geq 0$. Consequently, $xf(x) = 0$ and x is a complementarity solution. ■

THEOREM 4.7: (Uniqueness) *For a strongly connected network (N, A) , suppose that t , the vector of the volume delay functions, and $-D$, the vector of the negative demand functions, are strictly monotone. Then the arc volumes, v , and the accessibility vector u for the equilibrium problem (3.1) are unique, and the set of equilibrium path flows are convex.*

PROOF: With the notation used in system (4.7), we have that $G = (t, -D)$ is strictly monotone on $L = \{\bar{\Delta}x = (v, u) : x = (h, u) \in \mathbb{R}^n\}$. Also, since $\bar{\Gamma}$ is skew systematic, it is positive semi-definite. (In fact, for any $x = (h, u)$ we have:

$$x\bar{\Gamma}x = (h, u) \begin{bmatrix} 0 & -\Gamma \\ \Gamma^T & 0 \end{bmatrix} (h, u)^T = -h\Gamma u + u \Gamma^T h = 0).$$

Thus, with $g = G$, $f = F$, $B = \bar{\Delta}$ and $A = \bar{\Gamma}$, by Lemma 4.1, $\bar{\Delta}x = (v, u)$ is unique for the nonlinear complementarity system (3.3) which implies that the arc volume v and the accessibility variable u are unique for the user-equilibrium problem (3.1). Also, the set of solutions $x = (h, u)$ to the nonlinear complementarity problem (3.3) is convex, which implies that the set of path flows, h , is convex for the user-equilibrium problem. ■

Notice that the required conditions for Theorem 4.7 are completely different than the conditions in Karamardian's Theorem, 4.2. Here, we require that the vector of volume delay functions is strictly monotone in terms of arc volume v , while in 4.2 we require that the nonlinear complementarity function, $F(x)$, is strictly monotone in terms of path flows, h . As we showed in Example 4.1, the path flows, h , might not be unique even

if we assume that all t_a and $-D_i$ are strictly monotone.

Note that both of the functions t and $-D$ are required to be strictly monotone in Theorem 4.7 to insure the uniqueness of (v,u) . In the next Theorem we show that this restriction on D can be relaxed, and that uniqueness of u is maintained if either of t or $-D$ is strictly monotone.

THEOREM 4.8: *For a complete network (N, A) , suppose that t and $-D$ are both monotone functions. If either of t or $-D$ is strictly monotone, then u is unique. Also, if t is strictly monotone, then (v,u) is unique.*

PROOF: Suppose that $x^1 = (h^1, u^1)$ and $x^2 = (h^2, u^2)$, $x^1 \neq x^2$, are two solutions. As in lemma 4.1, with $g = G$, $f = F$, $B = \bar{\Delta}$ and $A = \bar{\Gamma}$, we have by equation (4.8)

$$(\bar{\Delta}x^1 - \bar{\Delta}x^2)(G(\bar{\Delta}x^1) - G(\bar{\Delta}x^2)) \leq 0.$$

But $G = (t, -D)$ is monotone because it has monotone components, i.e.

$$(\bar{\Delta}x^1 - \bar{\Delta}x^2)(G(\bar{\Delta}x^1) - G(\bar{\Delta}x^2)) \geq 0.$$

Therefore

$$(\bar{\Delta}x^1 - \bar{\Delta}x^2)(G(\bar{\Delta}x^1) - G(\bar{\Delta}x^2)) = 0. \quad (4.9)$$

By substituting for $\bar{\Delta}$, $x = (h,u)$ and $G = (t, -D)$ we obtain:

$$(\Delta h^1 - \Delta h^2)(t(\Delta h^1) - t(\Delta h^2)) + (u^1 - u^2)(-D(u^1) + D(u^2)) = 0 \quad (4.10)$$

But both t and $-D$ are monotone functions, thus each term in (4.10) is zero; that is,

$$(\Delta h^1 - \Delta h^2)(t(\Delta h^1) - t(\Delta h^2)) = 0 \quad (4.11)$$

$$- (u^1 - u^2)(D(u^1) - D(u^2)) = 0 \quad (4.12)$$

If $-D$ is strictly monotone, then equation (4.12) implies that $u^1 = u^2$, or u is unique.

Now, suppose that t is strictly monotone. Then (4.11) implies that $v^1 = \Delta h^1 = \Delta h^2 = v^2$, or that the arc volume vector v is unique. But uniqueness of arc volume vector implies that the travel time, $t_a(v)$, on each arc is unique, which obviously implies that u is unique. ■

When all the traffic from different origins have the same effect on the travel time of each arc, and there is no interaction between opposing lanes of two-way traffic or right or left turn penalties, or in other words, t_a is a function only of the total volume in the arc, then the strictly monotone condition on t can be relaxed for the uniqueness results.

COROLLARY 4.1: (Special case) *For a strongly connected network (N, A) , suppose that each t_a is a function only of v_a , and that it is monotone. Also, suppose that $-D$ is monotone. Then u is unique.*

PROOF: Obviously t , the vector of the volume delay functions, is monotone because each of its components is monotone. Thus equation (4.11) in Theorem 4.8 is true,

$$(\Delta h^1 - \Delta h^2)(t(\Delta h^1) - t(\Delta h^2)) = 0. \quad (4.13)$$

But since each component of t is monotone, (4.13) can be separated into a single form for each arc:

$$(v_a^1 - v_a^2)(t_a(v_a^1) - t_a(v_a^2)) = 0.$$

This implies that $t_a(v_a^1) = t_a(v_a^2)$, or that the travel time on each arc is unique and, consequently, that u , the minimum path travel time, is unique. ■

CHAPTER 5

COMPUTING AN EQUILIBRIUM

5.1 INTRODUCTION

In this chapter we discuss some of the basic approaches that have been applied to find an equilibrium solution to the traffic assignment problem. We consider exclusively the deterministic case. For a discussion of stochastic approaches, see Sheffi [T-S1] and the references that he cites.

Almost all previous efforts can be classified as being:

i) Heuristic

or ii) mathematical programming-based.

In this chapter we briefly discuss approaches from each category, and their limitations. We conclude the chapter by introducing a new linearization approach that is based upon mathematical programming, although we have not been able to prove its convergence.

5.2 HEURISTIC TECHNIQUES

Since 1952, a large number of algorithms have been developed for the traffic assignment problem. Most of the earlier techniques have been based upon intuition, without considering congestion effects or any formal concept of equilibrium. The goal of these approaches was to assign flow between different paths so that the paths have almost equal travel times.

The first of these algorithms is the "diversion curve" technique [T-M5, T-M9, T-W5] in which the total number of trips between an origin-

destination pair are divided between two routes, one an expressway or the like, and the other an arterial, or equivalent, highway. These techniques are only suitable for small networks. The next generation of this type of algorithm is the "all-or-nothing" or "desire" assignment technique used in 1958 in the Detroit Transportation Study [T-D6]. None of these algorithms incorporates congestion effects or an equilibrium concept.

The first attempt to account for the capacity of the system is known as the "capacity restrained" technique [T-C3, T-D13, T-I1, T-I2, T-S3]. Manheim and Martin [T-M3], in a procedure known as the "incremental traffic assignment" technique, were the first to account for both congestion and equilibrium concepts in the context of the traffic assignment problem. This procedure tries to load the network by a small percentage of flow incrementally, updating the system performance and congestion measures after each flow change.

Recently, more sophisticated heuristic techniques have been developed and applied to the large networks (see Jacobson [T-J1] or Manheim and Ruiter [T-M4], for example). However, neither is there any good theoretical justification to guarantee the convergence of these algorithms, nor is there enough computational experience to show how good they perform in practice.

5.3 MATHEMATICAL PROGRAMMING TECHNIQUES

As we pointed out in the previous chapter, the traffic equilibrium problem can be formulated as a non-linear complementarity problem or as a fixed-point problem. Therefore, at least theoretically, any

algorithm for solving these problems might be used to solve the traffic equilibrium problem. Also, any non-linear complementarity problem or fixed-point problem can, again in theory, be visualized as an equivalent optimization problem (see Todd [C-T1]). Therefore, an optimization algorithm might be used to solve the problem.

Unfortunately, the limitations on existing algorithms in terms of both the size of the problems that they can solve, especially for the fixed-point and complementarity approaches, and, in terms of the required assumptions, especially for the optimization-based approaches, makes it almost impossible to apply them to solve any real-life traffic equilibrium problem. In section 5.3.1 we briefly discuss the validity of these algorithms and review the efforts of various researchers to use these techniques.

However, under some mild assumptions, the equilibrium problem can be formulated as special optimization problems for which there are efficient algorithms currently available. In section 5.3.2, we discuss this method and its generalizations.

5.3.1 Fixed-Point Techniques

In the literature, there are many algorithms for solving fixed-point and non-linear complementarity problems [C-F1, C-L3, C-K4, C-S2, C-T1, C-T2]. Generally, these algorithms are based upon some division scheme that subdivides the working region into a number of simplexes, and then use some clever search (or pivoting) procedure to move among the simplexes until one is found that approximates a fixed-point (see Scarf [C-S2] or Todd [C-T1]). A major advantage of these algorithms

is that they require very few assumptions on the problem, and have the capability of dealing with highly non-linear problems. Another advantage is that they can provide solutions to within any prescribed degree of accuracy.

However, naturally, the generality and power of these algorithms creates some disadvantages as well. One disadvantage is a relatively high solution time, which limits the size of problem that they can solve. For example, the solution time is on the order of a couple of seconds for a five-variable problem (see Kojima [C-K4] or Lutti [C-L3], and a couple of minutes for a problem with 50 variables. Another disadvantage is that these algorithms, because of their generality, do not exploit any inherent properties of the problem under study.

For the transportation applications that we are considering, the variables for the associated non-linear complementarity problem are the available paths in the network. Even for a small-sized network with 100 nodes and 1000 arcs, the number of paths is on the order of millions, although most of them have zero flow. Not even the most efficient general purpose algorithm for the non-linear complementarity problem would be able to solve a problem of this size. Also, generally, the transportation applications do not require the degree of accuracy that these algorithms are capable of providing.

Finally, regardless of what kind of algorithm is used to solve the equilibrium problem, knowledge of shortest paths is essential. Since there are a number of extremely efficient algorithms available for finding shortest paths, any efficient algorithm for the traffic

equilibrium problem might be expected to incorporate shortest path computations as a subroutine. Fixed-point algorithms, generally, do not take advantage of this aspect of the equilibrium problem.

However, at least theoretically, the fixed-point algorithm will solve any general equilibrium problem, even when other algorithms might fail.

In 1977, Kuhn [T-H2] devised a fixed-point method, equipped with a special pivoting scheme, to solve equilibrium problems with fixed demands and with separable volume delay functions. Applications of the algorithm to a small 4-node equilibrium problem required 7 seconds of computation time and provided a very accurate solution. In 1977, Aashtiani [T-A1] formulated a more general equilibrium problem as a non-linear complementarity problem and studied the existence of solutions. Asmuth [T-A4] proposed a similar model which included point-to-set volume delay functions and demand functions. He proposed a fixed-point algorithm and applied it to some small examples that could not be solved by any other method. The algorithm found accurate solutions, but, again, the solution time was so high that it does not encourage the application of this algorithm to large, real-life transportation problems.

5.3.2 Optimization Technique

In 1956, Beckman, McGuire, and Winsten [T-B1], by imposing the following restrictions, were the first to formulate the traffic equilibrium problem as an optimization problem.

Problem Restrictions:

i) The link performance functions are independent, i.e.,

$$t_a(f) = t_a(f_a) \quad \text{for all } a \in A$$

where

$$f_a = \sum_{i \in I} \sum_{p \in P_i} \delta_{ap} h_p.$$

ii) The demand functions are independent, i.e.,

$$D_i(u) = D_i(u_i) \quad \text{for all } i \in I.$$

iii) $t_a(f_a)$, for all $a \in A$, is an increasing function.

iv) $D_i(u_i)$, for all $i \in I$, is a strictly decreasing function.

By imposing these restrictions[†], they showed that the Kuhn-Tucker condition for the following convex minimization problem is equivalent to the user-equilibrium system corresponding to the traffic equilibrium law:

$$\text{Minimize } \sum_{a \in A} \int_0^{f_a} t_a(x) dx - \sum_{i \in I} \int_0^{d_i} w_i(y) dy$$

subject to:

[†]In this section we assume that $t(f)$ and $D(u)$ are positive, continuous functions and that they are differentiable.

$$(5.1) \left\{ \begin{array}{l} \sum_{p \in P_i} h_p - d_i = 0 \quad \text{for all } i \in I \quad (5.1a) \\ f_a = \sum_{i \in I} \sum_{p \in P_i} \delta_{ap} h_p \quad \text{for all } a \in A \quad (5.1b) \\ h \geq 0 \quad (5.1c) \\ d \geq 0 \quad (5.1d) \end{array} \right.$$

where $w_i(d_i)$ is the inverse function of the demand function $D_i(u_i)$; it always exists because $D_i(u_i)$ is strictly decreasing. The dual variables corresponding to the first set of constraints (5.1a) are the accessibility variables, u_i .

In addition, they showed that when $t_a(f_a)$ is strictly increasing, then the minimization problem has a unique solution in terms of f and u .

Notice that, although the above formulation has been given for a single-mode traffic assignment problem, this formulation is valid for the multi-modal case as long as the assumptions (i) and (ii) hold. In fact, the problem can be separated into distinct minimization problems, one associated with each mode.

In the last decade, a number of researchers [T-F4-7, T-G5, T-L2-3, T-N2-6] have developed algorithms based upon this formulation for both fixed and elastic demand functions. Among these algorithms is the one developed by Leblanc [T-L2-3] using the Frank-Wolfe feasible direction method [T-F9] for fixed demands. Nguyen [T-N3] developed an algorithm based upon the convex simplex method. Later Nguyen and Florian [T-F6],

using Benders decomposition, extended the range of applications to include elastic demand functions. These algorithms have been applied with success to solve some small real-life problems.

The first attempt to generalize the equivalent minimization approach to multi-class users, at least theoretically, was made by Dafermos [T-D2]. She relaxed the restrictions (i) and (iii) as follows:

i)' $t_a(f)$ is a function of the vector of f and

$\nabla t(f)$ is symmetric. Here t is the vector of t_a .

iii)' $\nabla t(f)$ is a positive definite matrix.

For the fixed demand function, Dafermos proposed a minimization problem of the form:

$$\text{Minimize } S(f)$$

subject to:

$$(5.2) \left\{ \begin{array}{l} \sum_{p \in P_i} h_p - d_i = 0 \quad \text{for all } i \in I \\ f_a = \sum_{i \in I} \sum_{p \in P_i} \delta_{ap} \cdot h_p \quad \text{for all } a \in A \\ h \geq 0 \\ f \geq 0 \end{array} \right.$$

She showed that this minimization problem is equivalent to the equilibrium problem if,

$$\frac{\partial S(f)}{\partial f_a} = t_a(f) \quad \text{for all } a \in A$$

and showed that this system of differential equations has a solution if, and only if, $\nabla t(f)$ is symmetric. Then S is the specification for the following line integral:

$$S(f) = \int_0^f t(x) dx$$

or

$$S(f) = \int_0^f \sum_{a \in A} t_a(x) dx_a .$$

Furthermore, Dafermos showed that $S(f)$ is a strictly convex function if, and only if, $\nabla t(f)$ is positive definite.

To generalize the minimization approach to a more general setting, we permit not only $t_a(f)$ to be a function of other link flows, but also let $D_i(u)$ be a function of the full vector u . In other words, we include any destination or mode choice demand function in the model. Moreover, we require new restrictions that are weaker than the previously quoted assumptions, namely:

- i)' $t_a(f)$ is a function of the vector of f and $\nabla t(f)$ is symmetric. Here t is the vector with components t_a .
- ii)' $D_i(u)$ is a function of the vector u and $\nabla D(u)$ is symmetric. D denotes the vector with components D_i .
- iii)' $\nabla t(f)$ is a positive definite matrix.
- iv)' $-\nabla D(u)$ is a positive definite matrix with $\frac{-\partial D_i}{\partial u_j} \leq 0$ for $i \neq j$ and $-\nabla^2 D_i(u)$ is a positive semi-definite matrix for all $i \in I$.

We propose the following minimization problem:

$$\begin{aligned} \text{Minimize} \quad \psi(h,u) &\equiv S(f) + \theta(u) \\ &= \oint_0^f t(x)dx - uD(u) + \oint_0^u D(v)dv \end{aligned}$$

subject to:

$$(5.3) \left\{ \begin{array}{ll} G_i(h,u) \equiv \sum_{p \in P_i} h_p - D_i(u) = 0 & \text{for all } i \in I \quad \lambda_i \\ f_a = \sum_{i \in I} \sum_{p \in P_i} \delta_{ap} \cdot h_p & \text{for all } a \in A \\ h_p \geq 0 & \text{for all } p \in P_i, \quad i \in I \quad \mu_p \\ u_i \geq 0 & \text{for all } i \in I \quad \gamma_i \end{array} \right.$$

where \oint denotes a line integral; the symmetry assumptions (i.e., (i)' and (iii)') guarantee the existence of the line integrals.

