



# Research

**Building Our Way  
Out Of Congestion**

## Technical Report Documentation Page

1. Report No. MN/RC – 2002-01	2.	3. Recipients Accession No.	
4. Title and Subtitle BUILDING OUR WAY OUT OF CONGESTION? HIGHWAY CAPACITY FOR TWIN CITIES		5. Report Date October 2001	
		6.	
7. Author(s) Gary A. Davis and Kate Sanderson		8. Performing Organization Report No.	
9. Performing Organization Name and Address Department of Civil Engineering University of Minnesota 500 Pillsbury Drive S.E. Minneapolis, MN 55455		10. Project/Task/Work Unit No.	
		11. Contract (C) or Grant (G) No. C) 74708 wo) 125	
12. Sponsoring Organization Name and Address Minnesota Department of Transportation 395 John Ireland Boulevard Mail Stop 330 St. Paul, Minnesota 55155		13. Type of Report and Period Covered Final Report 1999-2001	
		14. Sponsoring Agency Code	
15. Supplementary Notes			
16. Abstract (Limit: 200 words)  <p>What would it take to build our way out of congestion in the Twin Cities? As part of this research project, researchers identified a method to answer that question and found a minimal set of highway capacity expansions that would accommodate future travel demand and guarantee mobility.</p> <p>The problem of identifying a set of capacity expansions that are in some sense optimal, while accounting for traveler reaction, is known as a network design problem. A literature review reveals numerous formulations and solution algorithms over the last three decades, but the problem of implementing these for large-scale networks has remained a challenge.</p> <p>This project presents a solution procedure that incorporates the capacity expansion as a modified step in the Method Successive Averages, providing an efficient algorithm capable of solving realistic problems of real-world complexity. Application of this method addresses the network design problem for the freeway system of the Twin Cities by providing a lower bound on the extent to which physical expansion of highway capacity can be used to accommodate future growth. The solution estimates that adding 1,844 lane-kilometers, or 1,146 lane-miles, would be needed to accommodate the demand predicted for the year 2020.</p>			
17. Document Analysis/Descriptors Traffic Assignment Network Design Regional Growth		18. Availability Statement No restrictions. Document available from: National Technical Information Services, Springfield, Virginia 22161	
19. Security Class (this report) Unclassified	20. Security Class (this page) Unclassified	21. No. of Pages 99	22. Price

# **BUILDING OUR WAY OUT OF CONGESTION? HIGHWAY CAPACITY FOR TWIN CITIES**

## **Final Report**

Prepared by

Gary A. Davis  
Associate Professor and Principal Investigator

Kate Sanderson  
Research Assistant

Department of Civil Engineering  
University of Minnesota  
500 Pillsbury Drive SE  
Minneapolis, MN 55455

**October 2001**

Published by

Minnesota Department of Transportation  
Office of Research Services  
395 John Ireland Boulevard – MS 330  
St Paul Minnesota 55155

This report represents the results of research conducted by the authors and does not necessarily represent the views or policy of the Minnesota Department of Transportation. This report does not contain a standard or specified technique.

## TABLE OF CONTENTS

Chapter 1	INTRODUCTION	1
Chapter 2	THE NETWORK DESIGN PROBLEM REVIEWED	6
Chapter 3	FORMULATION OF THE TWIN CITIES NETWORK DESIGN PROBLEM	17
Chapter 4	SELECTION OF A SOLUTION PROCEDURE FOR THE LARGE-SCALE NETWORK DESIGN PROBLEM	26
Chapter 5	THE TWIN CITIES OPTIMAL TRANSPORTATION SUPPLY	34
Chapter 6	CONCLUSIONS	54
	REFERENCES	59
	APPENDIX A NDP Modified MSA Proof	
	APPENDIX B Computer Code	
	APPENDIX C Network Data	

## **LIST OF FIGURES**

Figure 5.1	'Little' Network	37
Figure 5.2	'Sioux Falls' Network	39
Figure 5.3	'Waseca'	42
Figure 5.4	Year 2020 'Do-nothing' Twin Cities AM Peak Hour Speeds	46
Figure 5.5	Year 2020 'Do-nothing' Twin Cities PM Peak Hour Speeds	47
Figure 5.6	Year 2020 Twin Cities AM Capacity Expansion	49
Figure 5.7	Year 2020 Twin Cities PM Capacity Expansion	50
Figure 5.8	Year 2020 Twin Cities AM and PM Capacity Expansion	51

## **LIST OF TABLES**

Table 5.1	Comparison of Network Examples	34
Table 5.2	'Little' O-D Matrix	36
Table 5.3	'Little' Link Parameters and Results	37
Table 5.4	'Sioux Falls' Results	40
Table 5.5	'Waseca' Capacity Expansion Results	43
Table 5.6	Twin Cities Level of Service under 'do-nothing' alternative	45
Table 5.7	Twin Cities Capacity Expansion	52
Table 5.8	Twin Cities Lane Groups of Capacity Expansion	53

## **Executive Summary**

An enduring characteristic of life on the surface of Planet Earth is that we often find ourselves separated from where we want or need to be. A requirement for an economically and socially healthy urban region is thus an ability to move people and goods in a timely manner. In much of the developed world this is achieved by the public provision of the infrastructure on which travel occurs, and an ongoing subject of concern, and of occasional contentious debate, is how to accomplish this in a efficient yet effective manner. In particular, one oft-stated claim is that "we can't build our way out congestion", which is usually taken to mean that in a future world where most travel is by personal auto, the amount of additional highway capacity needed to support this travel at publicly acceptable speeds is so large as to be prohibitive. Yet at the same time there is often a reluctance to aggressively deploy and use alternative transportation technologies and this, coupled with an expectation on the part of the public that transportation agencies are duty bound to provide whatever additional infrastructure is needed to support convenient, unrestricted travel, may mean that "building our way out of congestion" becomes the default transportation policy. If this expectation is unrealistic, it would be useful to find that out sooner, rather than later.

The objective of this project was to determine, for the Twin Cities, what a plausible "building our way out of congestion" policy would look like. More specifically, we sought to find a capacity expansion policy for the Twin Cities' freeway system that satisfied three conditions. First, the expansion should support more or less

unrestricted travel during the peak periods. This requirement was defined quantitatively to mean that all freeway sections should operate at level of service C or better, so that traffic speeds would approximately equal free-flow speeds. Second, the policy should account for travelers' responses to the new facilities. This was accomplished by requiring that level of service C be achieved for the flows resulting from an equilibrium assignment of travel demand. Finally, the policy should in some sense be minimal, in that no lesser policy satisfied the first two conditions. One can always propose clearly unrealistic policies (such as 10 lanes in all directions, on all freeway sections) that could satisfy future travel demand, but only if the minimal acceptable policies are unattainable can it be said that we cannot build our way out of congestion. This was accomplished by seeking a set of capacity expansions that minimized a measure of the land needed for the new facilities, subject to satisfaction of the first two requirements.

The expansion policy satisfying these three conditions can be found by solving a variant of a non-linear optimization problem called a continuous network design problem. Such problems tend to be notoriously difficult to solve for road networks of realistic size, but by exploiting the structure of this variant it was possible to find a straightforward solution algorithm. After verifying, for a number of example problems, that this algorithm produced solutions similar to those produced by the nonlinear optimization code MINOS, we applied the algorithm to the road network model used by the Metropolitan Council for transportation planning. With around 1,100 zones, 7,400 nodes, and 20,000 links, this was much larger than could be solved using MINOS (or any other available optimization code). Taking as input the predicted morning and

afternoon peak period origin-destination matrices prepared by the Metropolitan Council for the year 2020, our algorithm was used to solve for a set of expansions to the Twin Cities freeway network that, at equilibrium, guaranteed approximately free-flow speeds on all freeway sections. We found that this would require approximately 1,150 lane-miles of new capacity, representing a 70% expansion of the existing freeway system.





# CHAPTER ONE

## INTRODUCTION

What would it take to build our way out of congestion in the Twin Cities? The answer to this question has two components: first a procedure must be selected to solve a large-scale Network Design Problem (NDP), and second this procedure must be implemented to solve the Twin Cities NDP, producing a minimal set of highway capacity expansions that accommodate future travel demand and guarantee mobility. The successful implementation of a procedure to solve such a large-scale (though realistically sized) example is shown in this project, which demonstrates that it is possible to solve much bigger problems than have previously appeared in the NDP literature. Additionally, the implementation of this procedure produces a solution to the Twin Cities NDP, which contributes to the discussions concerning the extent to which new highway construction can accommodate the expected growth in travel demand over the next twenty years.

The impetus behind this project is the debate concerning how to deal with congestion. Many metropolitan areas of the US are suffering varying degrees of traffic congestion, and few people need convincing that it is one of the top concerns in the Twin Cities, which was recently ranked as the fifteenth most congested city in the US (Texas Transportation Institute 2001). A poll of Twin Cities residents (Metro State University 2000) puts traffic congestion as the chief metro problem and reviews of the local media also highlight its importance in the local psyche. Recent headlines from the Twin Cities'

Star Tribune newspaper include: 'There's no question: We're big on congestion' (5/8/2001), 'Finding a way out of our traffic jam' (2/11/2001), 'Traffic congestion mounts in Twin Cities' (2/11/2001).

Traffic congestion in the Twin Cities is expected to worsen as forecasted travel demand grows with projected increases in population, housing and jobs. According to Metropolitan Council forecasts, the Twin Cities region is expected to gain approximately 616,000 people, 319,000 households and 312,000 jobs between the years 2000 and 2025. Additionally, the amount of vehicle miles traveled is expected to increase faster than the population--38% compared to 25% (MetCouncil 2000).

Numerous solutions have been proposed to reduce urban highway congestion, and remedies can be separated into supply- and demand-side strategies. Supply-side measures are aimed at increasing capacity of the system, and encompass solutions such as road building and ramp metering. Demand-side schemes are intended to reduce car demand; examples include pricing, travel substitution by telecommuting, and land use and zoning policies.