If we substitute for the variable f and let λ , μ , and γ be the dual variables for the constraints (5.3), then the Kuhn-Tucker conditions for the above problem are:

$$(5.4) \left\{ \begin{array}{l} \nabla_h \psi(h,u) + \lambda \nabla_h G(h,u) - \mu = 0 \\ \nabla_u \psi(h,u) + \lambda \nabla_u G(h,u) - \gamma = 0 \\ G(h,u) = 0 \\ \mu h = 0 \\ \gamma u = 0 \\ h \geq 0, u \geq 0, \mu \geq 0, \gamma \geq 0. \end{array} \right.$$

Dafermos provided intuition for choosing $S(f)$. To motivate our choice of $\theta(u)$, let us look at the second equation in (5.4),

$$\frac{\partial \theta(u)}{\partial u_i} - \sum_{j \in I} \lambda_j \frac{\partial D_j}{\partial u_i} - \gamma_i = 0 \quad \text{for all } i \in I \quad (5.5)$$

where,

$$\begin{aligned} \frac{\partial \theta(u)}{\partial u_i} &= \frac{\partial}{\partial u_i} [-uD(u) + \int_0^u D(v) dv] \\ &= -D_i(u) - \sum_{j \in I} u_j \frac{\partial D_j(u)}{\partial u_i} + D_i(u) \end{aligned}$$

or

$$\frac{\partial \theta(u)}{\partial u_i} = - \sum_{j \in I} u_j \frac{\partial D_j(u)}{\partial u_i} .$$

We will show that when $-\nabla D(u)$ is positive definite, the above choice for $\theta(u)$ guarantees that $u = -\lambda$ and $\gamma = 0$. Thus (5.5) becomes:

$$\frac{\partial \theta(u)}{\partial u_i} + \sum_{j \in I} u_j \frac{\partial D_j(u)}{\partial u_i} = 0 \quad (5.6)$$

It is easy to see that symmetry of $\nabla D(u)$ implies that $\theta(u) = -uD(u) + \int_0^u D(v) dv$ is a solution for the system of differential equations (5.6), and this motivates our choice of $\theta(u)$.

Involving assumptions (i)' and (ii)' in (5.4), it becomes:

$$\begin{aligned}
 (5.7) \quad & \left. \begin{aligned}
 & \sum_a \delta_{ap} t_a(f) + \lambda_i - \mu_p = 0 \quad \text{for all } p \in P_i, i \in I \\
 & \sum_j u_j \frac{\partial D_i}{\partial u_j} + \sum_j \lambda_j \frac{\partial D_i}{\partial u_j} + \gamma_i = 0 \quad \text{for all } i \in I \\
 & \sum_{p \in P_i} h_p - D_i(u) = 0 \quad \text{for all } i \in I \\
 & f_a = \sum_{i \in I} \sum_{p \in P_i} \delta_{ap} h_p \quad \text{for all } a \in A \\
 & \mu h = 0 \\
 & \gamma u = 0 \\
 & h \geq 0, u \geq 0, \mu \geq 0, \gamma \geq 0.
 \end{aligned} \right\}
 \end{aligned}$$

It is easy to see that when $u = -\lambda$ and $\gamma = 0$, then (5.7) is equivalent to the nonlinear complementarity problem associated with the equilibrium system. To prove that assumptions (iii)' and (iv)' guarantee that $u = -\lambda$ and $\gamma = 0$, we need the following lemmas.

LEMMA 5.1: *If A is a positive definite matrix, then $Ax = 0$ implies that $x = 0$.*

PROOF: Suppose it is not, and $x \neq 0$, then

$$x^T Ax = x^T \cdot 0 = 0$$

but this is a contradiction, because A is positive definite. ■

LEMMA 5.2: Suppose that A is a positive definite matrix and that $a_{ij} \leq 0$ for all $i \neq j$. Then $Ax = \delta$ has a non-negative solution for any $\delta \geq 0$.

PROOF: We prove the result by induction. For $n = 1$ it is clear.

Suppose that it is true for $n = k$. We show that it is true for $n = k + 1$.

It is clear that not all x_i can be negative because $x^T Ax = x^T \delta \leq 0$, which contradicts the assumption that A is positive definite. Thus, suppose that $x_m \geq 0$ for some m .

By eliminating the m^{th} row and taking the m^{th} column to the other side of the equation we get the following system of equations:

$$\sum_{j \neq m} a_{ij} x_j = \delta_i - a_{im} x_m \quad \text{for all } i \neq m.$$

Clearly, the matrix associated with the above system has all the properties of the original matrix, and also, since $a_{im} \leq 0$ for all $i \neq m$, thus $\delta_i - a_{im} x_m \geq 0$ for all $i \neq m$. Therefore, by induction, the new system of equations has a non-negative solution, which completes the proof. ■

THEOREM 5.1: Suppose that $t(f)$ and $D(u)$ are positive (componentwise) continuous vector functions and, furthermore, that $\nabla t(f)$ and $-\nabla D(u)$ are symmetric and positive definite matrices with $-\frac{\partial D_i(u)}{\partial u_j} \leq 0$ for all $i \neq j$. Then (5.7) is equivalent to the equilibrium system, i.e., $u = -\lambda$ and $\gamma = 0$.

PROOF: First, suppose that $u > 0$. Then the complementarity equation $u\gamma = 0$ implies that $\gamma = 0$. Thus, the second equation in (5.7) becomes,

$$\sum_{j \in I} (u_j + \lambda_j) \frac{\partial D_j(u)}{\partial u_j} = 0 \quad \text{for all } i \in I$$

or

$$(-\nabla D(u))(u + \lambda) = 0 .$$

By lemma 5.1, we have $u + \lambda = 0$ or $u = -\lambda$.

Now, suppose that $u_m = 0$ for some $m \in I$. Since $D_m(u) > 0$, there is at least one path $p' \in P_m$ with $h_{p'} > 0$ which implies, by complementarity, that $\mu_{p'} = 0$, or that

$$\sum_a \delta_{ap'} t_a(f) = -\lambda_m .$$

Also we have,

$$(-\nabla D(u))(u + \lambda) = \gamma \geq 0 .$$

Lemma 5.2 implies that $u + \lambda \geq 0$ and, in particular, that $u_m + \lambda_m \geq 0$, or $\lambda_m \geq 0$. Thus we have

$$\sum_a \delta_{ap'} t_a(f) = -\lambda_m \leq 0$$

which contradicts the assumption $t_a(f) > 0$. This completes the proof. ■

Now the question is, when is the minimization problem (5.3) equivalent to the Kuhn-Tucker system (5.7). Assuming fixed demand, Dafermos showed that a necessary and sufficient condition is that $\nabla t(f)$ be a positive definite matrix, which implies that the objective function is a strictly convex function in terms of the link flow vector f .

For the general case, to have a strictly convex objective function it is sufficient that $\theta(u)$ be a strictly convex function, or equivalently, that $\nabla^2 \theta(u)$ be a positive definite matrix. Previously we showed that:

$$\frac{\partial \theta(u)}{\partial u_i} = - \sum_{j \in I} u_j \frac{\partial D_j(u)}{\partial u_i} \quad \text{for all } i \in I$$

or equivalently,

$$\nabla \theta(u) = - \nabla D(u) \cdot u .$$

Also, for any $i \in I$ and $k \in I$ we have:

$$\frac{\partial}{\partial u_k} \left(\frac{\partial \theta(u)}{\partial u_i} \right) = - \frac{\partial D_k(u)}{\partial u_i} - \sum_{j \in I} u_j \frac{\partial}{\partial u_k} \left(\frac{\partial D_j(u)}{\partial u_i} \right)$$

or equivalently,

$$\nabla^2 \theta(u) = - \nabla D(u)^T - \sum_{j \in I} u_j \nabla^2 D_j(u) .$$

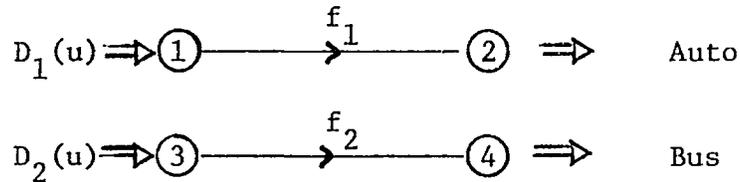
Since $\nabla D(u)$ is symmetric, thus we have:

$$\nabla^2 \theta(u) = - \nabla D(u) - \sum_{j \in I} u_j \nabla^2 D_j(u) .$$

One sufficient condition for $\nabla^2 \theta(u)$ to be positive definite is that, $-\nabla D(u)$ be positive definite and that $-\nabla^2 D_i(u)$ be positive semi-definite for all $i \in I$.

However, it is not clear under what conditions the minimization problem and the equilibrium problem are equivalent. Also, the validity of the assumptions is another question, because even the symmetry assumption for both $\nabla t(f)$ and $\nabla D(u)$ is not valid for real-life problems. This is one reason why this approach might not be applicable for the general case.

EXAMPLE 5.1: Consider a single line network with two modes of transportation, auto and bus, using the same link. To formulate this problem as a single mode case, we duplicate the network and use a separate link for each mode, as follows:



If we let $t_1(f_1, f_2)$ and $t_2(f_1, f_2)$ denote the volume delay functions, then the equilibrium problem can be written simply as:

$$\left\{ \begin{array}{l} f_i = D_i(u_1, u_2) \quad i = 1, 2 \\ u_i = t_i(f_1, f_2) \quad i = 1, 2 \\ f \geq 0, u \geq 0 . \end{array} \right.$$

The corresponding minimization problem would be,

$$\begin{aligned}
 \text{Min} \quad & \int_0^f t_1(x, y) dx + t_2(x, y) dy - u_1 D_1(u_1, u_2) - u_2 D_2(u_1, u_2) \\
 & + \int_0^u D_1(v_1, v_2) dv_1 + D_2(v_1, v_2) dv_2
 \end{aligned}$$

$$\text{subject to: } \left\{ \begin{array}{l} f_i = D_i(u_1, u_2) \quad \text{for } i = 1, 2 \\ f \geq 0, u \geq 0 . \end{array} \right.$$

Now consider a special case with a logit demand function and linear volume delay functions with the following functional forms:

$$\begin{cases} D_1(u_1, u_2) = d \cdot e^{-\theta_1 u_1} / (e^{-\theta_1 u_1} + e^{-\theta_2 u_2}) \\ D_2(u_1, u_2) = d \cdot e^{-\theta_2 u_2} / (e^{-\theta_1 u_1} + e^{-\theta_2 u_2}) \end{cases} \quad \theta > 0$$

$$\begin{cases} t_1(f_1, f_2) = t_1^0 + \alpha_1 f_1 + \beta_1 f_2 \\ t_2(f_1, f_2) = t_2^0 + \alpha_2 f_1 + \beta_2 f_2 \end{cases}$$

where d is the total population and f_1 is the number of passengers using autos and f_2 is the number of passengers using buses.

$\nabla D(u)$ is symmetric if, and only if, $\theta_1 = \theta_2$ and $\nabla t(f)$ is symmetric if, and only if, $\beta_1 = \alpha_2$. However, none of these assumptions are valid for real-life problems, because $\theta_1 = \theta_2$ implies that both modes have equal direct and cross elasticities. Also, $\alpha_2 = \beta_1$ implies that an auto passenger effects a bus passenger as much as a bus passenger effects an auto passenger.

When $\theta_1 = \theta_2$ and $\beta_1 = \alpha_2$, the line integrals become

$$\begin{aligned} \int_0^f t_1(x, y) dx + t_2(x, y) dy \\ &= \int_0^{f_1} t_1(x, 0) dx + \int_0^{f_2} t_2(f_1, y) dy \\ &= t_1^0 f_1 + \frac{1}{2} \alpha_1 f_1^2 + t_2^0 f_2 + \alpha_2 f_1 f_2 + \frac{1}{2} \beta_2 f_2^2 \end{aligned}$$

and

$$\int_0^u D(v) dv = \frac{d}{\theta} \ln \frac{e^{\theta u_1} + e^{\theta u_2}}{2} .$$

Thus the minimization problem becomes:

$$\text{Min } t_1^0 f_1 + t_2^0 f_2 + \frac{1}{2}(\alpha_1 f_1^2 + 2\alpha_2 f_1 f_2 + \beta_2 f_2^2)$$

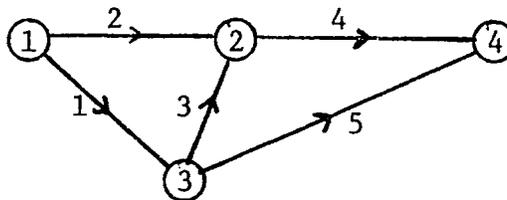
$$-d \left[\frac{u_1 e^{-\theta u_1} + u_2 e^{-\theta u_2}}{e^{-\theta u_1} + e^{-\theta u_2}} - \frac{1}{\theta} \ln \frac{e^{-\theta u_1} + e^{-\theta u_2}}{2} \right]$$

subject to:

$$\left\{ \begin{array}{l} f_i = d \frac{e^{-\theta u_i}}{e^{-\theta u_1} + e^{-\theta u_2}} \quad \text{for } i = 1, 2 \\ f \geq 0, \quad u \geq 0. \end{array} \right.$$

It is easy to see that $\nabla t(f)$ is positive definite if, and only if, $\alpha_1 \beta_2 > \alpha_2 \beta_1$. But $-\nabla D(u)$ cannot be positive definite, although it is positive semi-definite. ■

EXAMPLE 5.2: Consider the following transportation network with 5 one-way links and 4 nodes:



Suppose that there are two types, modes, of movement in the network, auto and truck. The auto movement is between origin-destination pairs 1-3 and 1-4, given by a destination choice demand function. The truck movement is only between O-D pair 1-3.

Also suppose that the volume delay function for each mode on the first link depends on the flow by both modes. And suppose that there is a right turn penalty at node 2, i.e., $t_2(f) = t_2(f_2, f_3)$ and $t_3(f) = t_2(f_2, f_3)$.

To transform the problem into a single mode network we change it as follows:

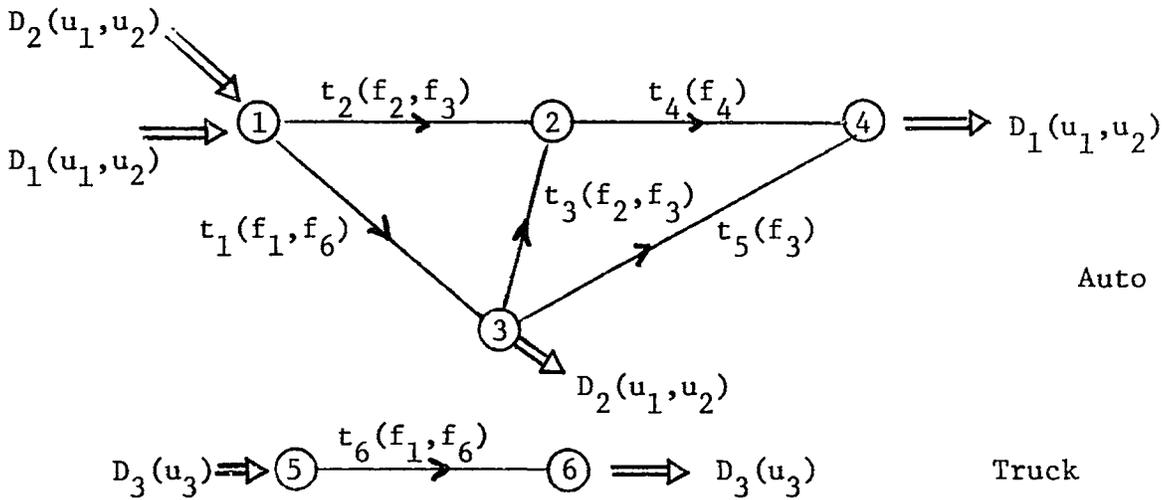


Figure 5.1 Modified Network Configuration for Example 5.2

For the following linear demand functions

$$\begin{cases} D_1(u_1, u_2) = \alpha_1 - \theta_{11}u_1 + \theta_{12}u_2 \\ D_2(u_1, u_2) = d_2 + \theta_{21}u_1 - \theta_{22}u_2 \\ D_3(u_3) = d_3 - \theta_3u_3 \end{cases} \quad \theta > 0$$

VD is symmetric if, and only if, $\theta_{12} = \theta_{21}$, and $-\nabla D(u)$ is positive definite if, and only if, $\theta_{11}\theta_{22} > \theta_{12}\theta_{21}$. With these assumptions,

the components of the objective function become,

$$\begin{aligned} \int_0^f t(x) dx &= \int_0^{(f_1, f_6)} t_1(x, y) dx + t_6(x, y) dy \\ &+ \int_0^{(f_2, f_3)} t_2(x, y) dx + t_3(x, y) dy \\ &+ \int_0^{f_4} t_4(x) dx + \int_0^{f_5} t_5(x) dx \end{aligned}$$

and

$$\begin{aligned} -uD(u) + \int_0^u D(v) dv &= -\sum_{i=1}^3 u_i D_i(u) + \int_0^{(u_1, u_2)} D_1(v_1, v_2) dv_1 + D_2(v_1, v_2) dv_2 \\ &+ \int_0^{u_3} D_3(v_3) dv_3 \\ &= -\sum_{i=1}^3 u_i D_i(u) + \int_0^{u_1} D_1(v_1, 0) dv_1 \\ &+ \int_0^{u_2} D_2(u_1, v_2) dv_2 + \int_0^{u_3} D_3(v_3) dv_3 \\ &= \frac{1}{2}[\theta_{11} u_1^2 + (\theta_{12} + \theta_{21}) u_1 u_2 + \theta_{22} u_2^2] + \frac{1}{2} \theta_{33} u_3^2. \end{aligned}$$

Then the minimization problem becomes,

$$\begin{aligned} \text{Min } &\int_0^{(f_1, f_6)} t_1(x, y) dx + t_6(x, y) dy + \int_0^{(f_2, f_3)} t_2(x, y) dx + t_3(x, y) dy \\ &+ \int_0^{f_4} t_4(x) dx + \int_0^{f_5} t_5(x) dx + \frac{1}{2}[\theta_{11} u_1^2 + (\theta_{12} + \theta_{21}) u_1 u_2 + \theta_{22} u_2^2] + \frac{1}{2} \theta_{33} u_3^2 \end{aligned}$$

subject to:

$$\left\{ \begin{array}{l} h_1 - D_1(u) = 0 \\ h_2 + h_3 + h_4 - D_2(u) = 0 \\ h_5 - D_3(u) = 0 \\ f_1 = h_1 + h_3 + h_4 \\ f_2 = h_2 \\ f_3 = h_2 + h_4 \\ f_5 = h_3 \\ f_6 = h_5 \\ h \geq 0, \quad u \geq 0 \end{array} \right.$$

where h is the vector of path flows, with single paths h_1 and h_5 for O-D pairs 1-3 and 5-6, respectively, and three paths, h_2 , h_3 , h_4 , for O-D pair 1-4. ■

REMARK 5.1: For any link satisfying $t_a(f) = t_a(f_a)$, the line integral becomes the regular integral. Also, when $D_i(u) = D_i(u_i)$, then the minimization problem for the general case is equivalent to the one given by Beckman et al, without explicit use of the inverse function of $D_i(u_i)$. This alternate form results from the following fact,

$$\int_0^{v=D(u)} D^{-1}(v) dv = uD(u) - \int_0^u D(t) dt + \text{constant} . \quad \blacksquare$$

5.4 A LINEARIZATION TECHNIQUE

As we showed in section (3.3), under some mild assumptions the equilibrium problem can be formulated as a non-linear complementarity problem (NCP), i.e.,

$$(NCP) \begin{cases} xF(x) = 0 \\ F(x) \geq 0 \\ x \geq 0 \end{cases}$$

Usually for transportation applications, the size of this problem is so large that it cannot be solved by using existing non-linear complementarity algorithms, such as [C-K4, C-L3]. For example, for a small problem with 100 O-D pairs, the nonlinear complementarity problem contains on the order of 1000 variables (if we only consider 10 paths per O-D pair), whereas the largest (NCP) that can be solved is on the order of 100 variables (taking a few minutes of CPU time).

One possible way to resolve this difficult and to solve large scale problems is by an *iterative procedure*. The idea of an iterative procedure is that, constructing a "movement scheme" to move from one point to a new point and follow the following steps:

Iterative Procedure

Step 1 - Choose a starting point x^0 , and set $q = 0$.

Step 2 - Apply a "movement scheme" to x^q , to move to a new point x^{q+1} .

Step 3 - Set $q = q+1$, if x^q is a "reasonable" solution to NCP, then stop. Otherwise, go to step 2.

For any iterative procedure, it is essential to answer three types of questions:

i) What is the "movement scheme", the starting solution, and

the characterization of a "reasonable" solution?

- ii) When is the procedure guaranteed to reach a reasonable solution (convergence)?
- iii) How efficient is the movement scheme, and how many iterations does it require?

Usually there is a trade-off between the simplicity of the movement scheme and the number of iterations needed, and, as the movement scheme becomes easier to apply, more or less, we expect to have more iterations. We discuss all these questions in this section briefly and, in the next chapter, in more detail.

As we mentioned previously, to solve the (NCP) associated with the equilibrium problem, we face two types of difficulties--the size of the problem (which is in terms of the number of paths), and the difficulty, in general, in solving the (NCP) (even for small sized problems).

To overcome the first difficulty, the size of the problem, we use an iterative procedure called a *decomposition scheme*. In this procedure, we decompose the set of variables $\{x_i; i \in I\}$ into a collection of the mutually exclusive subsets I_1, \dots, I_n . Then corresponding to each subset I_J , we define a subproblem as follows:

$$(SP_J) \begin{cases} F_i(x)x_i = 0 & \text{for all } i \in I_J \\ F_i(x) \geq 0 & \text{for all } i \in I_J \\ x_i \geq 0 & \text{for all } i \in I_J \end{cases}$$

where all x 's are fixed except those x_i with $i \in I_j$. Obviously, each (SP_j) is a restricted version of the original non-linear complementarity problem.

We propose the following iterative procedure to solve the original NCP:

Decomposition Scheme:

Step 1 - Choose a starting point x^0 and set $q = 0$.

Step 2 - For all $J = 1, \dots, M$, solve each (SP_J) to determine values for x_J by fixing $x_i = x_i^q$ for all $i \in I - I_J$. Let x^{q+1} denote the new point that is generated.

Step 3 - Set $q = q+1$. If x^q is a "reasonable" solution to (NCP), then stop. Otherwise, go to step 2.

The efficiency of this procedure is heavily dependent upon how the set I is decomposed. Naturally, it is better to collect together those variables that are most related to each other, so that the corresponding subproblem has the characteristics of the original problem. For example, for transportation applications when $D_i(u) = D_i(u_i)$, if we decompose the problem by O-D pairs, then each subproblem simply becomes a new traffic equilibrium problem in a smaller restricted network with only single O-D pairs. And, in the case of destination choice demand functions, we might decompose the problem in terms of origins. We describe the decomposition criteria in more detail in the next chapter.

If we decompose the set I into smaller subsets, then step 2 of the procedure becomes easier to carry out, while the number of iterations increases rapidly, to the point where the algorithms might never converge.

For the equilibrium problem, it seems, the decomposition in terms of the O-D pairs, provides the smallest subproblems that inherit the essential characteristics of the original problem. But, even for this decomposition, the number of variables (corresponding to the existing paths between the origin and destination) is so large that, no non-linear complementarity algorithm can be used directly to solve the subproblems. Although the number of paths with positive flow is usually small (on the order of 4 or 5) even by knowing those paths it is still not efficient to use any general purpose non-linear complementarity algorithm, because the number of functional evaluations is enormous (at each vertex all the link-volume delay functions have to be evaluated).

This difficulty, which is in the nature of the (NCP), is overcome by introducing another iterative procedure called a *linearization scheme*, which is similar to Newton's method.

We define the linearized problem for (NCP) at \bar{x} as follows:

$$(LCP) \begin{cases} [F(\bar{x}) + (x - \bar{x})\nabla F(\bar{x})]x = 0 \\ f(\bar{x}) + (x - \bar{x})\nabla f(\bar{x}) \geq 0 \\ x \geq 0 \end{cases}$$

Now we propose an iterative procedure to solve (NCP) for x , as follows:

Linearization Scheme

Step 1 - Choose a starting point \bar{x}^0 and set $q = 0$.

Step 2 - Solve (LCP) linearized at \bar{x}^q to find a new point called \bar{x}^{q+1} .

Step 3 - Set $q = q + 1$. If \bar{x}^q is a "reasonable" solution to (NCP), then stop. Otherwise, go to step 2.

Clearly, (LCP) is a linear complementarity problem. As is well-known, when $\nabla F(x)$, the Hessian of $F(x)$, is a positive semi-definite matrix, there are efficient algorithms available [C-C1-2, C-E1, C-L1] to solve the problem. Problems with 100 variables can be solved in an order of a few seconds of CPU time. Therefore, if the iterative procedure gives us a "reasonable" solution in a few iterations, then the linearization scheme would be much faster than any general purpose non-linear complementarity algorithm (which requires on the order of a few minutes of CPU time).

Applying this technique to the traffic equilibrium problem has an important property, that is, the linearized problem (LCP) has the characteristics of the original problem, but is much easier to solve. In other words, the linearized problem is a traffic equilibrium problem with linear functions. But, even for this simplified traffic equilibrium problem, there is no algorithm currently available in the transportation literature to find a solution (in the general case), even though the problem can be solved by linear complementarity algorithms.

In this iterative procedure, because the linearized problem is a traffic equilibrium problem, we can exploit the nature of the problem as being cast in terms of path flows. We do not need to include all paths in the problem at each iteration. Instead, we can include only those paths that have positive flows. This is possible because we can generate shorter travel time paths, if there are any, (using a shortest

path algorithm) at each iteration. Therefore, the (LCP) is much smaller in size than the (NCP) and, consequently, much easier to solve, so that problems with 100 O-D pairs can be solved easily without using any decomposition.

For the traffic equilibrium problem, it is easy to see that $\nabla F(x)$ is positive semi-definite when both $\nabla t(v)$ and $-\nabla D(u)$ are positive semi-definite matrices. To see this, following the notation in section 4.3 we have

$$x = (h, u) \text{ and } v = \Delta h$$

and

$$F(x) = (\Delta^T t(\Delta h) - I'u, \Gamma^T h - D(u)) .$$

Thus,

$$\nabla F(x) = \begin{bmatrix} \Delta^T \cdot \nabla t(\Delta h) \cdot \Delta & , & -\Gamma \\ & & \\ & \Gamma^T & , & -\nabla D(u) \end{bmatrix} .$$

Clearly $\nabla F(x)$ is a positive semi-definite matrix, because for any $\bar{x} = (\bar{h}, \bar{u}) \geq 0$ and $\bar{v} = \Delta \bar{h}$ we have:

$$\begin{aligned} \bar{x}^T \nabla F(x) \bar{x} &= (\bar{h}^T \Delta^T) \nabla t(\Delta h) \cdot (\Delta \bar{h}) - \bar{h}^T \Gamma \bar{u} + \bar{u}^T \Gamma^T \bar{h} - \bar{u}^T \nabla D(u) \bar{u} \\ &= \bar{v}^T \nabla t(v) \bar{v} + \bar{u}^T (-\nabla D(u)) \bar{u} \geq 0 . \end{aligned}$$

EXAMPLE 5.3: To illustrate graphically how the linearization scheme works and how fast it approaches the equilibrium solution, consider a single-link network written as the following equations,

$$\begin{cases} (t(h) - u)h = 0 \\ (h - D(u))u = 0 \\ t(h) - u \geq 0 \\ h - D(u) \geq 0 \\ h \geq 0, u \geq 0 \end{cases}$$

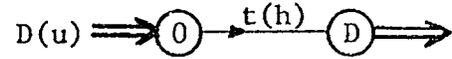


Figure 5.2 represents this problem graphically. $E^* = (h^*, u^*)$ is the equilibrium point. Let us initiate the procedure at point $E^0 = (h^0, u^0)$. Then the linearized problem at E^0 can be shown graphically by lines Lt_1 and LD_1 , where Lt_1 is the supporting line for $t(h)$ at h^0 and LD_1 is the supporting line for $D(u)$ at u^0 . $E^1 = (h^1, u^1)$ represents the solution of this linear complementarity problem. Similarly, $E^2 = (h^2, u^2)$ represents the solution of the linearized problem at E^1 .

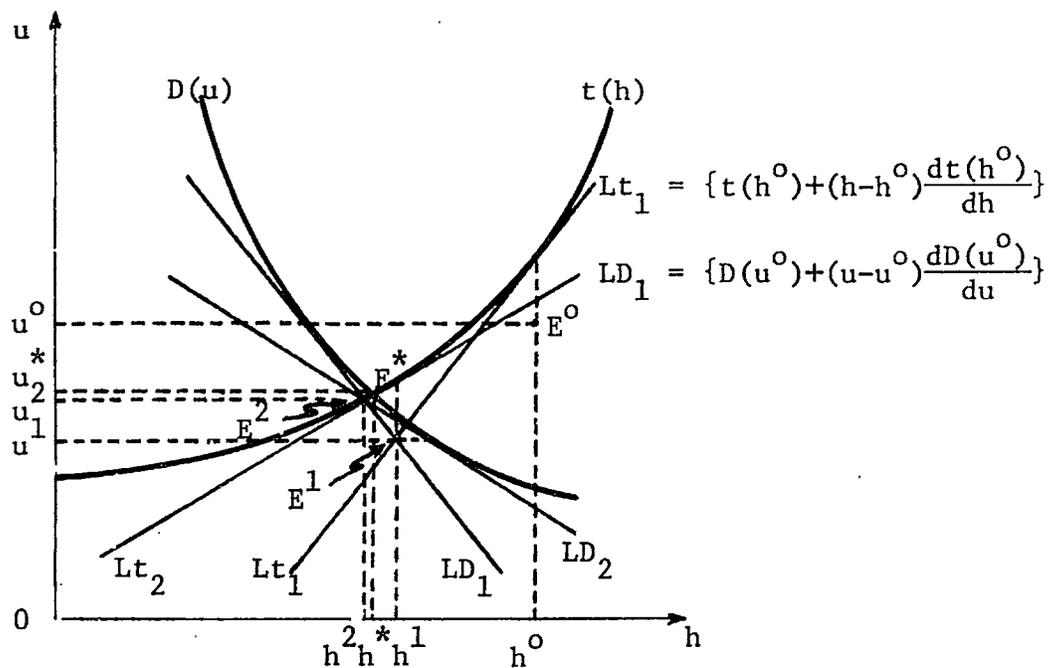


Figure 5.2 Linearization Scheme

In this example, the algorithm converges to the equilibrium point very fast. The computational results in this report show that the algorithm, in general, does not require more than a few iterations. In the appendix, we prove the convergence for this special case of a single link, but we do not have any formal proof for the general case. ■

Still, for any real-life problem, the size of the linearized problem is so big that the procedure cannot be applied directly. However, we can combine the two iterative procedures (a decomposition scheme and a linearization scheme). We propose the following algorithm to solve the original (NCP):

General Scheme

Step 1 - Choose a starting point x^0 and set $q = 0$.

Step 2 - Set $J = 0$.

Step 3 - Set $J = J + 1$. If $J > M$, go to step 6. Otherwise, choose a starting point \bar{x}_J^0 and set $q' = 0$.

Step 4 - Solve (LSP_J) , linearized at \bar{x}_J^q , to find a new point called \bar{x}_J^{q+1} .

Step 5 - Set $q' = q' + 1$, if $\bar{x}_J^{q'}$ is a "reasonable" solution to (LSP_J) then go to step 3. Otherwise set $x_J^{q+1} = \bar{x}_J^{q'}$ and go to step 4.

Step 6 - Set $q = q + 1$. If x^q is a "reasonable" solution to (NCP), then stop. Otherwise, go to step 2.

In this algorithm description, (LSP_J) corresponds to the linearization of (SP_J) at \bar{x} , defined as follows:

$$(\text{LSP}_J) \left\{ \begin{array}{l} (F_i(\bar{x}) + (x_j - \bar{x}_j) \cdot \nabla F_i(\bar{x}))x_i = 0 \quad \text{for all } i \in I_J \\ F_i(\bar{x}) + (x_j - \bar{x}_j) \cdot \nabla F_i(\bar{x}) \geq 0 \quad \text{for all } i \in I_J \\ x_i \geq 0 \quad \text{for all } i \in I_J \end{array} \right.$$

x_j denotes the vector of x_i for all $i \in I_J$, and $\nabla F_i(\bar{x})$ denotes the vector of $\frac{\partial F_i(\bar{x})}{\partial x_k}$ for all $k \in I_J$.

Clearly, when $n = 1$, this scheme is the same as the linearization scheme, and, when all the functions are linear, this scheme is the same as the decomposition scheme.

In the next chapter, when we describe the details of this algorithm, we show how to choose the starting point, and give some practical criteria for assessing when a solution is "reasonable".

Although we do not have any formal proof for the convergence of this algorithm, the computational results are so promising that they encourage the use of this algorithm in practice.

CHAPTER 6

LINEARIZATION ALGORITHM AND COMPUTATIONAL RESULTS

6.1 INTRODUCTION

In this chapter we apply the linearization technique, discussed in section 5.4, to the *general* single mode traffic assignment problem (which includes multi-modal situations) as defined in chapter 3.

In particular, we define an ϵ -approximation equilibrium and describe an algorithmic procedure for computing it. We describe a method for finding a starting solution, discuss possible ways to decompose the problem into subproblems, and give the steps of the algorithm in detail. We also delineate assumptions that are needed for applying the algorithm.

We apply the linearization algorithm to a variety of test problems that have been solved by other researchers and compare our results with theirs. Finally, in this chapter, we present appropriate data structures to solve large scale problems using out-of-core storage facilities.

Throughout this chapter we refer to a *cycle* whenever we solve all subproblems once, and refer to an *iteration* whenever we solve a linearized subproblem.

6.2 LINEARIZATION ALGORITHM

6.2.1 ϵ -Approximation Solution

For any $\epsilon > 0$, a flow pattern h^* is called an " ϵ -approximation" solution or " ϵ -reasonable" solution if it satisfies the conditions:

$$\frac{(\text{Max}_{\substack{p \in P_i \\ h_p^* > 0^i}} \{T_p(h^*)\}) - u_i^*}{\text{Max}_{\substack{p \in P_i \\ h_p^* > 0^i}} \{T_p(h^*)\}} \leq \varepsilon \quad \text{for all } i \in I \quad (\text{A1})$$

$$\left| \frac{\sum_{p \in P_i} h_p^* - D_i(u^*)}{D_i(u^*)} \right| \leq \varepsilon \quad \text{for all } i \in I \quad (\text{A2})$$

where

$$u_i^* = \text{Min}_{p \in P_i} T_p(h^*) \quad \text{for all } i \in I .$$

The first condition (A1) guarantees that the percentage difference between the longest path with positive flow and the shortest path is less than ε for all O-D pairs. The second condition guarantees that the percentage difference between the flowing-flow, $\sum_{p \in P_i} h_p^*$, and the demand, $D_i(u^*)$, is less than ε for all O-D pairs. Sometimes we refer to ε as the accuracy of the solution.

When we are applying the iterative method, it is not a good idea to solve each subproblem to within the ultimate accuracy ε , because the accuracy for any subproblem will be destroyed when another subproblem is solved. Therefore, it is better to start with a less stringent accuracy requirement and to decrease it until the ultimate accuracy is achieved. For example, we can start with $\delta^n \varepsilon$ for some integer $n \geq 0$ and some $\delta > 1$. When the accuracy $\delta^n \varepsilon$ has been achieved, the algorithm continues to impose accuracy requirements $\delta^{n-1} \varepsilon$, $\delta^{n-2} \varepsilon$, . . . and finally, after n steps, accuracy ε . This feature increases the efficiency of the algorithm enormously.