The Metropolitan Council's Transport Policy Plan (TPP) states that it wants to "keep the region mobile and livable by enhancing economic competitiveness, community and neighborhood livability, mobility options and improving the environment" (MetCouncil 2000). Recognizing fiscal and social realities, they propose measures that include a significantly improved transit system, travel demand management and very limited amounts of highway capacity additions. In a similar vein, the Minnesota Department of Transportation's (Mn/DOT) Transportation Systems Plan

(TSP) has three stated objectives: supporting transit advantages, removing bottlenecks, and improving corridor connections between economic centers (Mn/DOT 2000). Mn/DOT's support for transit in the TSP, and the limited provisions for capacity expansion, seems to indicate the growing importance of multi-modal alternatives, which provide complementary services to the highway system, moving away from their traditional focus on roads.

An active area of debate, however, concerns how effective these multi-modal strategies can hope to be. In Road Work (Small et al. 1989), it is argued that while these policies produce important benefits, and have significant value in improving urban living, none of them will eliminate, or even substantially reduce, congestion. Arguments supporting pricing are presented, but there has been a marked reluctance by decision makers to adopt such policies, as noted by Moore and Thorsnes (1994).

Although it is generally agreed that we cannot rely solely on highway capacity expansion to meet transportation needs, the travelling public are seemingly not opposed to capacity expansions so long as associated costs--such as land taken, community severance, air and noise pollution--are borne by others. Individuals seem to prefer system improvements to actual changes in lifestyle. The poll that cited traffic congestion as the Twin Cities' top concern also named building more roads as the first choice among respondents for reducing traffic congestion (Metro State University 2000).

Transportation policy therefore runs the risk of being undermined by these competing interests. If local interests are strong enough, then 'building our way out of congestion' may become *de facto* policy. Thus, it is proposed that the debate could

benefit from an approach that takes a wider look at the capabilities and limitations of adopting a 'build-only' approach to meeting our transportation needs. To delineate a vision of life under a 'build-only' policy, a solution to a NDP for the expected travel demands at a point in the future is found. This solution identifies a minimal set of capacity expansions that provide relatively uncongested travel on the Twin Cities limited access freeway system, while accounting for traveler reaction to these expansions.

The NDP is a simplified, abstract version of a challenge often faced by the agencies responsible for managing a network of roads and highways--how to choose improvements or additions to the network, in a manner that makes efficient use of resources, in order to achieve some stated objective, such as reducing traffic congestion. The major ideas from the NDP literature are reviewed in Chapter Two, which includes issues about how to model traffic assignment and account for user responses to capacity expansion. Chapter Three describes the mathematical formulation of the NDP to solve 'what it would take to build our way out of congestion'. The initial solution procedure chosen is outlined in Chapter Four. The large-scale size of the Twin Cities NDP meant that the computational effort required by this method was too large to be handled by generally available computational resources. A modified procedure is then described and this was implemented to successfully solve the large-scale Twin Cities NDP. Appendix A contains a proof that this algorithm identifies candidate solutions.

The optimal transportation supply was determined for the AM and PM peak period in the year 2020, and these results are presented in Chapter Five. The main result is that it would take an additional 1,844 lane-kilometers (1,146 lane-miles) on the

existing limited access road system of 2,588 lane-kilometers (1,608 lane-miles) to 'build our way out of congestion' in 2020. Chapter Six then draws together the main conclusions of this project and its contribution to the field--which are the numerical answer to the question, 'what would it take to build our way out of congestion in the Twin Cities?' and the result that this work demonstrates that it is possible to solve such large-scale NDP's. This is followed by some further ideas for future research.

## **CHAPTER TWO**

### **THE NETWORK DESIGN PROBLEM REVIEWED**

In order to determine what it would take to 'build our way out of congestion' in the Twin Cities, it is necessary to determine how best to expand the network, in a manner which makes efficient use of resources, in order to achieve the stated goal of reducing congestion. The NDP encompasses a class of optimization problems which models this process and a substantial body of research explores the various different formulations this problem can take, along with associated solution procedures. To determine the most appropriate formulation to solve the Twin Cities NDP, relevant major ideas are presented and discussed here. Issues that arise include: how to increase capacity, determining optimal supply, modeling user responses, demand issues, and reviewing available solution procedures. Due to the intrinsic complexities of these various issues, the NDP is recognized as one of the most difficult problems to solve in transport (Yang and Bell 1998). Finally, the size of the Twin Cities network is compared to networks for which NDP solutions have appeared in the literature. Solutions that have been published are limited to standard test problems, rather than larger, more realistic examples.

## **A General Description of the Network Design Problem**

The 'Network Design Problem' is a class of related optimization problems, which describe abstract, simplified versions of a challenge often faced by agencies responsible for managing a network of roads and highways--how to choose improvements or additions to the network, in a manner that makes efficient use of resources, in order to achieve some stated objective, such as reducing traffic congestion or air pollution. An agency often encounters this problem when faced with anticipated increases in travel demand that exceed existing network capacity. The problem of identifying a minimal transportation structure under a 'build-only' policy in the Twin Cities can be formulated as a NDP. The purpose of the Twin Cities NDP is to choose a minimal set of improvements to the network in order to reduce traffic congestion.

## **The NDP Bi-level Framework**

The Network Design Problem concerns the design of an optimal amount of transportation supply for a given transportation system. This is achieved by modeling the supply-side decisions that are made by the operator of the transportation system, typically a public authority, and is undertaken in order to achieve some objective, such as the reduction of traffic congestion. Congestion occurs when the level of travel demand approaches the available capacity of a facility and travel times increase well above the average under low demand conditions (Ortuzar and Willumsen 1994). However unlike the network supply, travel demand is not directly under the influence of the decision maker--it is decided by individual trip makers.



Interaction therefore exists between the supply and demand sides, such that as sections of the network are improved, users will respond to these changes by adjusting their own travel behavior. This process can be described as a bi-level programming problem, with the upper level being a supply problem and the lower level being a demand problem. The agency responsible for the network is the leader, attempting to take into account how the users follow. If the leader has prior knowledge of the responses of the followers, this is known in game theory as a 'Stackelberg Game'. The upper level model that describes the decisions that the agency makes concerning the optimal network supply is prescriptive--what *should* they do to best achieve their objectives. The lower level model is descriptive, it characterizes what travelers *would* do if they were pursuing some rational objective (Oppenheim 1995).

### ***Upper Level Problem***

#### *Continuous and Discrete System Improvements*

Adding capacity to a transportation network can be achieved in two ways--either through enhancements of the existing network, or by the addition of new parts to it. This distinction poses the NDP in either a continuous or discrete form. The continuous version, as described by Bell and Iida (1997), takes as given the topology of the network, and is concerned with modifying parameters of the network, for example, increasing link capacity. The discrete version concerns changes to the actual topology of the network--the addition or removal of links. In mature urban areas, such as the Twin Cities, the roadway network is now essentially fixed in its topology. There is little opportunity to

build entirely new roads and transportation planning policy, where it deals with the expansion of the network, deals with the continuous form--increasing the capacity of existing roads, rather than building new ones. Therefore, the continuous form of the NDP is relevant to this problem.

### *Determining the Optimal Transportation Supply*

One can imagine that in practice a highway agency faced with anticipated increases in travel demands and a given network may decide to specify several different sets of capacity improvements which accommodate these increased demands, and then analyze each of these proposed solutions separately. Although all of these proposals may be feasible solutions to the problem, it is conceivable that, in addition, there are many more alternative possible solutions. Fiscal and social responsibilities make it preferable that the solution uses the minimal amount of resources that satisfy the stated objectives. A suitable procedure to achieve this would be to use travel demand forecasts for a future year, assign traffic and then analyze the system performance to determine what set of capacity expansions will relieve congestion. A mathematical problem to solve this would minimize an objective, such as the amount of capacity expansions, subject to satisfying specified volume to capacity ( $V/C$ ) ratios, associated with a given level of service (LOS) requirement.

Unfortunately, capacity improvements that are optimal under existing conditions can become sub-optimal once travelers respond to them, and it is possible that congestion can become worse on the expanded network. This is illustrated by Braess'

Paradox, whereby the addition or enhancement of facilities in a congested network without taking into account the reaction of users may actually increase network-wide congestion (Friesz 1985). Therefore it is essential to account for travelers' responses and methods of doing this are discussed in the next section.

### ***Lower Level Problem***

#### *User Equilibrium*

The lower level problem models how travelers respond to system changes. To account for how the link volumes will change in response to capacity increases, it is necessary to introduce a descriptive model of the way in which network travelers select their routes from origin to destination, and how these route choices determine the link volumes. The process of choosing routes should reflect rational decision-making behavior by trip-makers.

The common assumption in network design is that, when left to their own devices, travelers independently search for the lowest-cost route from origin to destination. A descriptive model of this individually rational behavior would require that each traveler pursue their own interests individually (or selfishly) without regard to the interests of others. The assumption of individualistic rationality in traffic assignment produces what is known as Wardrop's Equilibrium, where the search for these minimum paths terminates when, at user equilibrium, no traveler can decrease his or her travel costs by unilaterally switching to another route (Bell and Iida 1997).

Equilibrium conditions, by definition, imply that a state has been reached in the distribution of travel activity (demand and supply) in which no further changes, on the part of either individual travelers or travel suppliers are expected, provided that the prevailing conditions do not change. A criticism of the equilibrium model is that a point in time where the above conditions are satisfied is unlikely to occur in the real world. Although alternative models, such as the disequilibrium formulation (Oppenheim 1995), have been studied, the equilibrium concept remains the dominant assumption, largely because it leads to tractable formulations and the rationale seems plausible, at least for situations where the decision making rate of travelers is much higher than the rate at which network changes occurs.