6.2.2 Starting Solution

To find a starting solution to initiate the iterative algorithm, we can use an *All-or-Nothing assignment* [T-D4]. Corresponding to each O-D pair i , this assignment finds the shortest path p_i^0 when all links have zero flow, and assigns all of the generated demand to that path, i.e.,

$$h_{p_i}^0 = D_i(u^0) \quad \text{for all } i \in I$$

where

$$u_{p_i}^0 = T_{p_i}^0(0) \quad \text{for all } i \in I.$$

Notice that in the above assignment, we assign the flow generated by the demand function to a shortest path for each O-D pair sequentially, without considering the effect of the congestion from the flow previously assigned. This might lead us to assign too much flow on some links, with low free travel times. To avoid this, we can update the minimum travel times, u , after each assignment. Also, in the case of an elastic demand function, since the initial u , compared to the u at equilibrium, is small, and, since the demand functions are usually increasing, the all-or-nothing assignment procedure generates too much initial flow, far from the value at the equilibrium. To avoid this, we can assign only some fraction of the generated demands to the shortest paths. We have used this *modified all-or-nothing assignment*, with the choice of 0.5 for the fraction, in our computational results.

6.2.3 Path Generation

As we mentioned previously, we do not need to include all existing paths in the problem, we only include those paths that might have positive flow and we refer to them as the set of *working paths* (denote by P_i^W). Then a solution (h^*, u^*) is called ϵ -approximation in respect to the working paths if conditions (A1) and (A2) are satisfied for sets P_i^W for all $i \in I$. To guarantee that this solution is an ϵ -approximation over all existing paths, that is, the sets P_i for all $i \in I$, we have to satisfy the following condition:

$$\left| \frac{u_i^* - \min_{p \in P_i} T_p(h^*)}{u_i^*} \right| \leq \epsilon \quad \text{for all } i \in I. \quad (\text{A3})$$

To construct the set of working-paths, we start with the paths in the initial solution. We add any path that gives $\min_{p \in P_i} T_p(h^*)$ and that satisfy condition (A3) to the corresponding set of P_i^W . Also, we delete any path with zero flow from the set of P_i^W to maintain the size of the working-paths sets as small as possible.

Although many very efficient algorithms for generating the shortest paths [T-B3, T-D8, T-D9, T-D11, T-G5] are available, because of the enormous number of applications of this algorithm (once for each iteration), it becomes one of the most time consuming components of the linearization algorithm. To reduce the number of applications of the shortest path algorithm, we recall that most of the shortest path algorithms find all the shortest paths from one origin to all destinations

simultaneously. This feature suggests a method for decomposing the problem, that is, decomposition by origin. In other words, we do not collect two O-D pairs with different origins as one subproblem unless all the other O-D pairs with the same origins are included in the subproblem.

Secondly, as we mentioned previously, we do not want to spend too much time in one subproblem to find a very accurate solution, because this accuracy will be destroyed after solving other subproblems. Instead, we prefer to spread our work over all subproblems to achieve, simultaneously, the same, but relaxed, accuracy for all of them. This suggests that we test condition (A3) and generate a shortest path for each O-D pair only once in each cycle, rather than generating a new shortest path after each iteration (linearization). When no linearization (change of flow) takes place in one cycle, then the given accuracy has been achieved and condition (A3) is satisfied.

6.2.4 Decomposition

For the traffic equilibrium problem, various forms of decompositions can be used. The selection from among the various options depends upon the size of the problem and the nature of the demand function. For the reasons discussed in the previous sections and also based upon our intuitions, we have decided to consider the following levels of decomposition:

Level 1 - No decomposition

Level 2 - Decomposition by origin

Level 3 - Decomposition by O-D pair

Level 4 - Decomposition by O-D pair and mode.

Moving from level 1 to level 4, we expect to have more cycles and less iterations within each cycle (because the subproblems become easier to solve). Therefore, it is not clear which level of decomposition is best in terms of efficiency. However, as the size of the problem increases we are forced to use the higher levels of decomposition. On the other hand, as the demand dependency increases, the lower levels of the decomposition will be preferred.

Overall, level 1 will be chosen when we have a completely dependent demand function (i.e., the demand for the O-D pairs depends upon the full vector of accessibility variables). Level 2 will be chosen when we have a destination choice demand function. Level 3 will be chosen when we have only mode choice demand function, otherwise, level 4 will be used. In each case, if the size of the subproblem does not permit us to use that level, we move to the next higher level of decomposition.

Notice that, when there is no mode dependency in the demand function, decomposition by mode might be best as the first level of decomposition.

6.2.5 Algorithm

To see how the linearization algorithm, discussed in chapter 5, is applied to the traffic equilibrium problem, we describe in this section

the steps of the algorithm for the case of decomposition by O-D pairs.

To reduce the number of applications of the shortest path algorithm, we use a two-level decomposition scheme. In the first level, we decompose the problem in terms of origins, and find the shortest path tree for each origin. Then for each origin, in the second level, we decompose the problem in terms of destinations to construct subproblems. In this way, we only solve shortest path problems once each cycle.

To simplify the notation, we consider the single mode case. For the multi-modal case, when the above decomposition is used, the steps of the algorithm remain unchanged except everything is in a vector space corresponding to all modes. For example, each subproblem corresponds to one O-D pair and all possible modes between that O-D pair. The algorithm would be slightly different if we first decompose the problem in terms of modes and then in respect to origins.

Figure 6.1 shows the steps of the algorithm to find an ϵ -approximation solution.

- Step 0 - (initialization) Choose some $\epsilon > 0$, an integer $\bar{n} \geq 0$, and some $\delta > 1$. Apply the Modified All-or-Nothing assignment algorithm to find shortest paths p_i^0 and corresponding, u_i^0 and $h_{p_i}^0$ for all $i \in I$.
 Set $P_i^W = \{p_i^0\}$, $h_{p_i}^0 = h_{p_i}^0$, and $u_i = T_{p_i}^0(h)$ for all $i \in I$.
 Set $n = \bar{n}$, $\epsilon_n = \delta^n \epsilon$ and $IC = 0$.
- Step 1 - Set $IC = IC + 1$ and $IT = 0$.
- Step 2 - Choose an origin I_0 . Apply the shortest path algorithm from origin I_0 to find the shortest paths p_i^S to all $i \in I$ that have origin $(i) = I_0$; set $u_i^S = T_{p_i}^S(h)$.
- Step 3 - Choose an O-D pair $i \in I$ with origin $(i) = I_0$.
- Step 4 - Set $u_i = \min_{p \in P_i^W} T_p(h)$. If $(\max_{\substack{p \in P_i^W \\ h > 0}} T_p(h) - u_i) / \max_{\substack{p \in P_i^W \\ h > 0}} T_p(h) < \epsilon_n$ then go to step 6. Otherwise,
- Step 5 - Solve (LSP_i) linearized at (h, u) , update h , set $IT = IT + 1$, and go to step 4.
- Step 6 - If $(u_i - u_i^S) / u_i < \epsilon_n$ go to step 7. Otherwise set $u_i^S = u_i$, delete any p_i with $h_{p_i} = 0$ from P_i^W . Set $P_i^W = P_i^W \cup \{p_i^S\}$ and $h_{p_i^S} = 0$. Go to step 4.
- Step 7 - If $|(D_i(u) - \sum_{p \in P_i^W} h_p) / D_i(u)| \geq \epsilon_n$, go to step 5. Otherwise,
- Step 8 - If steps 3-7 have been run once for all O-D pairs $i \in I$ with origin $(i) = I_0$, then go to step 9. Otherwise, go to step 3 for a new i .

Figure 6.1 Steps of the Linearization Algorithm

Step 9 - If steps 2-8 have been run once for all origins, then go to step 10. Otherwise, go to step 2 for a new origin, IO.

Step 10 - If $IT = 0$, then go to step 11. Otherwise, go to step 1.

Step 11 - Set $n = n-1$. If $n \leq 0$, then stop (an ϵ -approximation solution has been achieved). Otherwise, set $\epsilon_n = \delta^n \epsilon$, go to step 1.

Figure 6.1 (continued)

In this algorithm, IC denotes the cycle number and IT denotes the iteration number, or the linearization number, within the cycle. Steps 4-7 guarantee that h is an ϵ_n -approximation solution within each subproblem. In other words, step 4 guarantees that the difference between the longest path and the shortest path among working-paths (P_i^W) is less than ϵ_n , condition (A1). Step 6 guarantees that the difference between the shortest path among working-paths (P_i^W) and all paths (P_i) is less than ϵ_n , condition (A3). And finally, step 7 guarantees that the difference between the actual generated demand, $D_i(u)$, and the current carried flow, $\sum_{p \in P_i^W} h_p$, is less than ϵ_n , condition (A2).

Although, within each cycle each subproblem has been solved to within accuracy ϵ_n , at the end of the cycle the solution might not be an ϵ_n -approximation solution for all subproblems simultaneously, because any flow change (linearization) for any O-D pair destroys the accuracy for the other subproblems. However, $IT = 0$ at step 10 guarantees that

the current solution h is an ϵ_n -approximation for the problem, because no flow change has taken place and, furthermore, no new path is needed to be added to the sets of working-paths. Thus conditions (A1), (A2), and (A3) apply to within accuracy ϵ_n . Finally, at step 11, when $n = 0$ the required accuracy ϵ has been achieved.

To solve each subproblem, we will use the solution from previous steps as the starting solution. It is more reasonable, in the overall procedure, that we always use the most recently generated information. To do this, we update all of the data (including path flows, volume delays, minimum travel times, and so forth) whenever any change in the flow occurs. This strategy is applied to the all-or-nothing assignment and to the decomposition and linearization schemes.

For this algorithm, the corresponding subproblem for O-D pair i with the set of working-paths P_i^W can be written as:

$$(SP_i) \left\{ \begin{array}{l} (T_p(h) - u_i) \cdot h_p = 0 \quad \text{for all } p \in P_i^W \\ T_p(h) - u_i \geq 0 \quad \text{for all } p \in P_i^W \\ (\sum_{p \in P_i^W} h_p - D_i(u)) \cdot u_i = 0 \\ \sum_{p \in P_i^W} h_p - D_i(u) \geq 0 \\ h_p \geq 0, u_i \geq 0 \quad \text{for all } p \in P_i^W \end{array} \right.$$

where,

$$T_p(h) = \sum_{a \in A} \delta_{ap} \cdot t_a(f) \quad \text{for all } p \in P_i^W$$

and

$$f_a = F_a + \sum_{p \in P_i^W} \delta_{ap} \cdot h_p \quad \text{for all } a \in A$$

where F_a is the sum of the flows by other O-D pairs on link a . The linearization of (SP_i) at (\bar{h}_p, \bar{u}_i) for $p \in P_i^W$ is:

$$(LSP_i) \left\{ \begin{array}{l} (T_p(\bar{h}) + \sum_{p' \in P_i^W} (h_{p'} - \bar{h}_{p'}) \frac{\partial T_p(\bar{h})}{\partial h_{p'}} - u_i) \cdot h_p = 0 \quad \text{for all } p \in P_i^W \\ T_p(\bar{h}) + \sum_{p' \in P_i^W} (h_{p'} - \bar{h}_{p'}) \frac{\partial T_p(\bar{h})}{\partial h_{p'}} - u_i \geq 0 \quad \text{for all } p \in P_i^W \\ (\sum_{p \in P_i^W} h_p - D_i(\bar{u}) - (u_i - \bar{u}_i) \frac{\partial D_i(\bar{u})}{\partial u_i}) \cdot u_i = 0 \\ \sum_{p \in P_i^W} h_p - D_i(\bar{u}) - (u_i - \bar{u}_i) \frac{\partial D_i(\bar{u})}{\partial u_i} \geq 0 \\ h_p \geq 0, u_i \geq 0 \quad \text{for all } p \in P_i^W \end{array} \right.$$

where

$$\frac{\partial T_p(\bar{h})}{\partial h_{p'}} = \sum_{a \in A} \sum_{a' \in A} \delta_{ap} \cdot \delta_{a'p'} \frac{\partial t_a(\bar{f})}{\partial f_{a'}} \quad \text{for all } p, p' \in P_i^W.$$

Although, computation of the coefficient matrix for each (LSP_i) at

each iteration looks difficult and time-consuming, there are efficient ways to perform these computations (see appendix B).

6.2.6 Assumptions

From the computational point of view, the linearization algorithm only requires some mild assumptions. More restrictive assumptions might be needed, however, to guarantee convergence of the algorithm (see appendix A). These mild assumptions are:

- i) The vector functions $t(f)$ and $D(u)$ are continuous and differentiable.
- ii) Both $t(f)$ and $-D(u)$ are monotone functions, i.e., $\nabla t(f)$ and $-\nabla D(u)$ are positive semi-definite matrices.

It is easy to show that, for any form of decomposition discussed in section 6.4, the coefficient matrix associated with any (LSP_i) is positive semi-definite, and this is a sufficient condition to solve (LSP_i) by linear complementarity algorithms [C-K3].[†]

For transportation applications, these are very mild assumptions that are valid for most of the demand and volume delay models presented earlier in chapter 2.

6.3 COMPUTATIONAL RESULTS

In this section, we present computational results for some small problems with different demand models and for some larger examples to see how the linearization algorithm behaves both in terms of the convergence and efficiency, compared to the other algorithms.

†

It is easy to verify that (LSP_i) satisfies the conditions of the existence theorem 4.4, therefore it always has a solution.

Example 6.1 is a hypothetical problem with a mode-choice demand function for a multi-modal assignment problem. Example 6.2 is a small problem presented in [T-L2] with a destination-choice demand function. Example 6.3 is another test problem presented in [T-S4]. Examples 6.4 and 6.5 are larger problems presented in [T-L2] and [T-F6, T-N3] with fixed and elastic demand functions.

We use Lemke's Algorithm [C-L1], which is an efficient algorithm and can solve the problems with a few hundred variables in a couple of seconds, to solve the linear complementarity problem. To find the shortest path trees, we use the algorithm presented by Golden [T-G5] which is based upon Bellman's method [T-E3]. This is a rather fast algorithm, faster than Dijkstra's [T-D11], that can solve problems with 1000 nodes and 5000 links in less than one second. Recently, other efficient algorithms [T-D9] have been developed to find shortest path trees, using carefully conceived link structures to represent the tree.

All of the programs have been run on an IBM 370/168 using the Fortran G compiler. Reported CPU times do not include I/O times.

EXAMPLE 6.1: The network for the example has 7 nodes, 12 arcs, 2 O-D pairs, and 2 modes (auto and bus) and has the following configuration.

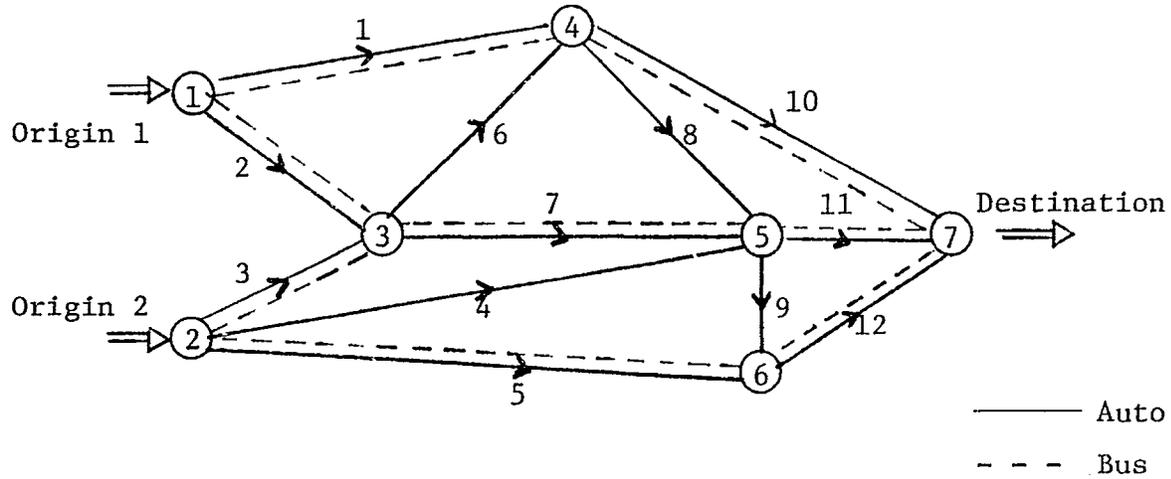


Figure 6.2 Network Configuration for Example 6.1

The dark lines denote auto routes and the dash lines denote bus routes. (1-7) is the first O-D pair and (2-7) is the second O-D pair. We use m to designate the mode, letting $m = 1$ denote the auto mode and $m = 2$ denote the bus mode.

We use the following type of volume delay function for each link:

$$t_a^m(f_a^1, f_a^2) = \bar{t}_a^m \left(1 + .15 \left(\frac{f_a^1 + .2f_a^2}{C_a} \right)^4 \right)$$

where:

\bar{t}_a^m = fixed travel time for link a and mode m (in minutes).

C_a = capacity of link a .

f_a^m = total flow on link a by mode m .

Notice that f_a^2 is the number of passengers who travel by bus. Thus the number of buses will vary with the number of passengers traveling by

the bus mode[†]. We have assumed that each bus carries 20 passengers and each bus is equivalent to 4 autos from the point of view of congestion (or each bus passenger is equivalent to $0.2 = 4/20$ auto passengers).