#### *Stochastic and Deterministic User Equilibrium*

In network design, two forms of stating the user equilibrium conditions are encountered: deterministic user equilibrium (DUE) and stochastic user equilibrium (SUE). DUE assumes that all travelers have perfect information regarding travel time and perceive costs identically, so all trips are made on minimum cost routes. SUE relaxes the assumption of perfect information, so that costs are perceived differently by different travelers, leading to a spread of responses, with trips being assigned to routes proportionally and low cost routes receiving a greater proportion of trips (Bell and Iida 1997).

While it is realistic to expect travelers to prefer least cost paths, it is unlikely that travelers possess perfect information about the comparative time costs of routes, so the

SUE assumptions, which relax the assumption of perfect information, seem closer to reality than DUE. Although most of the original NDP work adopted DUE assumptions, this has changed since certain tractability advantages associated with the SUE were identified (Davis 1994). Recent work has adopted SUE (Yang and Bell 1998, Huang and Bell 1999) because of these advantages, as does the NDP formulated for this research.

### **Travel Demand**

The lower side of the bi-level problem deals with the users of the network, the demand-side. Traditionally, the NDP takes travel demand as a fixed input, so that the travel choice is restricted to route choice (assignment). An alternative formulation (Huang and Bell 1999) solves for the case where demand is given by a function and the equilibrium demands and their distributions are obtained by solving an elastic-demand user equilibrium model. For SUE assumptions, the demand is constructed between an origin-destination (O-D) pair as a continuous and decreasing function of the expected, perceived cost between that O-D pair. This is realistic as travel times do play an important role in determining trip decisions. Another alternative to using a fixed O-D set is to apply a model that provides trip distribution in addition to assignment. The distribution model would provide origin and destination demand pairs based on travel costs.

While both these alternative formulations present plausible models of travel behavior, the time scale on which these decisions are made is much longer-term than the assignment problem, which is a daily choice. Work trips are assumed to constitute the

majority of AM and PM peak hour trips, which are usually those of concern in designing optimal supply for a network. Where to live and work are long-term decisions made over periods of perhaps years, as opposed to traffic assignment, which is based on the daily route choice. A commonly held assumption is that for short periods of time (weeks to months), the number of people deciding to travel to a work destination is fixed. Similarly, the decision of where to live is usually made on a much longer time scale than route choice. An assumption used in modeling to account for these differences between decision variables and the time scales on which they are made, is that O-D choices are exogenous to the analysis of trip assignment and therefore a fixed O-D matrix can be used (Ben-Akiva and Lerman 1985, p. 326).

The effects of elastic demand may be significant and closer consideration would be desirable. However, as no solutions to large-scale problems have previously appeared in the literature, it is reasonable to find a procedure that can solve the inelastic problem first, before looking at building more sophisticated systems incorporating elastic demand. It should be borne in mind though that the elastic demand assumption is likely to mean that reducing congestion may induce *more* trips. Over a longer period of time than that of the forecast year studied here, additional trips may be generated, so the requisite capacity expansion to reduce congestion would be even larger.

## **Solution Algorithms**

Various approaches to solving the network design problem have been formulated and a brief recap of the evolution of NDP research will provide a framework for the assumptions employed in formulating and solving the Twin Cities NDP.

An early heuristic called the Iterative Optimization Assignment algorithm (IOA) was employed by Steenbrink (1974). This procedure iteratively solves the system optimal network design problem with fixed link flows and the user equilibrium route choice problem with fixed network parameters. This heuristic algorithm has the advantage of computational efficiency, but has been shown to converge to solutions quite different from the optimal equilibrium network design (LeBlanc 1975). This is due to the fact that the IOA gives an exact solution for a Cournot-Nash game, since the solution of one problem only takes into account the solution of the other from the previous iteration, rather than for the more appropriate, Stackelberg game (Bell and Iida 1997).

Abdulaal and LeBlanc (1979) sought the optimal design parameters directly by using the Hooke-Jeeves pattern search method and using DUE assumptions. Unfortunately, the function relating the DUE to the capacity increases is non-differentiable (Abdulaal and LeBlanc 1979), which means that search algorithms that do not require derivatives of the cost function must be employed to solve the NDP. This in turn means that numerous DUE assignments must be performed at each iteration of the search routine, resulting in algorithms that are computationally demanding.

Other heuristic algorithms with reduced computational burden have been proposed for finding an approximate solution to the NDP, but none of them can be guaranteed to converge to the global or even a local optimum. Yang and Bell (2001) review a number of solution algorithms including Tan's DUE problem, which is expressed by a set of non-linear and non-convex, but differentiable, constraints in terms of path flows, and Marcotte's transference of the NDP into a single level equivalent differentiable optimization problem. However the number of constraints required is very large, and therefore computationally burdensome.

Suwansrikul et al. (1987) developed a heuristic method, equilibrium decomposed optimization (EDO) that appears to find nearly optimal solutions. In this algorithm, the direction of the search for the optimal design of each link being considered for improvement is guided by re-evaluating the reaction of the network users to the design changes.

Friesz et al. (1992) applied simulated annealing (SA) to solve the NDP and obtained some encouraging results. However although the algorithm is potentially globally convergent, this is achieved at the expense of extensive numerical computations and numerous searches.

If one switches from a DUE to a stochastic user equilibrium approach to describe the route selection behavior of travelers, the continuous NDP can be posed as a tractable, albeit large-scale, differentiable non-linear programming problem (Davis 1994). Thus, standard non-linear programming algorithms, such as sequential quadratic programming, can be used to solve the resultant non-linear program. Therefore by using the stochastic



user equilibrium approach in traffic assignment it may be able to represent the real situation more closely, and by switching to SUE assignment many of the technical difficulties that arise using DUE constraints can be avoided. Due to these numerical issues, SUE is the assignment model adopted for the formulation of the NDP is used in this project.

### **The Large-Scale NDP**

To date the largest NDP solution presented in the literature is the 76 link, 24 zone 'Sioux Falls' planning network, which was originally described by LeBlanc (1975) and subsequently used by numerous authors, including Suwansirikul et al. (1987), Friesz et al. (1992) and Davis (1994). Other networks of interest, such as the Twin Cities, are generally of a much larger scale than the 'Sioux Falls' example. The Twin Cities planning network used in this project has 20,380 links, and 7,397 nodes. Comparing the size of the 'Sioux Falls' problem to that of the Twin Cities, demonstrates the magnitude of the dimensional disparity between the computational experiments presented in the literature and a realistically sized problem. This difference is important because a finding has been that as the size of a problem increased, large-scale applications have proved too computationally expensive to be solved with available solution methods. An algorithm to solve a large-scale NDP could provide a useful planning tool, capable of handling real world complexity, whereas to date NDP's have remained mostly in the realm of academic interest.

# **CHAPTER THREE**

## **FORMULATION OF THE TWIN CITIES**

### **NETWORK DESIGN PROBLEM**

In this chapter the NDP is formulated as an optimization problem that will solve the research problem by finding the minimal set of capacity expansions that guarantee a specified LOS to travelers while taking the users' responses into account. To build an optimization problem, an objective function and appropriate constraints must be chosen. Nomenclature to describe the network in an abstract manner is discussed, and then used to describe how the NDP has been traditionally posed. Some modifications to the traditional formulation are then proposed to transform it into a form that can solve the research question posed. How, and why, this proposed NDP formulation differs from the traditional one is also discussed.

#### **Network Nomenclature**

The NDP models a simplified, abstract version of an agency's decision-making process of how to expand the road network. To construct an abstract version of the road network of interest some nomenclature must be introduced. The region's roadway system can be represented as a network consisting of a set of points, or nodes, connected by directed links. Of this set of nodes, a sub-set acts as origins and destinations for travel demand, and such nodes are commonly called centroids. The total travel demand

between each O-D pair is specified for some planning interval (such as the AM period hour), and quantified in the form of an O-D matrix. As discussed in the previous chapter, this work will focus on solving a system in which this O-D matrix is known *a priori*.

The road network consists of  $n$  links in the network, indexed  $k = 1, \dots, n$ . Associated with each of these links are three quantities:  $x_k$  denoting the volume of traffic on link  $k$ ,  $y_k$  denoting the proposed expansion of  $k$ 's capacity and  $z_k$  denoting the current capacity of link  $k$ . Thus the capacity after expansion of link  $k$  will be the sum of  $y_k$  and  $z_k$ . The O-D pairs in the network are indexed by  $j = 1, \dots, m$  and  $d_j$  is the demand between O-D pair  $j$  during the planning interval.

### **Formulation of the NDP**

In order to model how an agency adopting a 'build only' policy would go about the deciding which roads to expand, some modifications were made to the traditional NDP. Previous authors (for example Friesz et al. 1992, Davis 1994) have posed the continuous network design problem as the minimization of an objective function (usually the travel and construction costs), subject to flows being a user equilibrium flow pattern. However it will be argued that a slightly different problem formulation models the decision making process of an agency more realistically.

### *The Traditional NDP Objective Function and Constraints*

To understand how the modified Twin Cities NDP differs from its predecessor, the traditional NDP objective function and constraints are described. The traditional NDP minimizes a system cost function

$$\sum_{k=1}^n [x_k t_k(x_k, z_k + y_k) + \alpha g_k(y_k)] \quad (3.1)$$

where the first component gives the travel time costs and the second term the construction costs. Each link is assumed to have a known construction cost function,  $g_k(y_k)$ , which gives the cost of increasing link  $k$ 's capacity by  $y_k$ . The scaling coefficient  $\alpha$  converts units of construction cost into units of daily travel cost and  $t_k$  is the time taken to traverse link  $k$ .

The travel time,  $t_k$ , on a road increases as the flow increases on a road because of traffic congestion. To account for this the travel time is generally taken to be a non-linear function of the total traffic flow on the road. The functional form used for travel costs is the Bureau of Public Roads (BPR) function

$$t_k(x_k, y_k + z_k) = t_k^0 \left( 1 + b_k \times \left( \frac{x_k}{y_k + z_k} \right)^4 \right) \quad (3.2)$$

where  $t_k^0$  is the free-flow travel time on link  $k$  and  $b_k$  is a coefficient scaling the rate at which congestion increases the costs (Abdulaal and LeBlanc 1979).