The parameters for volume delay functions are:

a	1	2	3	4	5	6	7	8	9	10	11	12
\bar{t}_a^{-1}	5	3	4	6	9	3	4	3	2	7	5	4
\bar{t}_a^{-2}	8	6	6	-	12	-	6	-	-	10	7	6
C_a	500	500	500	500	1000	500	1000	500	500	1000	1000	1000

(Note: Dashes designate that there is no link for the bus mode)

Also, we have used a Cobb-Douglas product form of demand model as follows:

$$\begin{cases} D_i^1(u_i^1, u_i^2) = A_i^1 \cdot (u_i^1)^{-\alpha_i^1} \cdot (u_i^2)^{\beta_i^1} & \alpha_i^m \geq 0 \\ D_i^2(u_i^1, u_i^2) = A_i^2 \cdot (u_i^1)^{\alpha_i^2} \cdot (u_i^2)^{-\beta_i^2} & \beta_i^m \geq 0 \end{cases}$$

where A_i^m is constant, α_i^1 and β_i^2 are direct elasticities and α_i^2 and β_i^1 are cross elasticities. The parameter values are:

†

In the example the flow (schedule) of bus routes is not fixed as it might be in practice.

	A_i^m	α_i^m	β_i^m
i=1 and m=1	600,000	3.	0.7.
i=1 and m=2	200,000	0.5	2.
i=2 and m=1	750,000	3.5	0.8
i=2 and m=2	200,000	0.6	1.8

There are 8 auto and 2 bus routes available for each O-D pair, and 20 routes overall in the network. We are interested in finding the amount of flow along each route. There are four access times that we want to determine, corresponding to 2 O-D pairs and 2 modes (u_i^m , $m=1,2$ and $i=1,2$). Therefore, this problem is a nonlinear complementarity problem with 24 variables.

We decomposed the problem in terms of O-D pairs, started with an arbitrary initial solution, and used the linearization algorithm to find a solution with accuracy $\epsilon = 0.1$, with the starting parameters $\delta = 10$ and $\bar{n} = 2$.

Table 6.1 shows the total number of iterations, IT, at each cycle and also for each O-D pair separately. The program sloped after 14 cycles (with IT = 0). At cycles 2, 8, and 14 the solutions are, respectively, 10%, 1%, and 0.1% accurate.

Computational results show that the total demand and shortest path times u_i^m remain almost unchanged after 6 cycles, but that the distribution of the flow between the various paths varies for several additional cycles. We obtained the following solution after 14 cycles.

Cycle No.	1 2		3 4 5 6 7 8						9 10 11 12 13 14					
	No. of Iterations for first O-D	3	0	1	1	1	1	1	0	1	1	1	1	1
No. of Iterations for second O-D	2	0	1	1	1	1	0	0	1	1	1	2	0	0
Total Number of Iterations (IT)	5	0	2	2	2	2	1	0	2	2	2	3	1	0
n	2	2	1	1	1	1	1	1	0	0	0	0	0	0
Accuracy ϵ_n	10%								1%		0.1%			
Solution Time up to the cycle (sec.)	0.3								0.6		1.1			

Table 6.1 Computational Results for Example 6.1

Mode	O-D Pair	Total Demand	Shortest Path (min)	Longest Path(min) with positive flow
Auto	1	1278	16.4772	16.4795
Auto	2	688	14.6850	14.7010
Bus	1	1294	25.0442	25.0442
Bus	2	4424	20.3464	20.3464

Table 6.2 Equilibrium Solution for Example 6.1

The next table specifies the distribution of flow between available paths, but only for those paths with positive flow.

Mode	O-D Pair	Path (nodes)	Flow *
Auto	1	1-4-7	490
		1-3-5-7	583
		1-3-4-7	153
		1-3-5-6-7	52
	2	2-3-5-7	8
		2-5-7	618
2-6-7		62	
Bus	1	1-4-7	1294
	2	2-6-7	4424

*After round-off

Table 6.3 Path Flows for Example 6.1

Figure 6.3 shows the convergence behavior of the algorithm for the flows between the first O-D pair by the auto mode.

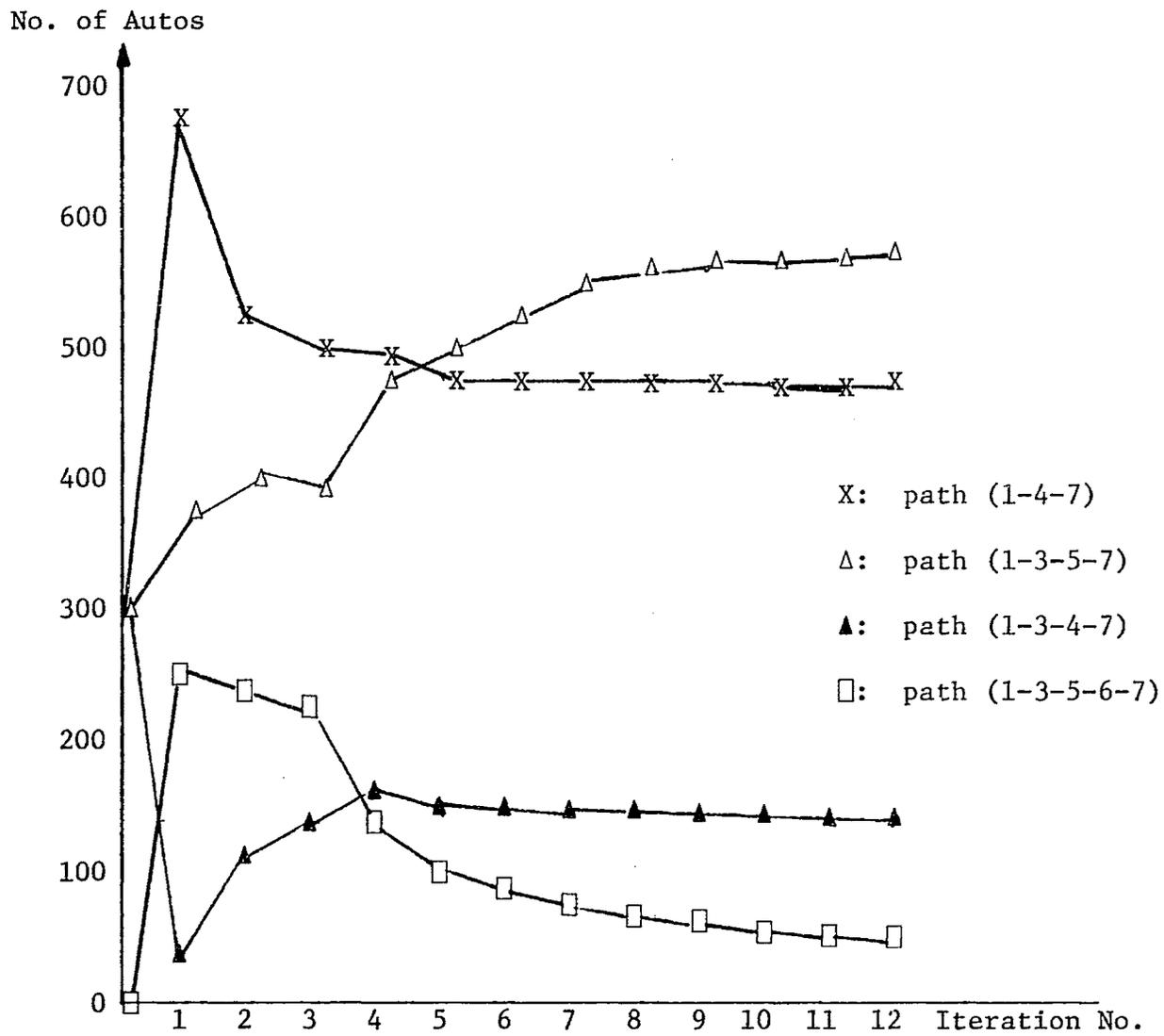


Figure 6.3 Auto Distribution Among the Paths for First O-D Pair

EXAMPLE 6.2: The network for this example has 4 nodes, 10 links and 12 O-D pairs (every pair of nodes corresponds to an origin-destination pair). Figure 6.4 shows the network configuration for this example.

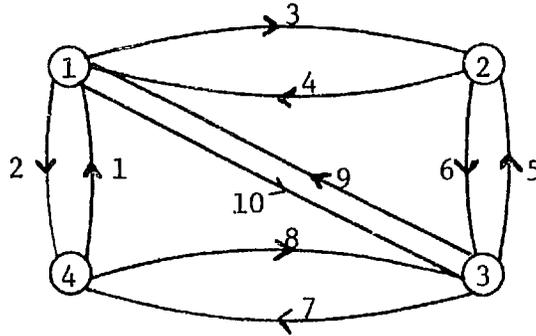


Figure 6.4 Network Configuration for Example 6.2

The volume delay functions are defined as follows:

$$t_a(f_a) = 1.5 + 0.0001 * (f_a^4) \quad \text{for } a = 1, 2, 7, 8, 9, 10$$

$$t_a(f_a) = 3. + 0.0001 * (f_a^4) \quad \text{for } a = 3, 4, 5, 6 .$$

For the first run we used a fixed demand equal to 20 units for each O-D pair.

We applied the linearization algorithm to this problem with $\epsilon = 0.8$, $\delta = 5$, and $\bar{n} = 2$, with the choice of decomposition by O-D pair. The algorithm terminated after 17 linearizations and 8 cycles, and required 0.54 seconds of CPU time. The non-linear complementarity problem associated with this small example has 50 variables (38 path-flow variables and 12 accessibility variables) and would probably require on the order of one minute of CPU time to solve [C-T3].

Table 6.4 shows the link flow at each cycle (at cycles 5, 7, and 8, flow remains unchanged). The final solution is not more than 0.002% away from the exact solution on each link (the last column in table 6.4), which shows how accurate the solution is. Leblanc [T-L2] applied the Frank-Wolfe minimization technique to solve this problem. The solution after approximately 35 iterations is not as accurate as the solution we found by the linearization algorithm.

Cycle Link No. No.	Initial	1	2	3	4	6	Exact Solution
1	20.	26.3739	28.6942	29.7542	29.7542	30.0000	30.
2	20.	28.5260	28.0820	29.3494	29.3494	29.9994	30.
3	20.	32.3401	31.2497	30.2407	29.7542	30.0000	30.
4	20.	31.4265	31.9045	30.6398	29.3494	29.9994	30.
5	40.	27.6602	28.7504	29.7595	30.2460	30.0001	30.
6	40.	28.5737	28.0955	29.3602	30.6507	30.0005	30.
7	40.	31.4741	31.9180	30.6507	30.6507	30.0006	30.
8	40.	33.6252	31.3059	30.2460	30.2460	30.0000	30.
9	20.	30.0188	22.8070	20.4864	19.9999	19.9999	20.
10	20.	26.9532	24.0742	21.2904	20.0000	20.0000	20.

Table 6.4 Link Flow for Example 6.2

For the second run, we used a destination choice model with the following logit functional form:

$$D_{ij}(u) = 30 \frac{r_{ij} e^{-u_{ij}}}{\sum_{k \neq i} r_{ik} e^{-u_{ik}}} \quad \text{for } i = 1, \dots, 4, j = 1, \dots, 4, \text{ and } i \neq j$$

where $D_{ij}(u)$ is the demand between origin i and destination j , 30 is the demand originating at each origin, and $r_{ij} = e^{\theta_{ij}}$, where θ_{ij} is given in table 6.5. The choice of θ_{ij} is such that, at the equilibrium, the flow from each origin to each destination is 10 units.

$i \backslash j$	1	2	3	4
1	-	8.0625	2.5	6.5625
2	8.0625	-	8.0625	14.6250
3	2.5	8.0625	-	6.5625
4	6.5625	14.6250	6.5625	-

Table 6.5

Parameters θ_{ij} of Demand Function for Destination Choice Model

We applied the linearization algorithm with $\epsilon = 0.16$, $\delta = 5$ and $\bar{n} = 3$, with the choice of decomposition by origin. The algorithm terminated after 84 linearizations and 23 cycles, and required 0.86 seconds of CPU time. Table 6.6 shows the number of linearizations and cycles that are needed to achieve different accuracies, and also shows the total link travel time, $\sum_{a \in A} f_a \cdot t_a(f_a)$ with initial value equal to 191.87. Table 6.7 shows the link flows when different accuracies have been achieved.

Accuracy ϵ_n	No. of Linearizations	No. of Cycles	Total Link Travel Time
20%	50	9	918.95
4%	60	14	927.64
0.8%	70	18	927.45
0.16%	84	23	927.49

Table 6.6 Computational Results for Destination Choice Model

Link No. \ ϵ_n	Initial	20%	4%	0.8%	0.16%	Exact Solution
1	10.	15.0112	15.0047	15.0017	15.006	15.
2	10.	15.0097	15.0005	15.0009	15.0004	15.
3	10.	14.9749	15.0340	15.0102	15.0017	15.
4	10.	15.0197	15.0026	15.0003	14.9996	15.
5	5.	14.8720	14.9665	14.9929	14.9986	15.
6	5.	14.9801	14.9971	14.9994	15.0002	15.
7	5.	14.8353	15.0061	14.9932	14.9993	15.
8	5.	14.9887	14.9951	14.9981	14.9993	15.
9	5.	9.6502	9.9134	9.9828	9.9974	10.
10	5.	10.2888	10.0794	10.0228	10.0026	10.

Table 6.7 Link Flows for Destination Choice Model

EXAMPLE 6.3: The network for this example consists of 9 nodes, 36 links, and 12 O-D pairs. The network configuration is shown in figure 6.5.

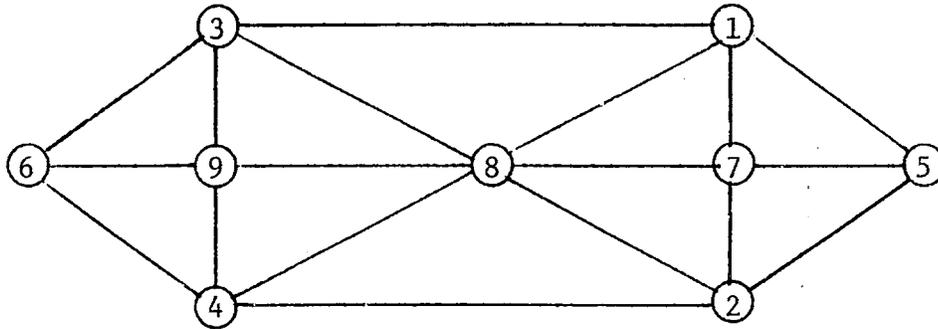


Figure 6.5 Network Configuration for Example 6.3

The volume delay functions are given as:

$$t_a(f_a) = \alpha_a + 0.002 \cdot \beta_a \cdot f_a$$

where α_a and β_a are defined in table 6.9. There is, for each O-D pair $i-j$ for $i = 1, \dots, 4$, $j = 1, \dots, 4$, and $i \neq j$, a fixed demand with values specified in table 6.8.

Origin \ Destination	1	2	3	4
1	-	2000	2000	1000
2	200	-	1000	2000
3	200	100	-	1000
4	100	200	100	-

Table 6.8 Trip Table for Example 6.3

Link No.	Link	α	β	Link Flow by Linearization	Link Flow by Steenbrink
1	1-5	.35	.35	566.8	562.
2	1-3	.0	1.0	1684.6	1694.
3	1-7	.15	.15	1598.1	1582.
4	1-8	.55	.55	1150.5	1162.
5	2-5	.40	.40	36.2	39.
6	2-4	1.0	1.0	1435.9	1430.
7	2-8	.60	.60	1017.5	1011.
8	2-7	.25	.25	710.4	720.
9	3-1	.0	1.0	276.9	236.
10	3-6	.35	.35	268.6	275.
11	3-8	.55	.55	100.0	100.
12	3-9	.15	.15	731.4	725.
13	4-2	1.0	1.0	189.2	199.
14	4-6	.40	.40	0.0	0.
15	4-8	.60	.60	33.9	65.
16	4-9	.25	.25	176.9	136.
17	5-1	.35	.35	36.2	39.
18	5-2	.40	.40	566.8	562.
19	5-7	.30	.30	0.0	0.
20	6-3	.35	.35	0.0	0.
21	6-4	.40	.40	268.6	275
22	6-9	.30	.30	0.0	0.

Table 6.9 Link Flow for Example 6.3

Link No.	Link	α	β	Link Flow by Linearization	Link Flow by Steenbrink
23	7-5	.30	.30	0.0	0.
24	7-1	.15	.15	163.8	161.
25	7-2	.25	.25	1433.2	1438.
26	7-8	.50	.50	711.5	703.
27	8-1	.55	.55	23.1	64.
28	8-2	.60	.60	110.8	101.
29	8-3	.55	.55	912.4	918.
30	8-4	.60	.60	1209.2	1206.
31	8-7	.50	.50	0.0	0.
32	8-9	.50	.50	757.8	752.
33	9-3	.15	.15	579.9	524.
34	9-4	.25	.25	1086.2	1089.
35	9-6	.30	.30	0.0	0.
36	9-8	.50	.50	0.0	0.

Table 6.9 (continued) Link Flow for Example 6.3

For a decomposition by O-D pair, and a choice of $\epsilon = 1.0$, $\delta = 5$, and $\bar{n} = 2$, the linearization algorithm solves this problem in 0.2 seconds of CPU time, after 10 cycles and 45 linearizations. Table 6.9 shows the link-flow volumes founded by this algorithm and also the solution found by Steenbrink [T-S4] using a quadratic programming technique. Notice that, since the volume delay functions for this problem are linear, the equivalent minimization problem is a quadratic programming problem. For different levels of accuracies, table 6.10 shows the value of

$$\sum_{a=1}^{36} \int_0^{f_a} t_a(x) dx ,$$

which is equivalent to the objective value function for the minimization problem. Comparing these values to 16970, the objective value found by Steenbrink, shows how accurate the linearization algorithm is, even though the goal of the algorithm is not minimizing the objective value. Even the solution with 5% accuracy is as good in objective value as the solution found by Steenbrink.

Accuracy ϵ_n	$\sum_a f_a \cdot t_a(f_a)$	$\sum_a \int_0^{f_a} t_a(x) dx$
Initial	102,031.44	53,246.00
25%	27,369.15	17,198.93
5%	27,003.61	16,971.05
1%	26,965.25	16,958.24

Table 6.10 Total Travel Time for Example 6.3

EXAMPLE 6.4: This test problem has been presented by Leblanc in [T-L2]. Figure 6.6 shows the network configuration. The network consists of 24 nodes, 76 links, and 552 O-D pairs (all possible choices of i - j for $i = 1, \dots, 24$, $j = 1, \dots, 24$, and $i \neq j$). There is a fixed demand between each O-D pair, and the volume delay functions are defined as:

$$t_a(f_a) = \alpha_a + \beta_a \cdot f_a^4.$$

All the data are specified in [T-L2].

We applied the linearization algorithm to this test problem with the choice of $\epsilon = 1$, $\delta = 5$, $\bar{n} = 2$, and used decomposition by O-D pair. The algorithm terminated after 18 cycles and 564 linearizations, and required 3.32 seconds of CPU time to find 1%-approximation solutions.

Table 6.11 contains the number of linearizations, total link travel time, and the maximum percentage change in the link flow after each cycle. Also, it includes the maximum percentage change after each iteration for the Frank-Wolfe algorithm, used by Leblanc. These results show how fast the linearization algorithm converges and it exhibits less of a tailing phenomenon than the Frank-Wolfe algorithm. In terms of computational time, the linearization algorithm requires 2.15 seconds on an IBM 370/168 to achieve 5% accuracy, while the Frank-Wolfe algorithm requires 10 seconds on the CDC 74 (notice that the IBM 370 is much faster than the CDC 74).

Cycle No.	No. of Linearizations	Total Link Travel Time	Max % Change in Link Flow	Max % Change in Link Flow by Frank-Wolfe
0	-	922.96	-	-
1	185	140.00	196.9	68.7
2	74	108.91	38.5	46.6
3	30	102.22	14.7	39.4
4	9	100.70	14.5	50.0
5	73	97.09	17.8	32.1
6	48	96.14	14.9	100.0
7	25	95.92	7.2	41.1
8	18	95.82	5.3	21.6
9	5	96.01	6.6	35.4
10	37	95.96	3.1	16.3
11	18	95.94	.8	25.0
12	14	95.94	.8	16.0
13	11	95.91	.8	13.9
14	8	95.92	.4	9.6
15	6	95.91	.5	11.4
16	2	95.92	.5	7.7
17	1	95.92	.5	11.2
18	0	95.92	.0	7.9

Table 6.11 Computational Results for Example 6.4

EXAMPLE 6.5: This example is a rather moderate-sized test problem from real-life data, which has been used by other researchers [T-F6, T-N3, T-N4]. The computational results for this example give some idea of how good the linearization algorithm is, compared to the other algorithms, both in terms of convergence and efficiency. The example is based upon the data for the street network of the city of Hull, Canada.

The network has 155 nodes, 376 one-way links, 27 zones, and 702 O-D pairs (all possible pairs of i - j for $i = 1, \dots, 27$, $j = 1, \dots, 27$, and $i \neq j$). There is only one mode of transportation (auto). The volume delay functions are given by the travel time function suggested in the BPR traffic assignment manual [T-B12], which has the form:

$$t_a(f_a) = t_a^0 [1 + .15(f_a/c_a)^4] \quad \text{for } a = 1, \dots, 376$$

with parameters defined as in section 2.2. Finally, there is a fixed demand between each O-D pair. The data for this problem is a slight modification of that used in [T-F6]. (Notice that there are some minor differences in the data. In particular, we scaled the demand by a factor of 10, and this is the reason for some differences between our results and the results reported in [T-F6, T-N3]).

For the choice of $\varepsilon = 1$, $\delta = 5$, $\bar{n} = 2$ and a decomposition by O-D pair, we applied the linearization algorithm to this problem. The algorithm terminated after 20 cycles and 590 linearizations, and required 16.37 seconds of CPU time to find a solution with 1% accuracy. The maximum number of paths between each O-D pair with positive flow is 4 and the maximum number of links in the paths with positive flow is 44.

Table 6.12 shows the number of cycles and linearizations to reach different levels of accuracy. Also, it shows the computational times and the total link travel time, $\sum_a f_a \cdot t_a(f_a)$.

Accuracy ϵ	No. of Cycles	No. of Linearizations	CPU Time (sec)	Total Link Travel Time (min)
Initial				590,336.
25%	4	179	3.81	257,570.
5%	12	405	10.49	236,631.
1%	20	590	16.37	235,776.

Table 6.12
Computational Results for Example 6.5 with Fixed Demand

Nguyen in [T-N3] used the Convex-Simplex Method to solve the equivalent minimization problem for the city of Hull. This algorithm required 42.16 seconds of CPU time on a CDC CYBER 74 to find a solution with an accuracy almost equivalent to 5%, as we defined it in section 6.2.1 (Nguyen has used different criteria for the accuracy).

In [T-F6], Florian and Nguyen reported other computational times for both fixed demand and elastic demand for variations of this problem, for different numbers of O-D pairs, up to 421. For the case of 421 O-D pairs, the CPU time is 43.42 seconds on the CDC CYBER 74 to find a solution with 5% accuracy, as we defined it in section 6.2.1. The linearization algorithm requires only 10.49 seconds on an IBM 370/168 for a problem with 702 O-D pairs.

In the second run, we used an elastic demand function with a linear functional form as follows:

$$D_i(u_i) = b_i - a_i u_i \quad \text{for } i = 1, \dots, 702$$

where a_i and b_i have been selected randomly, in a fashion similar to that reported in [T-F6] by Florian and Nguyen. Table 6.13 shows the results.

Accuracy ϵ	No. of Cycles	No. of Linearizations	CPU Time (sec)	Total Link Travel Time (min)
Initial				197,681.
25%	6	468	8.03	234,532.
5%	14	1542	11.21	234,344.
1%	20	2548	18.46	234,004.

Table 6.13 Computational Results for Example 6.5 with Elastic Demand

Comparing the results in Tables 6.12 and 6.13 shows that obtaining an equilibrium assignment with elastic demand only requires 15 per cent more computational time than the fixed demand case. Although the number of linearizations increases four fold, the computational time does not grow nearly as much. This is because the computational time for the linearization algorithm depends more on the number of cycles than the number of linearizations. Therefore, the algorithm does not depend too much on the type of the demand function.

The algorithm presented by Florian and Nguyen, which is based upon

Benders Decomposition Method, required 54.13 seconds on the CDC CYBER 74 to achieve 5% accuracy, even with only 421 O-D pairs. This is almost 25 per cent more than the time for the fixed demand problem as compared with 15 per cent for the linearization algorithm. For approximately 702 O-D pairs, the Florian-Nguyen algorithm required 80 seconds on the CDC CYBER 74 to achieve 5% accuracy. In contrast, the linearization algorithm required only 11.21 seconds on an IBM 370/168. Of course, the IBM 370/168 is faster than the CDC CYBER 74, but not more than four times faster. Also notice that they have used the optimizing FTN compiler, while we have used the FORTRAN G compiler.

Because of different operating environments, it is difficult to judge between these algorithms. At the very least, these results show that the linearization algorithm is as fast as, if not faster than, the specialized algorithms presented by Florian and Nguyen, which are among the fastest existing algorithms for solving the traffic equilibrium problem. However, the linearization algorithm has its own important advantage, which is the generality of the algorithm compared to any algorithm based upon minimization technique. A disadvantage to the linearization algorithm is that, at present, theoretical studies of its convergence behavior are limited.

6.4 STORAGE REQUIREMENT AND DATA STRUCTURE

The storage requirement for the linearization algorithm consists essentially of three parts, namely the computer program, the problem information, and path flow information. The computer program itself

need 10K words of computer memory. This includes the main program and all of the subroutines such as LCP (the Linear-Complementarity Program) and BELL (the shortest path algorithm).

To store all of the data specifying the problem requires, at most, $8|A| + 6|N| + 10|I|$ words of memory, as described in table 6.14, where $|A|$ is the number of links, $|N|$ is the number of nodes, and $|I|$ is the number of O-D pairs. This includes the network structure, the tree structure for the shortest path algorithm, the link flows and path flows, parameters of the volume delay and demand functions (such as the data for the city of Hull with elastic demand), and, finally, it includes vectors to store the update values for $t(f)$, $\nabla t(f)$, $D(u)$, and $\nabla D(u)$.

Dimension	Arrays
A	Start-node, End-node, Link flow f , Link travel time $t(f)$, $dt(f)/df$, 2 arrays for parameters of the volume delay function, one dummy array for BELL subroutine.
N	4 arrays (pointer to the first link starting at each node, pointer to the predecessor node, and level from the root) to represent the shortest path tree, shortest travel time from an origin to all nodes, one dummy array.
I	Origin node, destination node, 4 arrays for path flows h (those with positive flow), minimum travel time u , total demand $D(u)$, $dD(u)/du$, number of paths.

Table 6.14 Memory Requirement to Store the Problem Data

This data can easily be kept in memory on the IBM 370 even for networks with 10,000 links. For implementations with limited storage, this storage requirement can be reduced to $6|A| + 6|N| + 8|I|$, in favor of more computational time by reevaluating $t(f)$, $\nabla t(f)$, $D(u)$, and $\nabla D(u)$ whenever they are needed, instead of storing this data. Overall, these requirements do not create any major difficulty to store any large scale traffic assignment problem in core.

The last, and major, requirement for storage is the path information. If we assume that the maximum number of paths with positive flows is M_1 and the maximum number of links in any path is M_2 , then for each O-D pair we might allocate a fixed space equal to $M_1 * M_2$ to store arc-path chains. Therefore, to store all path information we require $M_1 * M_2 * |I|$ words of memory. For the choice of $M_1 = 4$, $M_2 = 50$, and $|I| = 700$, as is the case of example 6.5, the storage requirement would be 140K words, which can be stored in core on an IBM 370. But, most computers will charge for using extra core storage.

To make the linearization algorithm capable of solving larger problems and, also, to reduce the cost of using extra core storage, we have to reduce the storage requirement for the path information. There are two ways to achieve this goal--modifying the data structure for storing path information, and using out-of-core storage (such as disk or tape).

6.4.1 Modified Data Structure

Previously we allocated a fixed space equal to $M_1 * M_2$ for each O-D pair. In practice, though, O-D pairs will not have M_1 paths with

positive flow (for the city of Hull, there are only 947 paths with positive flows which shows that, on the average, there are only 1.35 paths with positive flows joining each O-D pair) and not all of the paths have M_2 links. Therefore, there is a great deal of un-used allocated storage. However, this fixed storage scheme has an advantage, and that is, a fast accessing mechanism to groups of paths with the same O-D pairs or with the same origins (this is important for the decomposition schemes that we use). In fact, in this allocation, the paths are stored in a sequential order in terms of O-D pairs and origins.

Since we are generating the paths, it is not easy to keep this sequential ordering when we use variables M_1^i and M_2^i for each O-D pair i . However, by introducing some pointers we can store the path information with variables M_1^i and M_2^i , and still have a good accessing mechanism to a group of paths. Naturally, the accessing time to any path will be increased above that required by the fixed space scheme. Thus, there is a tradeoff between CPU time and the storage utilization.

We have implemented the linearization algorithm for variable M_1^i and fixed M_2 . Two pointers are enough to locate any path; these are called FIRST and NEXT. For O-D pair i , FIRST (i) indicates the location of the first path joining O-D pair i , and NEXT (p) indicates the location of the next path with the same O-D pair as path p . Next (p) is set to zero when p is the last path joining an O-D pair.

The second row in table 6.15, designated problem P_1 , shows the computational results for implementing this modified data structure scheme for the city of Hull example with elastic demand functions. As

the results show, the modified algorithm requires 60K words to store the path information, compared to 140K for the original case (P_0). This reduces the in-core storage cost by \$2.54, while it increases the CPU cost by \$0.61. Thus, the total savings in cost is \$1.93.

Therefore, this modification makes the algorithm capable of solving larger traffic assignment problems and, at the same time, reduces the total running cost. An even better improvement might be achieved by allocating variable space for M_2^i , the number of links in the path, as well.

6.4.2 Out-of-Core Storage

In theory, we can always use out-of-core storage. But the question is when is it efficient and economical. This depends on the choice of record size, number of times we need to access to the records, and, more important, on the order we need to access the records (sequential or random).

For the case of fixed space allocation, all of the paths with the same O-D pair and same origin are listed in a sequential order. Thus, for the decomposition scheme discussed in section 6.2.4, we require, within each cycle of the algorithm, only sequential accessing to all of the records. This is not the case for the modified data structure. For this reason, it seems that an out-of-core storage facility is more appropriate for the fixed storage scheme than for the modified scheme. Now the question is what is the optimal record size in terms of total computer running cost.

We have examined two different record sizes. For the first run we have chosen the record size equal to $M_1 * M_2$ so that we could fit all of the path information corresponding to each O-D pair in one record.

The computational results for this run, indicated by problem P_2 , is shown in table 6.15.

Table 6.15 shows the computational results of different runs for example 6.5, city of Hull. The table contains a variety of information, to make the comparison more clear; these are: problem number, CPU time, CPU cost, record size, in-core storage cost, accessing cost to out-of-core storage (disk), other costs (which include compiling cost, I/O cost, and so forth), total running cost, number of accesses to out-of-core storage, in-core storage requirements for path information, and total in-core storage requirements. Notice that all the runs give the same solution.

Run P_2 required only 45K words, which is a reasonable storage requirement for any small computer, compared to 174K for P_0 . Therefore, the variable space storage scheme is practical for solving much larger traffic assignment problems. However, the total running cost increases, as we might expect, and by a factor of 3. The first factor contributing to this cost increase is the accessing cost to out-of-core storage. The second factor is the increase in in-core storage cost, even though this run requires less in-core storage, because the program must stay idle during the accessing process. Finally, the last factor is the increase in the CPU cost due to substantial swapping to transfer data from the out-of-core storage to the arrays in the memory. To reduce the total running cost for P_2 , we have to decrease the number of accesses to the out-of-core storage. To accomplish this, we need to increase the record size.

Problem No.	CPU Time (sec)	CPU Cost \$	Record Size (Words)	In-core Storage Cost \$	Accessing Cost to Out-of-Core Storage\$	Other Costs \$	Total Running Cost \$	No. of Accesses to Out-of-Core Storage	In-core Storage for paths (K words)	Total In-core Storage (K words)
P ₀	18.46	4.84	-	5.22	-	11.47	21.53	-	140	175
P ₁	20.85	5.45	-	2.68	-	10.46	18.69	-	60	100
P ₂	53.55	12.10	200	18.00	19.50	15.60	64.0	15620	.2	45
P ₃	93.61	20.62	5200	6.24	1.61	10.14	38.61	1333	5.2	52

Table 6.15

Computational Results for the City of Hull with Elastic Demand for Modified Data Structure and Using Out-of-Core Storage

For the next run, we have chosen the record size equal to $M_1 * M_2 * d$, where d is the maximum number of destinations associated with each origin (for the city of Hull, $d = 26$). In other words, we store all of the paths originating at each origin in one record. The computational results for this run, which is indicated by problem P_3 , is shown in table 6.15.

As the results show, the number of accesses to the out-of-core storage, and therefore the in-core storage cost and accessing cost, has decreased enormously for run P_3 as compared to run P_2 . Also, P_3 requires only 7K words more of in-core storage.

The reason that P_3 requires more CPU cost is that, in this case, we must, at each step, substitute all of the information from the one record into the corresponding array in memory. But there is a great deal of non-usable information in most records, because not all of the O-D pairs have M_1 paths with positive flows. In contrast, for the implementation P_2 , we only transfer the paths with positive flows, which are located at the top of the record, and discard the rest of the record. This is not possible for the implementation P_3 .

Over all, among the four schemes, when there is no in-core storage limitation, the modified data structure scheme (P_1) is the best in terms of total cost, and the original scheme (P_0) is the best in terms of speed. When storage is limited, then scheme P_3 seems to be best in terms of total running cost, at least for the computational testing on this one example.

REFERENCESTraffic Equilibrium

- [T-A1] AASHTIANI, H. Z., "The Multi-Modal Traffic Assignment Problems," DOT, Technical Summary 77-1 (January 1977).
- [T-A2] ACHIM, C., CHAPLEAU, R., CHRIQUI, C. and ROBILLARD, P., "TRANSCOM-Guide de l'Utilisateur," publication No. 23, Centre de Recherche sur les Transports, Université de Montréal (1975).
- [T-A3] ADLER, J. and BEN-AKIVA, M., "A Joint Frequency, Destination, and Mode Model for Shopping Trips," Transportation Research Board 569, pp. 136-150 (1975).
- [T-A4] ASMUTH, R. L., "Traffic Network Equilibrium," Technical Report SOL-78-2, Stanford University, California (1978).
- [T-A5] ATHERTON, T. J. and BEN-AKIVA, M., "Methodology for Short-Range Travel Demand Predictions," Journal of Transport Economics and Policy, Vol. 11, No. 3, pp. 224-261 (1977).
- [T-B1] BECKMAN, M. J., MCGUIRE, C. B. and WINSTEN, C. B., Studies in the Economics of Transportation, Yale University Press, New Haven, Connecticut (1956).
- [T-B2] BEILNER, H. and JACOBS, F., "Probabilistic Aspects of Traffic Assignment," in Newell (editor), Traffic Flow and Transportation, Elsevier, New York, pp. 183-194 (1972).
- [T-B3] BELLMAN, R., "On a Routing Problem," Quart. Appl. Math., Vol. 16, pp. 87-90 (1958).
- [T-B4] BEN-AKIVA, M. E., "Structure of Passenger Travel Demand Models," Ph.D. Dissertation, M.I.T., Department of Civil Engineering (1973).
- [T-B5] BEN-AKIVA, M. and LERMAN, S., "Disaggregate Travel and Mobility Choice Models and Measures of Accessibility," paper presented at the 3rd International Conference on Travel Modeling, Adelaide, Australia (1977).
- [T-B6] BERTSEKAS, D. P., "Algorithms for Optimal Routing of Flow in Networks," Coordinated Science Lab. Working Paper, University of Illinois at Champaign-Urbana (1976).
- [T-B7] BRADLEY, G., "Survey of Deterministic Networks," AIIE TRANSACTIONS, Vol. 7, pp. 222-234 (1975).