The objective function (equation 3.1) is subject to two constraints: firstly, that  $y_k$  is greater than, or equal to, zero, and secondly, that  $x_k$  is a user equilibrium flow pattern. In Chapter Two the continuous version of the NDP was deemed appropriate, so the number of network links is fixed and the capacity expansion is denoted by an expansion variable associated with candidate links. The first of these constraints requires that the capacity expansion is greater than, or equal to, zero. This means that the link is either expanded by adding capacity, or that it remains as is. The case where capacity is removed, that is where the value is negative, is not considered in this model. The other constraint is that  $x_k$ , the flow on each link  $k$ , constitutes the user equilibrium flow pattern--this model is more complex to describe and is where much of the computational work is expended. The second constraint is necessary, because, unlike the capacity increases, the link volumes are not directly under the influence of the decision maker. To model this it is assumed that drivers will seek the lowest-cost route from their origin to destination, and that the point at which no traveler can decrease his or her travel costs by switching to another route produces an equilibrium assignment of traffic to the network.

### *The Stochastic User Equilibrium Conditions*

Given a set of O-D trip rates,  $d_j$ , the SUE conditions are characterized by the solutions to the set of nonlinear equations

$$x_k = \sum_{j=1}^m d_j q_{jk} \quad \forall k \quad (3.3)$$

where  $q_{jk}$  denotes the probability that a trip between O-D pair  $j$  uses link  $k$ , with the functional form

$$q_{jk} = \sum_r \delta_{jrk} p_{jr} \quad (3.4)$$

where

$$\delta_{jrk} = \begin{cases} 1, & \text{if link } k \text{ lies on path } r \text{ between O - D pair } j \\ 0, & \text{otherwise} \end{cases} \quad (3.5)$$

and an explicit function of route choice probabilities can be found using logit route choice

$$p_{jr} = \frac{e^{-\theta t_{jr}}}{\sum_s e^{-\theta t_{js}}} \quad (3.6)$$

where  $\theta$  is a parameter which is determined by the variance of the route cost perception errors and  $t_{jr}$  is the cost of traversing path  $r$  between O-D pair  $j$ , and assuming that the path cost is simply the sum of the corresponding links costs, then

$$t_{jr} = \sum_{k=1}^n \delta_{jrk} t_k(x_k, y_k + z_k) \quad (3.7)$$

### ***Re-posing the NDP Objective Function***

A major criticism of the traditional NDP formulation is the way in which the objective function (equation 3.1) comprises weighted components of daily costs of travel time and long-term construction costs. The inclusion of the construction costs with the

travel time in the objective function in this way requires assigning a monetary value to the travel time of users of the network to compare the trade-off between the construction costs and travel time.

LeBlanc (1975) noted the difficulties in assigning a defensible value to the monetary equivalence of travel time and discussed methods of separating out the objective function components. This is often achieved through the inclusion of construction costs as a budget constraint, thus formulating the optimal amount of capacity expansion as the set whose user optimal flows have the least total vehicle-hours of travel. To overcome these difficulties, an alternative form is proposed that removes the travel costs from the objective function, and instead imposes a level of service constraint, which has the additional advantage that it guarantees a specified level of mobility.

#### *Guaranteeing Mobility by introducing a LOS constraint*

In Chapter Five, the LOS for solutions found via the traditional formulation are calculated. These values show that for the often used 'Sioux Falls' network, V/C as high as 3.70 occurs at the optimal solution, when the objective function is as stated in equation 3.1. V/C is used as an indication of LOS, and values higher than 1.0 indicate congested conditions. It seems unlikely an agency could secure funding without assuring some acceptable LOS, since when the public is pressing for highway expansion they are doing so in order to improve their own travel speeds, and are much less concerned with the total system costs. Also, the question more commonly asked by planning agencies

would seem to be 'how can we satisfy the traveling public through some efficient allocation of resources?'. To account for these issues, a LOS constraint was imposed. In the Highway Capacity Manual (HCM), the LOS criteria for basic freeway sections are based on V/C ratios, so that as the volume approaches capacity, the associated speeds decline (Roess et al. 1998). V/C ratios are found by taking the link flow (volume), and dividing it by the total capacity of that link. The LOS constraint is of the form

$$\frac{x_k}{y_k + z_k} \leq \tilde{c}_k \quad (3.8)$$

where  $\tilde{c}_k$  is the specified V/C ratio associated with a given LOS. Rearranging to place the independent variables on one side and the fixed quantities on the other, gives a linear equation

$$x_k - \tilde{c}_k y_k \leq \tilde{c}_k z_k \quad (3.9)$$

#### *An Alternate Objective Function*

Removing the travel time costs from the objective function (equation 3.1) leaves the construction costs. These costs are usually dealt with in the NDP literature by the following theoretical quadratic formulation (e.g. Abdulaal and LeBlanc 1979, Davis 1994)

$$g_k(y_k) = d_k y_k^2 \quad (3.10)$$



Abdulaal and LeBlanc (1979) explored a number of construction cost formulations, including this quadratic form in order to examine how the solution responds under different cost formulations, such as concave versus convex. These cost formulations have no realistic basis and a literature search revealed a dearth of reliable construction cost functions.

Reconsidering the purpose of this research, which was to determine 'what it would take to build our way out of congestion', suggested an alternative objective function--to minimize land taken. That is

$$\min(\sum_k l_k y_k) \quad (3.11)$$

where  $l_k$  is the length of link  $k$ , and  $y_k$  is the capacity expansion for link  $k$ . This is appropriate because one of the primary concerns associated with highway expansion by agencies and the public seems to be the taking of land for highway construction. The new objective function (equation 3.11) is subject to the same two constraints as the traditional formulation, which were

$$x_k = \sum_{j=1}^m d_j q_{jk} \quad (3.12)$$

$$y_k \geq 0 \quad (3.13)$$

and the additional LOS constraint

$$x_k - \tilde{c}_k y_k \leq \tilde{c}_k z_k \quad (3.14)$$

This alternative formulation was also attractive because it produces a linear objective function, which makes for simpler computations. Additionally, if an analyst wished to provide some function for construction cost, post-processing could be used to make this calculation.

### **Summary**

In this chapter the research question is formulated as a mathematical optimization problem to find the minimal set of capacity expansions which guarantee a specified LOS to travelers and which takes into account user response. Although this formulation now has a linear objective function, it remains a large-scale, non-linear programming problem for which a suitable solution procedure must be selected, this process is explored in Chapter Four.

## **CHAPTER FOUR**

### **SELECTION OF A SOLUTION PROCEDURE FOR THE LARGE-SCALE NETWORK DESIGN PROBLEM**

The NDP formulation posed in Chapter Three determines what it would take for the Twin Cities to accommodate future travel demand solely through highway capacity expansion. A procedure to solve a large-scale sized NDP must now be selected. There are a number of standard techniques for solving non-linear programming problems and several of these proved successful on test problems. The MINOS optimization software was selected from these for large-scale implementation. When the Twin Cities problem was investigated, however, its size meant that the computational effort required was too large to be handled by available computer resources. An alternative procedure, based on modifications to the Method of Successive Averages (MSA), was then developed and used to solve the Twin Cities NDP.

#### **Initial Selection of an NDP Solution Procedure**

Davis (1994) posed a differentiable and tractable version of the NDP, which meant that standard algorithms for solving non-linear programs could be, and were, employed to produce results for various NDP computational tests. The computation of the minimum of a differentiable objective function, subject to a set of differentiable constraints, is a problem for which numerous solution algorithms are available. Candidate methods initially considered to solve this nonlinear problem included

simulated annealing and the two methods previously used by Davis (1994), sequential quadratic programming (SQP) and the reduced-gradient algorithm. Simulated annealing was rejected after a literature review revealed that a large number of function evaluations would be required, which would be computationally prohibitive for this large problem. SQP was used to solve the standard numerical test network, 'Sioux Falls', with ten links being candidates for expansion. MINOS (Murtagh and Saunders 1988), which uses reduced-gradient algorithms, was then considered because of its success in various fields solving large non-linear optimization problems. MINOS was successful in solving the 'Sioux Falls' test problem, and was computationally much quicker than SQP. It was then used to solve the larger 'Waseca' network problem, with 126 links as candidates for expansion. MINOS was therefore selected to tackle the Twin Cities problem.

### **Implementation of Solution Procedure with MINOS**

For each trial solution in a MINOS implementation, the objective and constraint functions must be evaluated, along with their derivatives. Whenever possible, in optimization problems, computer code evaluating gradients of the non-linear objective function and non-linear constraints should be provided. For the NDP, the gradient for construction cost is straightforward. For the non-linear constraints, Davis (1994) provided the first tractable gradient for the stochastic user-equilibrium constraints using Van Vliet's (1981) modification to the Dial Assignment algorithm. The Dial Assignment is a method of stochastic network loading, the algorithm used to implement the procedure is commonly known as STOCH (Sheffi 1985, pp. 286-297).