- [T-B8] BRADLEY, S. P., "Solution Techniques for the Traffic Assignment Problem," Operations Research Center, University of California ORC 65-35, Berkeley, California (1956).
- [T-B9] BRAND, D. and MANHEIM, M. L. (editors), "Urban Travel Demand Forecasting," Proceedings of a Conference held at Williamsburg, Virginia, Special Report 143, Highway Research Board (1973).
- [T-B10] BRANSTON, D., "Link Capacity Functions: A Review," Transportation Research, Vol. 10, No. 4, pp. 223-236 (1976).
- [T-B11] BRUYNOCHE, M., GIBERT, A. and SAKAROVITCH, M., "Une Méthode d' Affectation du Traffic," in Fourth Symposium of the Theory of Traffic Flow, Karlsruhe (1968).
- [T-B12] BUREAU OF PUBLIC ROADS, "Traffic Assignment Manual," U. S. Department of Commerce, Urban Planning Division, Washington, D.C. (1964).
- [T-C1] CASEY, R. F., "Summary Data for Selected New Urban Transportation Systems," Transportation Systems Center, M.I.T., Cambridge, Massachusetts (1972).
- [T-C2] CHARLES RIVER ASSOCIATES, "A Disaggregated Behavioral Model of Urban Traffic Demand," Cambridge, Massachusetts, Report 156-2 (1972).
- [T-C3] CHICAGO AREA TRANSPORTATION STUDY, Final Report, Vol. 2, pp. 104-110 (1960).
- [T-D1] DAFERMOS, S. C., "An Extended Traffic Assignment Model with Applications to Two-Way Traffic," Transportation Science, Vol. 5, pp. 366-389 (1971).
- [T-D2] DAFERMOS, S. C., "The Traffic Assignment Problem for Multiclass-User Transportation Network," Transportation Science, Vol. 7, pp. 73-87 (1972).
- [T-D3] DAFERMOS, S. C. and SPARROW, F. T., "The Traffic Assignment Problem for a General Network," J. Research NBS Ser. B., 73B, pp. 91-118 (1969).
- [T-D4] DAGANZO, C., "Network Representation, Continuous Approximations and a Solution to the Spatial Aggregation Problem of Traffic Assignment," Working Paper 7702, Stochastic Traffic Assignment Project, Inst. of Trans. Studies, University of California, Berkeley, California (1977).
- [T-D5] DEMBO, R. S. and KLINCEMICZ, J. G., "An Approximate Second-Order Algorithm for Network Flow Problem with Convex Separable Costs," Yale School of Organization and Management, Working Paper Series No. 21 (1978).

- [T-D6] DETROIT AREA TRANSPORTATION STUDY, Vol. 2, pp. 79-107.
- [T-D7] DIAL, R. B., "A Combined Trip Distribution and Modal Split Model," paper presented at the 1974 Annual Meeting of the Highway Research Board (1973).
- [T-D8] DIAL, R. B., "Algorithm 360 Shortest Path Forest with Topological Ordering," Comm. ACM 12, pp. 632-633 (1969).
- [T-D9] DIAL, R. B., GLOVER, F., KARNEY, D. and KLINGMAN, D., "A Computational Analysis of Alternative Algorithms and Labeling Techniques for Finding Shortest Path Trees," Research Report CCS 291, Center for Cybernetics Studies, The University of Texas, Austin, Texas, (1977).
- [T-D10] DIAL, R. B. and BUNYAN, R. E., "Public Transit Planning System," Socio-Economical Plan. Science, Vol. 1, pp. 345-362 (1968).
- [T-D11] DIJKSTRA, E., "A Note on Two Problems in Connection with Graphs," Numerical Mathematics, pp. 269-271 (1959).
- [T-D12] DOMENEICH, T. A. and MCFADDEN, D., Urban Travel Demand, North-Holland, Amsterdam (1975).
- [T-D13] DOUGLAS, C. J., "A Method of Traffic Assignment to an Urban Network," Highway Research Board 224, pp. 64-71 (1959).
- [T-D14] DREW, D. R., Traffic Flow Theory and Control, McGraw-Hill, New York (1968).
- [T-F1] FERLAND, J. A., FLORIAN, M. and ACHIM, C., "On Incremental Methods for Traffic Assignment," Transportation Research, Vol. 9, pp. 237-239 (1975).
- [T-F2] FLEET, C. R. and ROBERTSON, S. R., "Trip Generation in the Transportation Planning Process," Highway Research Record 240, Highway Research Board, Washington, D.C. (1968).
- [T-F3] FLORIAN, M., Traffic Equilibrium Methods, Lecture Notes in Economics and Mathematical Systems 118, Springer-Verlag, Berlin (1974).
- [T-F4] FLORIAN, M., "A Traffic Equilibrium Model of Travel by Car and Public Transit Modes," Transportation Science, Vol. 11, pp. 166-179 (1977).
- [T-F5] FLORIAN, M., CHAPLEAU, R., NGUYEN, S., ACHIM, C., JAMES, L. and LEFEBVRE, J., "EMME: A Planning Method for Multi-Modal Urban Transportation Systems," Publication No. 62, Centre de Recherche sur les Transports, Université de Montréal (1977).

- [T-F6] FLORIAN, M. and NGUYEN, S., "A Method for Computing Network Equilibrium with Elastic Demand," Transportation Science, Vol. 8, No. 4, pp. 321-332 (1974).
- [T-F7] FLORIAN, M. and NGUYEN, S., "Recent Experience with Equilibrium Methods for the Study of a Congested Urban Area," Proceedings of the International Symposium on Traffic Equilibrium Methods, University of Montreal (1974).
- [T-F8] FLORIAN, M. and NGUYEN, S., "An Application and Validation of Equilibrium Trip Assignment Methods," Transportation Science, Vol. 10, No. 4, pp. 376-390 (1976).
- [T-F9] FRANK, M. and WOLFE, P., "An Algorithm of Quadratic Programming," Naval Research Logistics Quarterly, Vol. 3, pp. 95-110 (1956).
- [T-G1] GARTNER, N., LITTLE, J. D. C. and GABBAY, H., "Optimization of Traffic Signal Settings by Mixed-Integer Linear Programming, Part I and II," Transportation Science, Vol. 9, pp. 321-363 (1975).
- [T-G2] GEOFFRION, A., "Customer Aggregation in Distribution Modeling," Working Paper No. 259, Management Science Study Center, University of California, Los Angeles (1976).
- [T-G3] GILBERT, A., "A Method for the Traffic Assignment Problem when Demand is Elastic," LBS-TNT-85, Transport Network Theory Unit, London Business School, London (August 1968).
- [T-G4] GOLDEN, B., "A Minimum Cost Multi-Commodity Network Flow Problem Concerning Imports and Exports," Networks, Vol. 5, pp. 331-356 (1975).
- [T-G5] GOLDEN, B., "Shortest-Path Algorithm: A Comparison," Operations Research, Vol. 24, No. 6, pp. 1164-1168 (1975).
- [T-G6] GORDON, S. and DENEUEVILLE, R., "Design of Air Transportation Networks," Transportation Research, Vol. 7, pp. 207-222 (1973).
- [T-H1] HALL, H. H. and PETERSON, E. L., "Traffic Equilibria Analyzed via Geometric Programming," Discussion Paper No. 130, Center for Math. Studies in Economics and Management Sciences, Northwestern University (1975).
- [T-H2] HEARN, D. W. and KUHN, H., "Network Aggregation in Transportation Planning - Final Report," MATHTECH, Inc., Princeton, New Jersey (1977).
- [T-H3] HIGHWAY CAPACITY MANUAL, Highway Research Board, Special Report 87, Washington, D.C. (1965).

- [T-H4] HOLROYD, E. M. and SCRAGGS, D. A., "Journey Times by Bus and Car in Central London," Traffic Engineering and Control, Vol. 6, No. 3, pp. 169-173 (1974).
- [T-I1] IRWIN, N. A., DODD, N. and VON CUBE, H. G., "Capacity Restraint in Assignment Programs," Highway Research Board 297, pp. 109-127, (1961).
- [T-I2] IRWIN, N. A. and VON CUBE, H. G., "Capacity Restraint in Multi-Travel Mode Assignment Programs," Highway Research Board 347, pp. 258-289 (1962).
- [T-J1] JACOBSON, J., "Case Study Comparison of Alternative Urban Travel Forecasting Methodologies," S. M. Dissertation, Department of Civil Engineering, M.I.T., Cambridge, Massachusetts (1977).
- [T-J2] JEWELL, W. S., "Models for Traffic Assignment," Transportation Research, Vol. 1, pp. 31-46 (1967).
- [T-K1] KADOSCH, M., "Temps d'Attente dans les Transports Urbain en Commun," R.A.I.R.O. Recherche Opérationnelle, Vol. 10, No. 2, pp. 37-54 (1976).
- [T-L1] L'ABBE, B. et SCHERER, C., "Un Modèle Global pour l'Evaluation des Projets d'Extentions des Réseaux de Transport Public en Région Parisienne," Proceedings of the AFCET Conference on Traffic Control and Transportation Systems, North-Holland, pp. 677-688 (1974).
- [T-L2] LEBLANC, L. J., "Mathematical Programming Algorithms for Large Scale Network Equilibrium and Network Design Problems," Ph.D. Dissertation, Department of Industrial Engineering and Management Sciences, Northwestern University (1973).
- [T-L3] LEBLANC, L. J., MORLOK, E. and PIERSKALLA, "An Efficient Approach to Solving the Road Network Equilibrium Traffic Assignment Problem," Transportation Science, Vol. 9, pp. 309-318 (1975).
- [T-L4] LEVENTHAL, T. L., NEMHAUSER, G. L. and TROTTER, L. E., "A Column Generation Algorithm for Optimal Traffic Assignment," Transportation Science, Vol. 7, No. 2, pp. 168-176 (1973).
- [T-M1] MANHEIM, M. L., "Practical Implications of Some Fundamental Properties of Travel Demand Models," Highway Research Board Record 422, pp. 21-38 (1973).
- [T-M2] MANHEIM, M. L., Fundamentals of Transportation Systems Analysis, Department of Civil Engineering, M.I.T., Cambridge, Massachusetts (1974).

- [T-M3] MANHEIM, M. L. and MARTIN, B. V., "A Research Program for Comparison of Traffic Assignment Techniques," Highway Research Record 88 (1965).
- [T-M4] MANHEIM, M. L. and RUITER, E. R., "DODOTRANS I: A Decision Oriented Computer Language for Analysis of Multi-Mode Transportation Systems," Highway Research Record 314, pp. 135-163 (1970).
- [T-M5] MARTIN, B. V., MEMMOTT, F. W. and BONE, A. J., "Principles and Techniques of Predicting Future Demand for Urban Area Transportation," Research Report R63-1, M. I. T. Civil Engineering Department, Cambridge, Massachusetts (1963).
- T-M6] MCFADDEN, D., "The Theory and Practice of Disaggregate Demand Forecasting for Various Modes of Urban Transportation," presented at the Seminar on Emerging Transportation Planning Methods, Florida, pp. 1-55 (1976).
- [T-M7] MCLYNN, J. M. and WORONKA, T., "Passenger Demand and Modal Split Models," Northeast Corridor Project, U. S. Department of Transportation, Record 230 (1969).
- [T-M8] MCNEIL, D. R., "A Solution to the Fixed-Cycle Traffic Light Problem for Compound Poisson Arrivals," Journal of Applied Probability, Vol. 5, pp. 624-635 (1968).
- [T-M9] MERTZ, W. L., "Review and Evaluation of Electronic Computer Traffic Assignment Programs," Highway Research Board Bull, 297, pp. 94-105 (1961).
- [T-M10] MOSHER, W. W., "A Capacity Restraint Algorithm for Assigning Flow to a Transport Network," Highway Research Record 6, pp. 41-70 (1963).
- [T-M11] MURCHLAND, J. D., "Road Network Traffic Distribution in Equilibrium," paper presented at the Conference "Mathematical Methods in the Economics Sciences," Mathematisches Forschungsinstitut, Oberwolfach (1969).
- [T-N1] NETTER, M., "Equilibrium and Marginal Cost Pricing on a Road Network with Several Traffic Types," in Proceedings of the Symposium on the Theory of Traffic Flow and Transportation, G. F. Newell (editor), Springer-Verlag (1971).
- [T-N2] NGUYEN, S., "Une Approche Unifiée des Méthodes d'Équilibre pour l'Affectation du Trafic," Ph.D. Dissertation, Publication No. 171, Department d'Informatique, Université de Montreal (1973).
- [T-N3] NGUYEN, S., "An Algorithm for the Traffic Assignment Problem," Transportation Science, Vol. 8, No. 3, pp. 203-216 (1974).

- [T-N4] NGUYEN, S., "A Mathematical Programming Approach to Equilibrium Methods of Traffic Assignment with Fixed Demands," Publication No. 17, Centre de Recherche sur les Transports, Université de Montréal (1976).
- [T-N5] NGUYEN, S., "Equilibrium Traffic Procedures with Elastic Demands," Publication No. 39, Centre de Recherche sur les Transports, Université de Montréal (1976).
- [T-N6] NGUYEN, S. and JAMES, L., "TRAFFIC - An Equilibrium Traffic Assignment Program," Publication No. 17, Centre de Recherche sur les Transports, Université de Montréal (1975).
- [T-O1] OVERGAARD, K. R., "Urban Transportation Planning: Traffic Estimation," Traffic Quarterly, pp. 197-218 (1967).
- [T-P1] POTTS, R. B. and OLIVER, R. M., Flow in Transportation Networks, Academic Press, New York (1972).
- [T-Q1] QUANDT, R. E. and BAUMOL, W. J., "The Demand for Abstract Transport Modes: Theory and Measurement," Journal of Regional Science, Vol. 6, No. 2, pp. 13-26 (1968).
- [T-R1] ROSENAL, R. W., "The Network Equilibrium Problem in Integers," Networks, Vol. 3, pp. 53-59 (1973).
- [T-R2] ROSENAL, R. W., "Congestion Tolls: Equilibrium and Optimality," Economic Discussion Paper #94, Bell Laboratories (1977).
- [T-R3] RUITER, E. R., "The Prediction of Network Equilibrium - The State of the Art," Transportation System Division, Department of Civil Engineering, Massachusetts Institute of Technology, Cambridge, Massachusetts (1973).
- [T-S1] SHEFFI, Y., "Transportation Networks Equilibrium with Discrete Choice Models," Ph.D. Dissertation, Massachusetts Institute of Technology, Cambridge, Massachusetts (1978).
- [T-S2] SHEFFI, Y. and DAGANZO, C., "A Unified Approach to Transportation Forecasting: Hypernetworks and Supply-Demand Equilibrium with Disaggregate Demand Models," paper presented at the 57th TRB Meeting, Washington, D.C. (1978).
- [T-S3] SMOCK, R., "A Comparative Description of a Capacity - Restrained Traffic Assignment," Highway Research Record 6, pp. 12-40 (1963).
- [T-S4] STEENBRINK, P. A., Optimization of Transport Networks, Wiley London (1974).

- [T-S5] STEFANEK, R. G. and WILKIE, D. F., "Precise Determination of Equilibrium in Travel Forecasting Problem Using Numerical Techniques," Highway Research Record 369 (1971).
- [T-U1] UMTA Transportation Planning System (UTPS) User's Guide, U. S. Department of Transportation, Urban Mass Transportation Administration, Office of Transit Planning, Planning Methodology and Technical Support Division, Washington, D. C. (1976).
- [T-W1] WARDROP, J. G., "Some Theoretical Aspects of Road Traffic Research," Proc. Inst. Civil Engineers, Part II, Vol. 1, pp. 325-378 (1952).
- [T-W2] WEBSTER, F. V., "Traffic Signal Setting," U. K. Ministry of Transport, RRL Technical Report No. 39 (1968).
- [T-W3] WEINTRAUB, A., "Optimal Flows in Network and Games: The Multi-commodity Flow Problem in Integers," Publication 76/21/C, Department of Industrial Engineering, University of Chile (1976).
- [T-W4] WIGAN, M. R. and BANFORD, T. J. G., "An Equilibrium Model of Bus and Car Travel Over a Road Network," Transport and Road Research Laboratory, Report LR 559 (1973).
- [T-W5] WITHEFORD, D. K., "Traffic Assignment Analysis and Evaluation," Highway Research Record No. 6, pp. 1-11 (1963).
- [T-Z1] ZIPKIN, P. H., "Bounds for Aggregating Nodes in Network Problems," Research No. 79A, Graduate School of Business, Columbia University, New York (1978).

Complementarity Problems

- [C-C1] COTTLE, R. W. and DANTZIG, G. B., "Complementary Pivot Theory of Mathematical Programming," Linear Algebra and it's Applications, Vol. 1, pp. 103-125 (1968).
- [C-C2] COTTLE, R. W. and DANTZIG, G. B., "A Generalization of the Linear Complementarity Problem," Journal of Combinatorial Theory, Vol. 8, pp. 79-80 (1970).
- [C-E1] EAVES, B. C., "The Linear Complementarity Problem," Management Science, Vol. 17, pp. 612-634 (1971).
- [C-E2] EILLENBERG, S. and MONTGOMERY, D., "Fixed Point Theorem for Multi-valued Transformation," American Journal of Mathematics, Vol. 68, pp. 214-222 (1946).
- [C-F1] FISHER, M. L. and GOULD, F. J., "A Simplicial Algorithm for the Non-linear Complementarity Problem," Mathematical Programming, Vol. 6, pp. 281-300 (1974).