Using MINOS as the solution procedure, the process of solving the NDP begins with inputs that model the supply and demand sides of the system. The supply-side (the network) consists of connectivity information (from-nodes and to-nodes of links), and the associated link properties such as link length, capacity and free-flow speed. The demand side is given by an origin-destination matrix. The shortest-paths are then calculated based on current travel times. Initially these are based free-flow speeds, but in later iterations are based on congested travel times (equation 3.2) to account for the reduction in speeds as flow increases on a link. Once the shortest-paths are identified, the nodes are indexed according to the travel costs, and the traffic is assigned, using Dial's method. An equilibrium assignment is then found using the MSA (Sheffi 1985, pp. 324-331) without any capacity expansion, in order to provide the optimizer with a reasonable starting point. The next step is to start the optimization procedure. During the optimization, the constraint function values (equilibrium flows, non-negativity and LOS) and their derivatives are calculated, as are those for the objective function and its derivatives, in order to determine the search direction. For the constraint equilibrium values and derivatives to be calculated, full shortest path, indexing and assignment sub-routines must be performed. This constitutes the largest component of the computational time at each major iteration. The optimization algorithm terminates when the objective cannot be reduced by more than a user-defined tolerance, and all of the constraints are satisfied. The algorithm will also terminate with a sub-optimal, but feasible, solution after a user-defined maximum number of iterations, or if no new search direction can be found to improve the solution. Global optimality of the solution cannot be guaranteed by

most gradient-based algorithms, but is strongly indicated if the same solution is found having started at a number of initial conditions.

While MINOS successfully solved the NDP for test problems ranging from 7 to 184 links, a successful implementation for the Twin Cities problem with 20,380 links was not accomplished with this method. The size of the constraint Jacobian matrix that MINOS requires depends on the following parameters: the number of variables--flow on each link plus expansion of capacity on candidates--by the non-linear constraints--that is, each link. The Twin Cities network has 20,380 links, and the candidate set considered included 1,317 links (which constituted the metered and unmetered freeway system). So the size of the Jacobian was 21,697 by 20,380--a dense matrix consisting of over 400 million elements. One of the advantages of MINOS is that if the sparsity pattern of the Jacobian can be identified *a priori*, it can be exploited to reduce the memory space utilized and the element updating required. Unfortunately, the dense matrix approach had to be used, as the sparsity pattern of the Jacobian could not be predicted.

The large, dense Jacobian meant that the problem size was greater than could be accommodated in the memory available on most personal computers, so the problem could not be compiled and run. Use of supercomputing resources enabled the problem to be compiled and to run, but after twenty-four hours, the program had not completed an iteration, and so this approach was abandoned.

Various alternate methods were investigated which attempted to circumvent the problem with the large Jacobian. A modified procedure, capable of solving the same problem, was proposed and proved successful. This procedure uses the MSA and does

not require any derivative information. This considerably simplified the computational requirements, and enabled a solution to the Twin Cities problem to be determined.

### *The Modified MSA Solution Procedure*

The modified MSA procedure solves the same formulation as that used above with MINOS as implementation tool, which was the formulation derived in Chapter Three. That is

$$\min(\sum_k l_k y_k) \tag{4.1}$$

subject to the following constraints

$$x_k = \sum_{j=1}^m d_j q_{jk} \tag{4.2}$$

$$y_k \geq 0 \tag{4.3}$$

$$x_k - \tilde{c}_k y_k \leq \tilde{c}_k z_k \tag{4.4}$$

To understand the modified solution procedure, the decomposition steps that transform this problem are now presented. Suppose for the time being that the link flows,  $x_k$ , are given. Rearranging the objective function (equation 4.1), the sum of the minimum for each link can be taken instead

$$\min\left(\sum_k l_k y_k\right) = \sum_k \min(l_k y_k) \quad (4.5)$$

and rearranging the level of service constraints

$$x_k - \tilde{c}_k y_k \leq \tilde{c}_k z_k \Rightarrow y_k \geq \frac{x_k}{\tilde{c}_k} - z_k \quad (4.7)$$

The problem is now reduced to a set of one-dimensional linear programs that have the solution

$$y_k = \max\left(0, \frac{x_k}{\tilde{c}_k} - z_k\right) \quad (4.8)$$

The equilibrium constraint (equations 4.2) depends on the demand between origins and destinations and the probability that a trip between a given origin and destination pair uses a certain route (equation 3.4). The route choice probability uses the logit form (equation 3.6), and depends upon the link cost functions (equation 3.2). So the equilibrium constraints depend on  $x_k$  and  $y_k$  only via the link cost functions.

Substituting the values from (4.8) into the link cost functions (equation 3.2) gives two alternative forms depending on the value. When  $y_k$  is zero the function depends only on the variable link flow



$$t_k(x_k, z_k + y_k) = t_k^0 \left( 1 + b_k \left( \frac{x_k}{z_k} \right)^4 \right) \quad (4.9)$$

However when the link is congested, and  $y_k$  is greater than zero, then  $y_k$  equals  $\frac{x_k}{\tilde{c}_k} - z_k$ , so the link cost functions simplify to a constant, as shown

$$t_k(x_k, z_k + y_k) = t_k^0 \left( 1 + b_k \left( \frac{x_k}{\frac{x_k}{\tilde{c}_k} - z_k + z_k} \right)^4 \right) = t_k^0 \left( 1 + b_k (\tilde{c}_k)^4 \right) \quad (4.10)$$

It can be demonstrated that these are necessary conditions for a solution of the NDP posed (see Appendix A for proof), and computational tests have shown that the solutions arrived at by both the MINOS non-linear programming approach and this modified procedure are the same.

The step-by-step procedure for implementing this modified algorithm is follows.

### MSA algorithm with link expansion modification step

- Step 0: Initialization. Perform a stochastic network loading based on a set of initial travel times ( $t_k^0$ ), generating a set of link flows ( $x_k^n$ ). Set counter n to 1.
- Step 1: For each link k, set  $y_k^n = \max\left(0, \frac{x_k^n}{\tilde{c}_k} - z_k\right)$
- Step 2: Update travel times for each link based on current link flows and capacity,  $t_k^n(x_k^n, z_k + y_k^n)$ , then finding minimum paths via the Moore-Pape shortest path algorithm (Sheffi 1985, pp. 122-129 and Netlib 2000), ranking the nodes according to increasing cost.
- Step 3: Perform a stochastic network loading assignment procedure based on the current set of link travel times ( $t_k^n(x_k^n, z_k + y_k^n)$ ), yielding the auxiliary link flow pattern ( $x_k^{*n}$ ), using Dial's Algorithm.
- Step 4: Find the new flow pattern by setting  $x_k^{n+1} = x_k^n + \frac{(x_k^{*n} - x_k^n)}{n}$
- Step 5: If converged then stop. If not increment counter n and go back to step 1.

This procedure was then used to solve the Twin Cities Network Design Problem on a personal computer--results are presented in the Chapter Five.

## CHAPTER FIVE

### TWIN CITIES OPTIMAL TRANSPORTATION SUPPLY

An optimal transportation supply for a network can be determined by the solution to a Network Design Problem. To date, no algorithm has been successfully demonstrated to solve a large-scale NDP. In Chapter Four a solution procedure was selected to implement, and this chapter provides the numerical evidence that this algorithm is capable of solving large-scale NDP's.

Before attempting to solve a computationally expensive problem, such as the Twin Cities NDP, it is wise to test a procedure on smaller numerical examples. This procedure is tested on two previously presented networks (Davis 1994), and two additional, larger, more realistically, sized networks. The comparative size of the four networks used to test this algorithm are shown in Table 5.1.

**Table 5.1** Comparison of Network Examples

<b>Network Name</b>	<b>LITTLE</b>	<b>SIoux FALLS</b>	<b>WASECA</b>	<b>TWIN CITIES</b>
<b>Number of Nodes</b>	6	24	69	7,393
<b>Number of Links</b>	7	76	184	20,380
<b>No. of Candidates for Expansion</b>	7	10	136	1,317

The following numerical experiments review the different problems investigated in this project. Initially, the 'Little' and 'Sioux Falls' problems are examined, in order to demonstrate the limitations of the traditional NDP formulation, as compared to the approach advocated here, in which the LOS is explicitly considered as a constraint, and the objective function is posed simply in terms of the land requirements. Next, the larger 'Waseca' problem is investigated using the constrained LOS approach. This problem is somewhat larger when compared to the traditionally investigated networks, and so provides an excellent testing ground for the comparison of the different modeling approaches and optimization algorithms described in this work. The results of this comparative exercise are presented in this chapter--in particular, the full MINOS solution which requires all derivatives is compared to the simpler MSA approach in order to illustrate that the latter approach provides acceptable results, while requiring much less solution space, making it a good algorithm to investigate much larger problems. Finally, the 'Twin Cities' problem and solution are presented, demonstrating that the methods developed here are able to provide a solution to the largest system yet investigated with NDP.

The inputs required to solve this formulation of the NDP are models of the demand- and supply-sides of the system, that is a network representing the roads and highways and a set of trip demands between origin and destination nodes. These inputs were then fed into a computer program developed (see Appendix B for code) to implement the iterative NDP solution procedure, as described in Chapter Four.

## Numerical Experiments

Link characteristics and demand matrices for large networks comprise lengthy data tables, and as such, the network data is referenced at appropriate points rather than repeated here. The exception is the 'Little' network data, given in Tables 5.2 and 5.3, which is provided to give an example of the input parameters required.

### *'Little' Network*

The first experiment was on the seven link and six node 'Little' network, shown in Figure 5.1. The demand matrix is given in Table 5.2 and the link characteristics for this network are shown in Table 5.3--these are from Davis (1994). The solution to the NDP is calculated using the selected modified MSA procedure and these results are presented in Table 5.3, along with the V/C associated with Davis' (1994) solution. This latter solution was found using a standard formulation, in which a weighted sum of user and construction costs was minimized (see equation 3.1).

The level of service constraint used for this test is a volume to capacity ratio of 1.0. This means that with the given demand, all these candidate links operate at level of service volume to capacity ratios of 1.0 or better.

**Table 5.2** 'Little' O-D Matrix

<b>O-D Pair</b>	<b>Travel Demand</b>
1-5	500
1-6	400
2-5	400
2-6	600

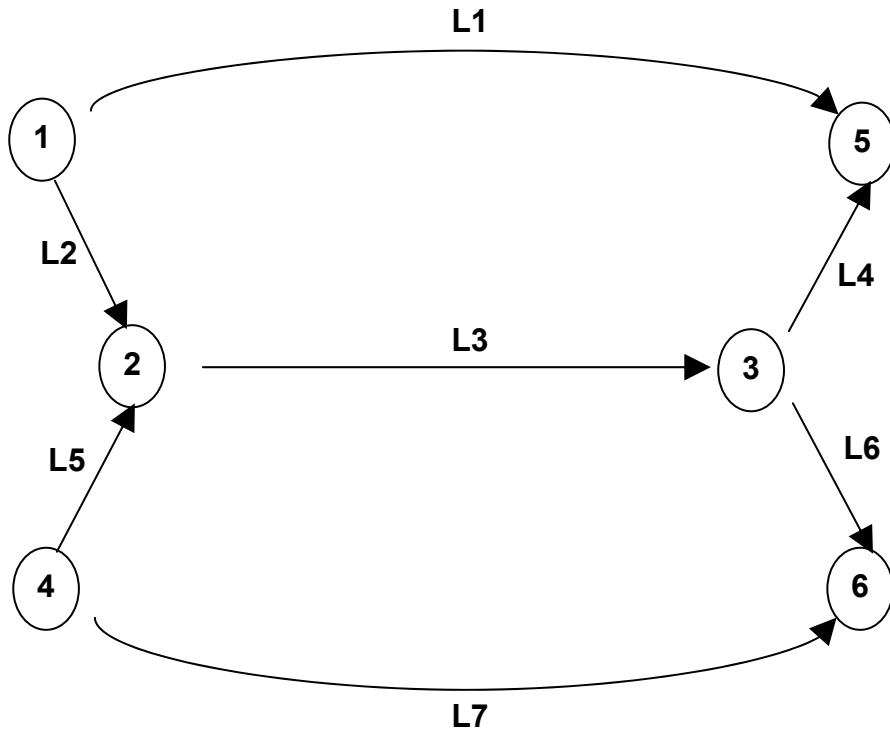


Figure 5.1 'Little' Network

Table 5.3 'Little' Link Parameters and Results ( $\theta = 1.0$ ,  $V/C = 1.0$ )

Link (k)	Free-flow travel time ( $t_0$ )	Capacity ( $z_k$ )	Capacity expansion ( $y_k$ )	Desired V/C ratio	Calculated V/C ( $\theta = 1.0$ , Davis 1994)
1	1.0	200	14.35	1.0	1.95
2	0.25	300	385.65	1.0	1.58
3	0.25	700	728.42	1.0	2.15
4	0.25	300	385.65	1.0	1.77
5	0.25	300	442.78	1.0	1.76
6	0.25	300	442.78	1.0	1.95
7	1.0	200	57.22	1.0	2.15

It is contended that the adoption of a level of service constraint provides a more useful solution than merely minimizing the costs for travel and construction. The final column in Table 5.3 shows the volume to capacity ratios for Davis' (1994) 'Little' solution. The optimal set of capacity expansions under the traditional NDP formulation given in Davis (1994) all have volume to capacity ratios of greater than 1.0, which means that the demand for the link is greater than its capacity. This is an example of a highway expansion program that will be seen by the public as obsolete immediately after its completion. Although it may be 'optimal' to minimize construction and system costs, it might not be a politically feasible solution. Political influence and local feelings usually push for highway agencies to expand road capacity to reduce congestion--the level of service constraint guarantees those mobility levels.

#### *'Sioux Falls' Network*

Traditionally the 'Sioux Falls' network, as shown in Figure 5.2, is used as the 'realistically' sized network to test algorithms. The link characteristics and demands for this network are from Suwansirikul et al. (1987). Note that for this network, the characteristics are identical for the directional link going from node A to node B and in the reverse direction. In planning networks representing realistic complexity, this is not always the case.

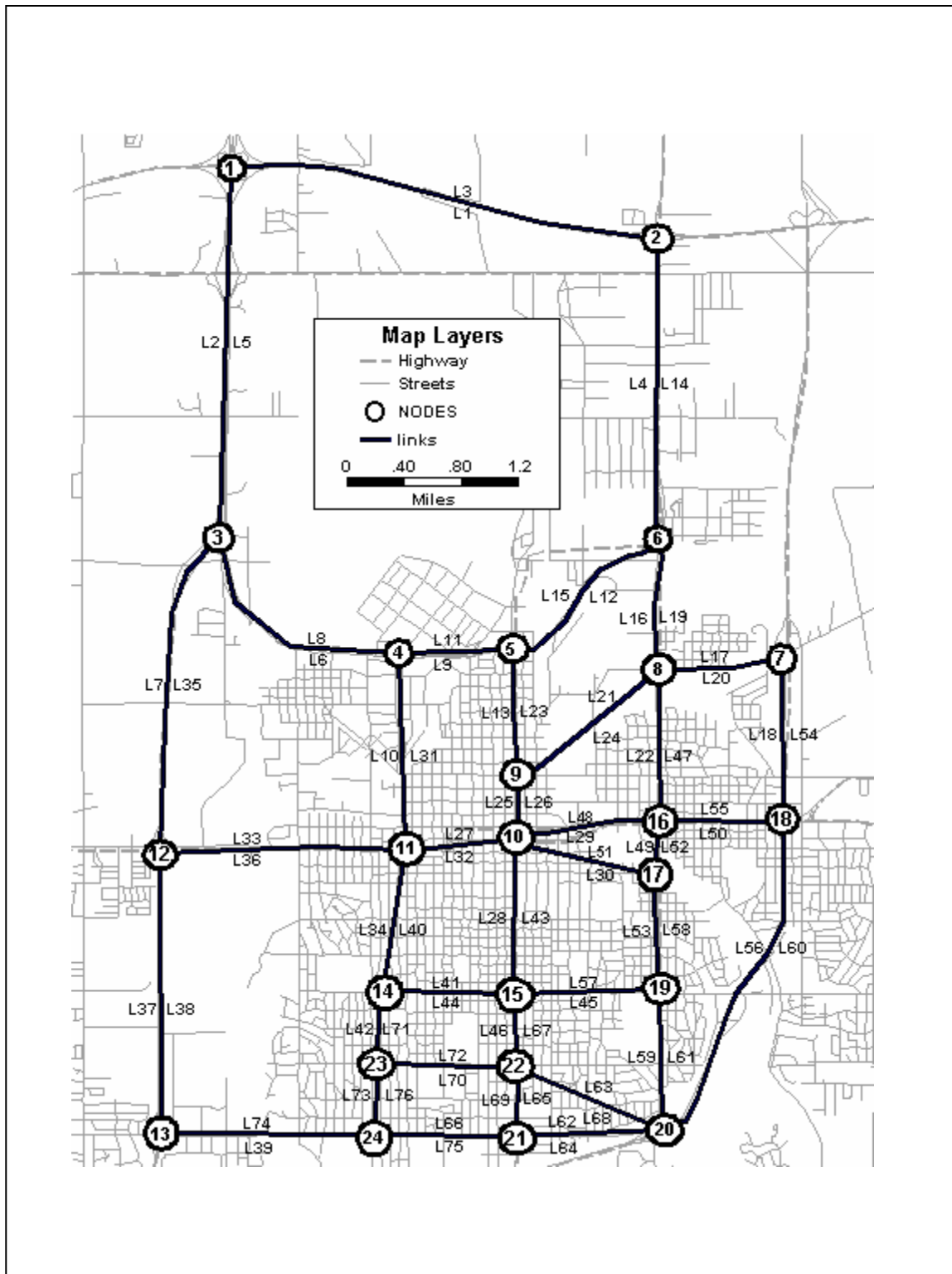


Figure 5.2 'Sioux Falls' Network



**Table 5.4** 'Sioux Falls' Results ( $\theta=10$ ,  $V/C = 1.0$ )

<b>Candidate Links</b>	<b>Capacity Expansions</b>	<b>Desired V/C ratio</b>	<b>Calculated V/C (<math>\theta=10</math>, Davis 1994)</b>
16	13.94	1.0	3.70
19	12.261	1.0	3.50
17	2.3895	1.0	1.44
20	5.8215	1.0	1.76
25	8.669	1.0	1.62
26	10.105	1.0	1.73
29	17.785	1.0	3.57
48	17.335	1.0	3.55
39	11.29	1.0	2.23
74	18.095	1.0	2.44

Table 5.4 shows the results found by implementing the modified MSA procedure selected and implemented for the 'Sioux Falls' network. These results are compared to the level of service associated with Davis' (1994) solution to the traditional NDP. As for the 'Little' network, this shows LOS much worse than would be acceptable for most forecast planning purposes.

*'Waseca' Network*

The more complex 'Waseca' network is shown in Figure 5.3. With 184 links, the 'Waseca' network is significantly larger than the 76 link 'Sioux Falls' example. The Waseca network has been used as a teaching example network for NDP heuristic methods in Transportation Engineering classes at the University of Minnesota. A fixed O-D table was calculated, with contrived socioeconomic data to predict trip generation,

which was then distributed, using a gravity model. Apart from the link lengths, other associated data, such as free-flow speed, was assumed. See Appendix C for the network data used. The dispersion parameter  $\theta$  is set at 0.2, following Leurent's (1995) work on case studies in the Paris metropolitan area. This means that if one route is shorter by five minutes than a second, then approximately three out of four drivers will choose the first road.

Most of the computational experiments were carried out on this network. As discussed in Chapter Four, the procedure employing the MINOS software successfully solved the 'Waseca' NDP, but not the Twin Cities problem. Two results are presented for this network, in Table 5.5, one is the solution calculated using MINOS' optimization routine and the other is the solution found using the modified MSA procedure. The specified LOS was 1.0.