- [C-F2] FOSDICK, L. D., "Algorithm 431: A Computer Routine for Quadratic and Linear Programming Problems [H]," Arunachalam Ravindran Record 24 (1970).
- [C-K1] KAKUTANI, S., "A Generalization of Brouwer's Fixed Point Theorem," Duke Math. Journal, Vol. 8, pp. 457-459 (1941).
- [C-K2] KARAMARDIAN, S., "The Nonlinear Complementarity Problem with Applications, Part I and II," Journal of Optimization Theory and Application, Vol. 4, pp. 87-98 (1969).
- [C-K3] KARAMARDIAN, S., "The Complementarity Problem," Mathematical Programming, Vol. 2, pp. 107-129 (1972).
- [C-K4] KOJIMA, M., "Computational Methods for the Nonlinear Complementarity Problem," Department of Administration Engineering, Keia University, Yokohama, Japan (1973).
- [C-K5] KOJIMA, M., "A Unification of the Existence Theorems of the Nonlinear Complementarity Problem," Mathematical Programming, Vol. 9, pp. 257-277 (1975).
- [C-L1] LEMKE, C. E., "Bimatrix Equilibrium Points and Mathematical Programming," Management Science, Vol. 11, pp. 681-689 (1965).
- [C-L2] LÜTHI, H. J., "Ein Algorithms Zur Approximation Von Lösungen des Nichtlinear Komplementaritäts Problems," Dissertation No. 5183, Swiss Federal Institute of Technology, Zürich, Switzerland (1973).
- [C-L3] LÜTHI, H. J., "A Simplicial Approximation of a Solution for the Nonlinear Complementarity Problem," Mathematical Programming, Vol. 9, pp. 278-293 (1975).
- [C-N1] NASH, F. J., "Non-Cooperative Games," Annals of Mathematics, Vol. 54, pp. 286-295 (1951).
-
- [C-S1] SAIGAL, R., "Extension of the Generalized Complementarity Problem," Mathematics of Operations Research, Vol. 1, pp. 260-266 (1976).
- [C-S2] SCARF, H., "The Approximation of Fixed Point of a Continuous Mapping," SIAM Journal on Applied Mathematics, Vol. 15, pp. 1328-1343 (1968).
- [C-T1] TODD, M. J., The Computation of Fixed Points and Applications, Lecture Notes in Economics and Mathematical System 124, Springer-Verlag, Berlin (1976).
- [C-T2] TODD, M. J., "A Generalized Complementary Pivoting Algorithm," Mathematical Programming, Vol. 63, pp. 243-263 (1974).

[C-T3] TODD, M. J., Private Communication, February (1977).

Others

- [O-A1] APOSTOL, T. M., Mathematical Analysis, Addison-Wesley (1957).
- [O-G1] GEOFFRION, A. M., "Generalized Bender Decomposition," Journal of Optimization Theory and Applications, Vol. 10, pp. 237-260 (1972).
- [O-M1] MANGASARIAN, O. L., Nonlinear Programming, McGraw-Hill, New York (1969).
- [O-O1] ORTEGA, J. M. and RHEINBOLDT, W. C., Iterative Solution of Non-linear Equations in Several Variables, Academic Press, New York, (1970).
- [O-S1] SAMUELSON, P. A., "Spatial Price Equilibrium and Linear Programming," American Economic Review 42, pp. 283-303 (1952).
- [O-T1] TAKAYAMA, T. and JUDGE, G. G., Spatial and Temporal Price and Allocation Models, North-Holland, Amsterdam (1971).

APPENDIX AA Convergence Property of the Linearization Algorithm

In this section we discuss some convergence properties of the linearization scheme for solving the nonlinear complementarity problem corresponding to a traffic equilibrium model.

Consider a simple network with a single arc, a single mode, and a single O-D pair. The user-equilibrium system for this network can be written as the following nonlinear complementarity problem:

$$(A1) \quad \left\{ \begin{array}{l} [t(h) - u] \cdot h = 0 \\ [h - D(u)] \cdot u = 0 \\ t(h) - u \quad \geq 0 \\ h - D(u) \quad \geq 0 \\ h \geq 0, \quad u \geq 0 . \end{array} \right.$$

We start at any arbitrary nonnegative point $(h^0, u^0) \geq 0$, linearize $t(h)$ at h^0 and $D(u)$ at u^0 , and solve the resulting linear complementarity problem (LCP) to find a new point (h^1, u^1) .

At iteration i , the (LCP) would be:

$$(A2) \quad \left\{ \begin{array}{l} [t(h^i) + (h^{i+1} - h^i) \frac{dt(h^i)}{dh} - u^{i+1}] \cdot h^{i+1} = 0 \\ [h^{i+1} - D(u^i) - (u^{i+1} - u^i) \frac{dD(u^i)}{du}] \cdot u^{i+1} = 0 \\ [t(h^i) + (h^{i+1} - h^i) \frac{dt(h^i)}{dh} - u^{i+1}] \quad \geq 0 \end{array} \right.$$

$$(A2) \begin{cases} [h^{i+1} - D(u^i) - (u^{i+1} - u^i) \frac{dD(u^i)}{du}] \geq 0 \\ h^{i+1} \geq 0, \quad u^{i+1} \geq 0 \end{cases}$$

The procedure is shown in the Figure A.1.

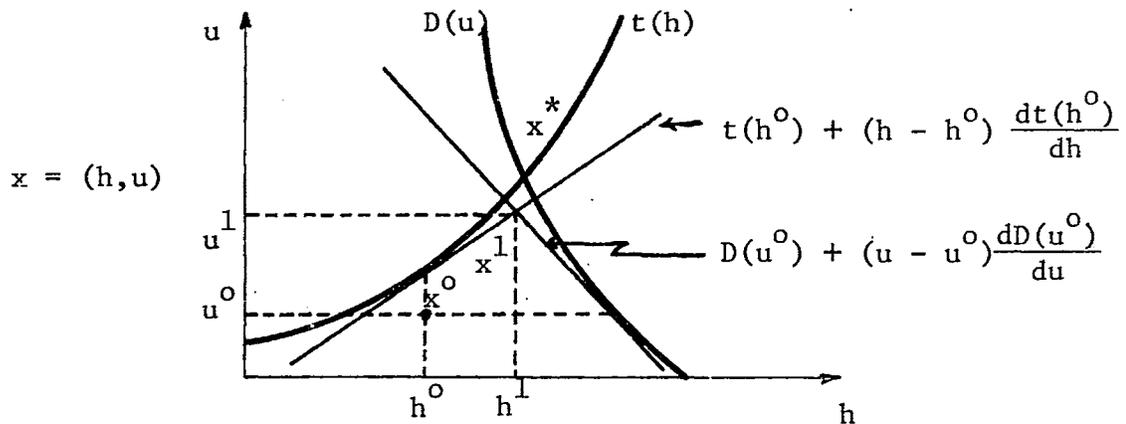


Figure A.1 The Linearization Algorithm

LEMMA A.1: Let $t(h)$ be a continuous differentiable increasing convex function on $h \geq 0$ and let $D(u)$ be a continuous differentiable decreasing convex function on $u \geq 0$. Then (A1) has a unique solution and furthermore the sequence $\{h^i, u^i\}$ generated by solving (A2) will converge to that solution (u^*, h^*) .

PROOF: For simplicity of exposition we assume that

$$t(0) > 0$$

$$D(0) > 0$$

$$\frac{dt(h)}{dh} > \delta \quad \text{for some small } \delta > 0$$

$$\left| \frac{dD(u)}{du} \right| < M \quad \text{for some large } M > 0$$

even though these assumptions are not required.

The proof of uniqueness is clear from the main theorems in chapter 4. Notice that $D(u)$ is a continuous differentiable decreasing function and, therefore, is bounded from above. Thus we prove only the last assertion of the lemma.

Let,

$$C = \{(h,u) \mid h \geq 0, t(h) - u \geq 0, \text{ and } D(u) - h \geq 0\}.$$

For any $(h,u) \in C$, since $D(u)$ is bounded thus h is bounded, and, since $t(h)$ is continuous, thus u is bounded. That implies C is bounded.

Also, the continuity of $t(h)$ and $D(u)$ imply that C is closed, therefore C is compact. Since $t(h)$ and $D(u)$ are convex, for any point in C the solution to the linearization problem (A2) lies in C as well. Therefore, starting at any point (h^0, u^0) in C , the sequence $\{h^i, u^i\}$ generated by solving (A2) will remain in C .

Now, we choose (h^0, u^0) in C such that $h^0 > 0$ and $u^0 > t(0)$.

It is easy to show that $u^i > 0$ at any iteration (intuitively u cannot be zero because it is the minimum travel time). In fact, we show that $\{u^i\}$ is an increasing sequence.

First we show that h^{i+1} can not be zero. Suppose that this assertion is not true and that h^{i+1} is zero. Then, since $u^{i+1} > 0$, by complementarity we have:

$$h^{i+1} - D(u^i) - (u^{i+1} - u^i) \frac{dD(u^i)}{du} = 0$$

or

$$u^{i+1} = u^i + \frac{D(u^i)}{-\frac{dD(u^i)}{du}} \geq u^i .$$

Therefore, the first complementarity equation becomes:

$$\begin{aligned} & t(h^i) + (h^{i+1} - h^i) \frac{dt(h^i)}{dh} - u^{i+1} \\ &= t(h^i) - h^i \frac{dt(h^i)}{dh} - u^{i+1} \\ &\leq t(0) - u^{i+1} < t(0) - u^i < 0 \end{aligned} \tag{A.3}$$

which is a contradiction. Thus $h^{i+1} > 0$ at any iteration. Notice that the first inequality in (A.3) is true because $t(h)$ is convex.

Therefore, if we start at any point in C with $u^0 > t(0)$ we have $h^i > 0$ and $u^i > 0$ for all i , which implies that system (A2) is equivalent to the following linear system:

$$(A.4) \left\{ \begin{aligned} h^{i+1} &= h^i + \frac{-(h^i - D(u^i)) - (u^i - t(h^i)) \frac{dD(h^i)}{du}}{1 - \frac{dt(h^i)}{dh} \cdot \frac{dD(u^i)}{du}} \\ u^{i+1} &= u^i + \frac{-(u^i - t(h^i)) - (h^i - D(u^i)) \frac{dt(h^i)}{dh}}{1 - \frac{dt(h^i)}{dh} \cdot \frac{dD(u^i)}{du}} \end{aligned} \right.$$

These equalities imply that $u^{i+1} \geq u^i$ for any (h^i, u^i) in C , because all terms in the fraction of the last inequality are positive. Therefore

$\{u_i\}$, as an increasing sequence in a compact set, has a limit point, i.e., $\lim_{i \rightarrow \infty} u^i = u^*$. Therefore, for any $\varepsilon > 0$, there exists an $I > 0$ such that, for any $i > I$ we have:

$$u^{i+1} - u^i < \varepsilon .$$

Then (A.4) implies that

$$\frac{(t(h^i) - u^i) + (D(u^i) - h^i) \frac{dt(h^i)}{dh}}{1 - \frac{dt(h^i)}{dh} \cdot \frac{dD(u^i)}{du}} < \varepsilon$$

or

$$\begin{cases} (t(h^i) - u^i)/B < \varepsilon \\ (D(u^i) - h^i)/B < \varepsilon / \left(\frac{dt(h^i)}{dh}\right) < \frac{\varepsilon}{\delta} \end{cases}$$

where

$$B = 1 - \frac{dt(h^i)}{dh} \cdot \frac{dD(u^i)}{du} .$$

Also (A4) implies that

$$\begin{aligned} |h^{i+1} - h^i| &\leq \left| \frac{D(u^i) - h^i}{B} \right| + \left| \frac{t(h^i) - u^i}{B} \right| \cdot \left| \frac{dD(u^i)}{du} \right| \\ &\leq \frac{\varepsilon}{\delta} + \varepsilon \cdot M = \varepsilon \left(M + \frac{1}{\delta} \right) . \end{aligned}$$

Therefore $\{h^i\}$ has a limit point too.

Suppose that $\{\hat{h}, \hat{u}\}$ is the limit point of the sequence $\{h^i, u^i\}$.

To show that $\{\hat{h}, \hat{u}\}$ is an equilibrium point, we know that,

$$\begin{cases} t(h^{i+1}) + (h^{i+1} - h^i) \frac{dt(h^i)}{dh} - u^{i+1} = 0 \\ h^{i+1} - D(u^i) - (u^{i+1} - u^i) \frac{dD(u^i)}{du} = 0 \end{cases} .$$

Since $t(h)$ and $D(u)$ are continuous differentiable functions, therefore at the limit we have:

$$\begin{cases} \lim_{i \rightarrow \infty} t(h^i) + (h^{i+1} - h^i) \frac{dt(h^i)}{dh} - u^{i+1} = t(\hat{h}) - \hat{u} = 0 \\ \lim_{i \rightarrow \infty} h^{i+1} - D(u^i) - (u^{i+1} - u^i) \frac{dD(u^i)}{du} = \hat{h} - D(\hat{u}) = 0 \end{cases}$$

and since (A1) has unique solution thus $(\hat{h}, \hat{u}) = (h^*, u^*)$, which completes the proof. ■

Notice that the coefficient matrix of the linear complementarity system (A2) is positive definite. Therefore (A2) has always a unique solution and B is positive.

REMARK A.1: Without convexity assumptions, the Linearization scheme might not converge. For example:

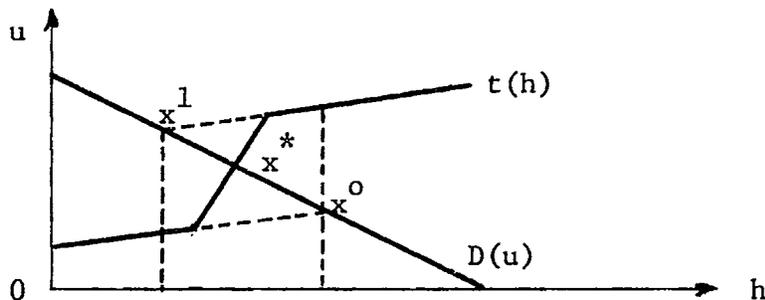


Figure A.2 The Linearization Algorithm Might Not Converge

In this example, if we start at x^0 the algorithm will not converge, but will oscillate between x^0 and x^1 . Note, however, that if at each iteration we linearize at $\frac{1}{n} \sum_{i=0}^{n-1} x^i$ we have convergence. This observation suggests that a modification of the algorithm might converge for nonconvex volume delay and demand functions. ■

REMARK A.2: The convergence properties for this simple case are related to Newton's Methods for solving systems of nonlinear equations (see, for example, Ortega and Renboldt [0-01]). ■

APPENDIX EComputing the Coefficient Matrix

In the notation of section 6.2.5, we stated the volume delay function for path p as:

$$T_p(h) = \sum_{a \in A} \delta_{ap} \cdot t_a(f) \quad \text{for all } p \in P_i^W \text{ and } i \in I$$

where

$$f_a = \sum_{i \in I} \sum_{p \in P_i^W} \delta_{ap} \cdot h_p \quad \text{for all } a \in A.$$

In practice, because of the enormous number of paths, it is not possible and efficient to store the arc-path incidence matrix $\{\delta_{ap}\}$ in a computer. Instead, as we noted in section 6.4, we can store the list of arcs in a subset of all paths, called the set of working paths. If A_p denotes the list of arcs in path p , then we have:

$$T_p(h) = \sum_{a \in A_p} t_a(f) \quad \text{for all } p \in P_i^W \text{ and } i \in I$$

and

$$\frac{\partial T_p(h)}{\partial h_{p'}} = \sum_{a \in A_p} \sum_{a' \in A_{p'}} \frac{\partial t_a(f)}{\partial f_{a'}} \quad \text{for any } p \text{ and } p'. \quad (\text{B.1})$$

By storing and updating values for $\nabla t(f)$ and using (B.1), we can efficiently evaluate the coefficients of the $\nabla t(h)$ matrix. First, notice that $\nabla t(f)$ is usually a highly sparse matrix because, in general, the

flow on each arc depends at most only upon the flow on a few other arcs. Therefore, for each arc, $a \in A$, only a few components of $\nabla t_a(f)$ are non-zero that can be stored in a few words of the computer memory. For the special case when $t_a(f) = t_a(f_a)$ (This has been the case for most of the computational results by other researchers), we have:

$$\frac{\partial T_p(h)}{\partial h_{p'}} = \sum_{a \in A_p \cap A_{p'}} \frac{dt_a(f_a)}{df_a} \quad \text{for any } p \text{ and } p' .$$

In this case, a one dimensional array is enough to store the data for $\nabla t(f)$. Also, when $\nabla t(f)$ is symmetric, as is the case when $t_a(f) = t_a(f_a)$, then $\nabla T(h)$ is symmetric which reduces the computation of the coefficients by one half. (Notice that we did not include this option in our computational results because we wanted to implement the algorithm in as general a form as possible).

Second, because of the decomposition scheme that we use, the dimension of the $\nabla T(h)$ matrix is not very large for each subproblem. In fact, the $\nabla T(h)$ matrix only includes those elements of $\frac{\partial T_p(h)}{\partial h_{p'}}$ for which both p and p' are in the subproblem. Also, to update values for $\nabla t(f)$ after each linearization (flow change), we only require updating those non-zero components of $\nabla t(f)$ for which $a \in A_p$ and p is in the subproblem.

Similar observations apply for computing the coefficients of the $\nabla D(u)$ matrix in the demand portion of the model.