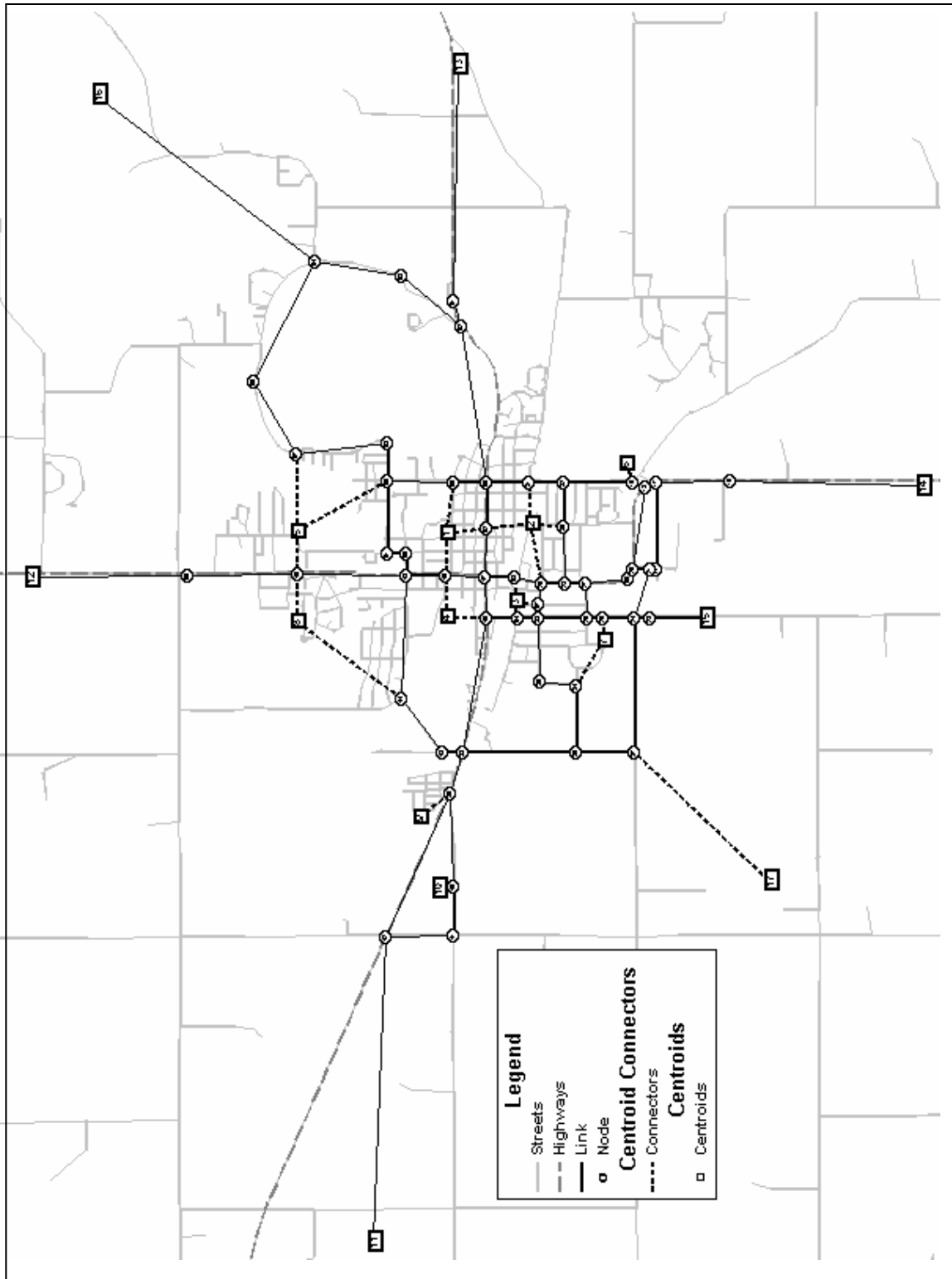


Figure 5.3 'Waseca' Network

**Table 5.5** 'Waseca' Capacity Expansion Results ( $\theta = 0.2$ ,  $V/C = 1.0$ )

<b>Link</b>	<b>Expansion (NLP with MINOS)</b>	<b>Expansion (Davis' modified MSA procedure)</b>	<b>Link</b>	<b>Expansion (NLP with MINOS)</b>	<b>Expansion (Davis' modified MSA procedure)</b>
23	183	183	124	172	172
24	437	437	125	469	469
57	183	183	126	515	515
69	41	41	129	496	496
93	243	243	130	783	783
96	825	825	134	782	782
98	303	303	135	527	527
100	1458	1458	142	327	327
101	88	88	143	317	317
104	545	545	165	26	26
117	225	225	167	302	302
118	334	334	170	348	348
122	491	491	178	527	527
123	398	398	179	437	437

*'Twin Cities' Network*

The detailed planning network used for the Twin Cities was supplied by the Metropolitan Council. While the network was originally developed in 1990, it has been subsequently updated and corrected. The forecasted travel demand was also supplied by the Metropolitan Council, in the form of the 2020 vehicle demand O-D matrices for both AM and PM peak hours. Additional data was available from the same source that forecast 2020 vehicle and transit-person trips. The research question was to find a lower bound on the extent to which physical expansion of highway capacity could be used to

accommodate future growth. So using vehicle-trips was a more conservative estimate of 2020 forecast trips than taking the combined forecast vehicle and transit person trips.

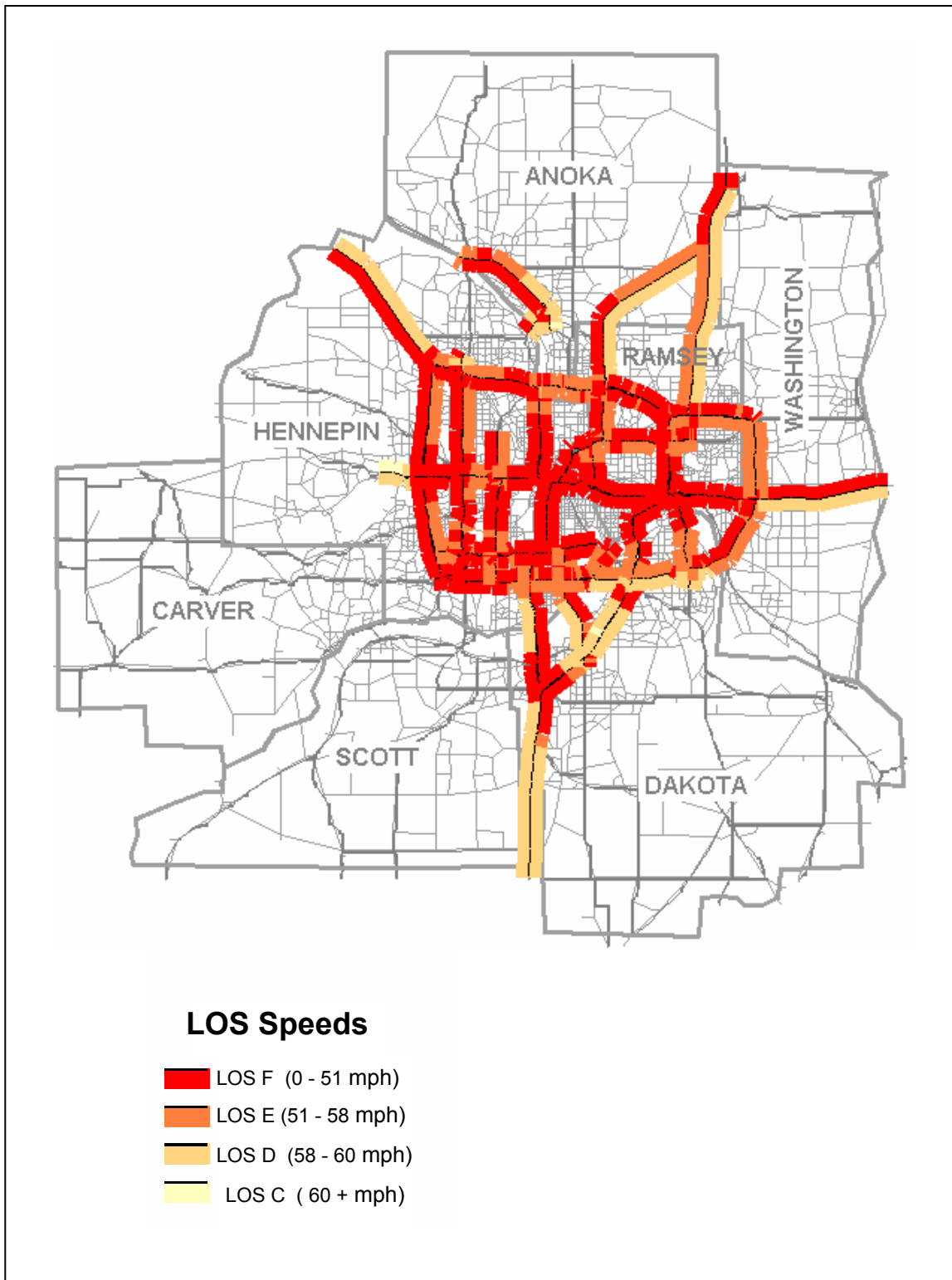
The 'candidate' set of roads considered for expansion in the Twin Cities network was confined to the limited access roadways. This means that while traffic is assigned to all roads, only the limited access roads were constrained to provide the designated level of service and considered for expansion. The limited access roads were adopted as the candidate set because this project's client was Mn/DOT's Traffic Management Center (TMC). Other assumptions made in the model are that the traveling public's desired speed on the limited access roads is about 60 mph. Again the dispersion parameter  $\theta$  was taken as 0.2, as in Leurent (1995). The LOS constraint was specified at 'C', and the maximum volume to capacity ratio for this LOS is 0.63 (Roess et al 1998). This level of service was chosen as it is still possible to travel at about the free-flow speed. At LOS C, "speeds are at or near free-flow speed, but freedom to maneuver is noticeably restricted. Lane changes require more care and vigilance by the driver...When minor incidents occur, local deterioration in service will be substantial. Queues may be expected to form behind any significant blockage" (Garber and Hoel 1999).

Before presenting the solution to the research question, 'what would it take to build our way out of congestion', the 'do-nothing' alternative is shown, that is the congested system speeds when there is no highway capacity expansion. With the fixed O-D matrix, a stochastic assignment was carried out until user equilibrium was reached. Using the Highway Capacity Manual (HCM) flow-speed curves (Roess et al 1998), the level of service was then determined for each link. To do this, the equilibrium link flow

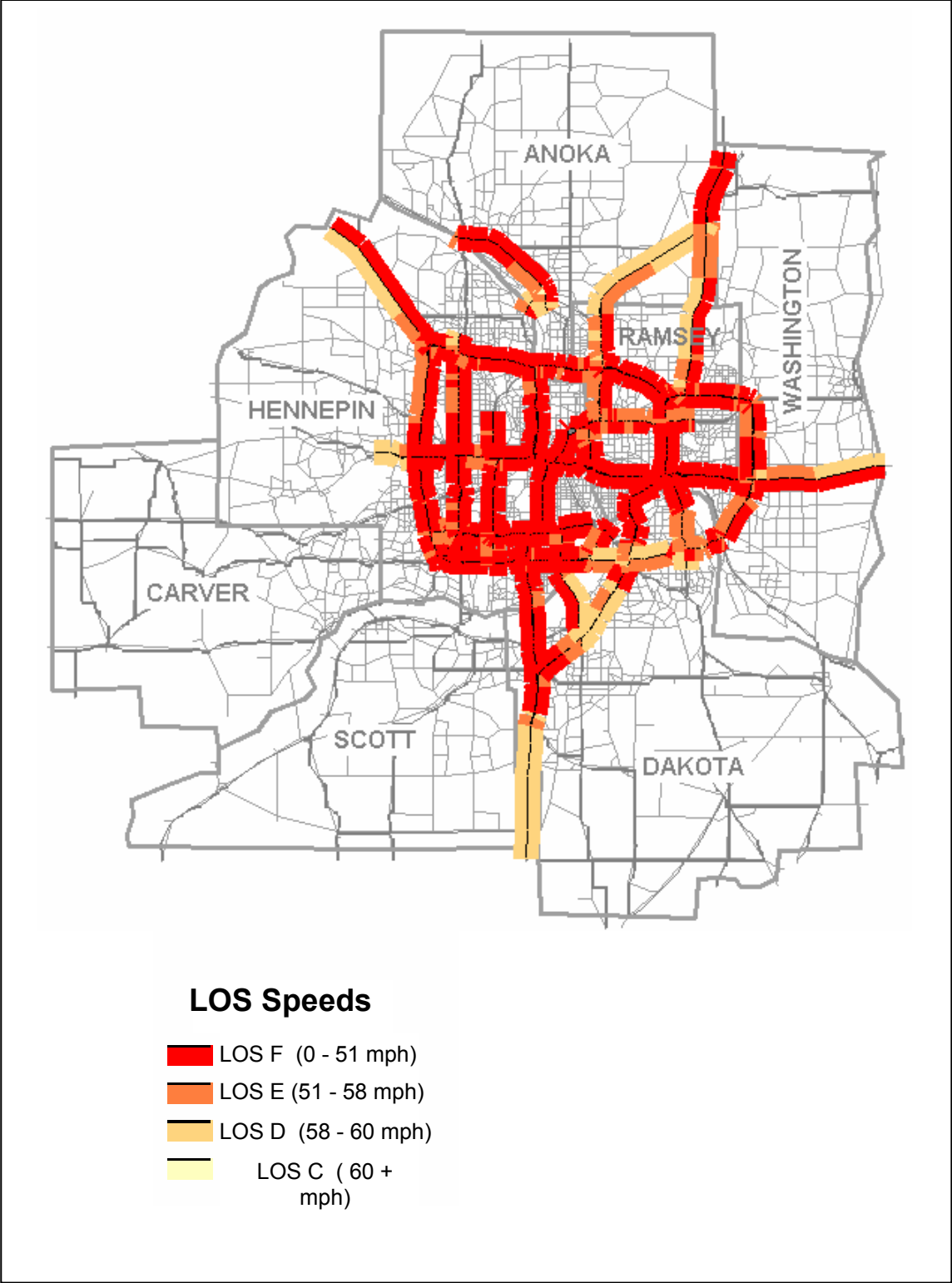
is determined for each link, denoted by  $x_k$ , and the capacity for that link under the do-nothing alternative is given by  $z_k$ . The volume to capacity ratio for each link is then calculated by dividing the flow,  $x_k$ , by the capacity,  $z_k$ . The LOS for this candidate set of roads is shown in Figure 5.4, the AM peak hour, and Figure 5.5, the PM peak hour, for the Twin Cities in the forecast year 2020. These figures indicate that significant portions of the system will be operating at worse than LOS C. Table 5.6 shows the number of kilometers operating at each LOS. Significant points to note from this table are that during the PM peak hour, approximately half the system operates at LOS C or worse, and during the AM peak hour more than a third operates at LOS C or worse.

**Table 5.6** Levels of Service Under 'Do-Nothing' Alternative (limited access roads)

<b>LOS</b>	<b>AM km</b>	<b>%</b>	<b>AM lane- km</b>	<b>%</b>	<b>PM km</b>	<b>%</b>	<b>PM lane- km</b>	<b>%</b>
<b>A</b>	301	29	780	30	82	8	216	8
<b>B</b>	217	21	536	20.7	196	19	512	20
<b>C</b>	159	15	406	15.7	171	16.6	424	16
<b>D</b>	130	13	328	12.7	183	17.8	488	19
<b>E-F</b>	227	22	538	20.8	401	38.8	948	37
<i>Total</i>	<i>1034</i>	<i>100</i>	<i>2588</i>	<i>100</i>	<i>1033</i>	<i>100</i>	<i>2588</i>	<i>100</i>



**Figure 5.4** Year 2020 'Do-nothing' Twin Cities AM Peak Hour Speeds

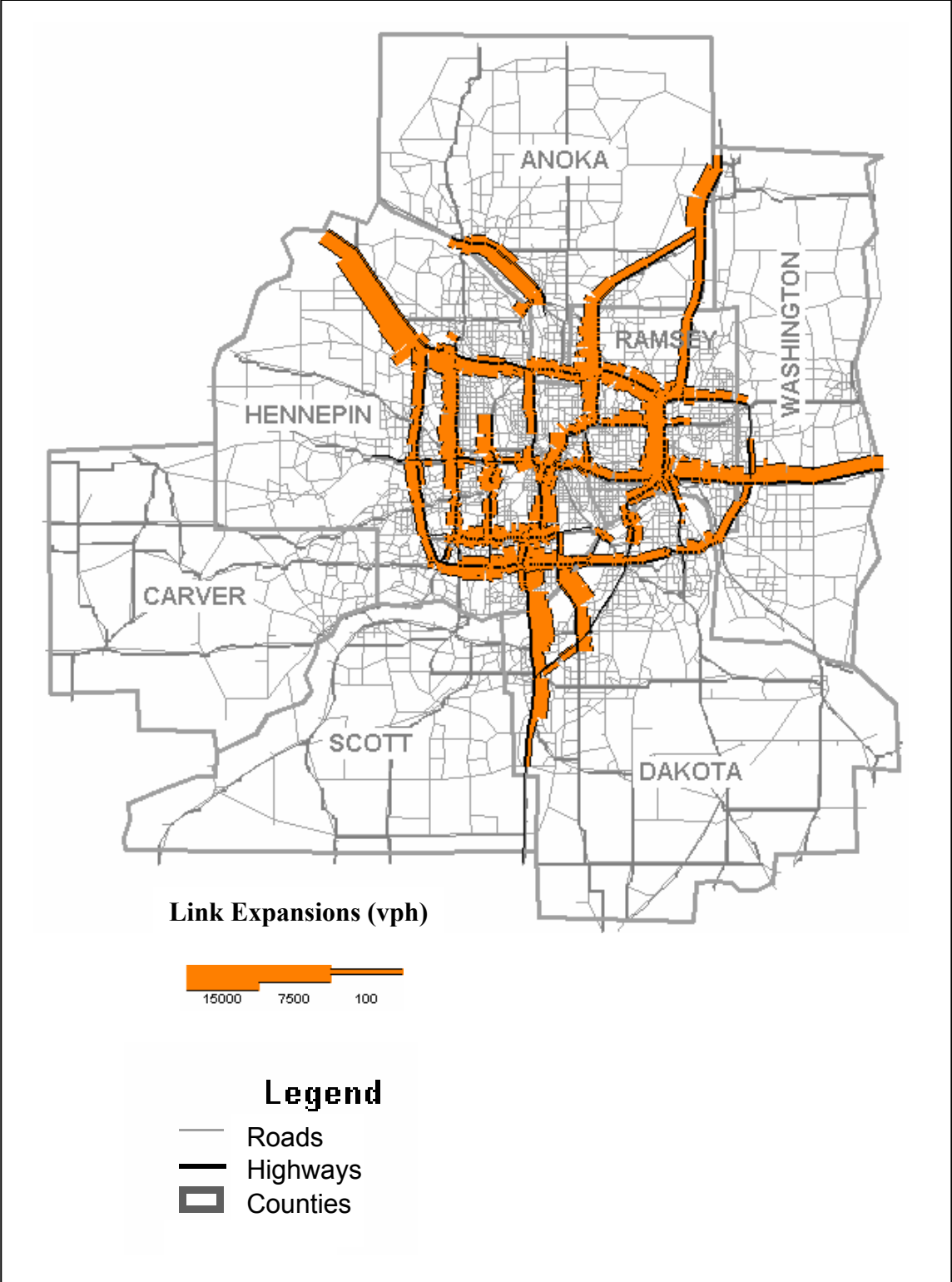


**Figure 5.5** Year 2020 'Do-nothing' Twin Cities PM Peak Hour Speeds

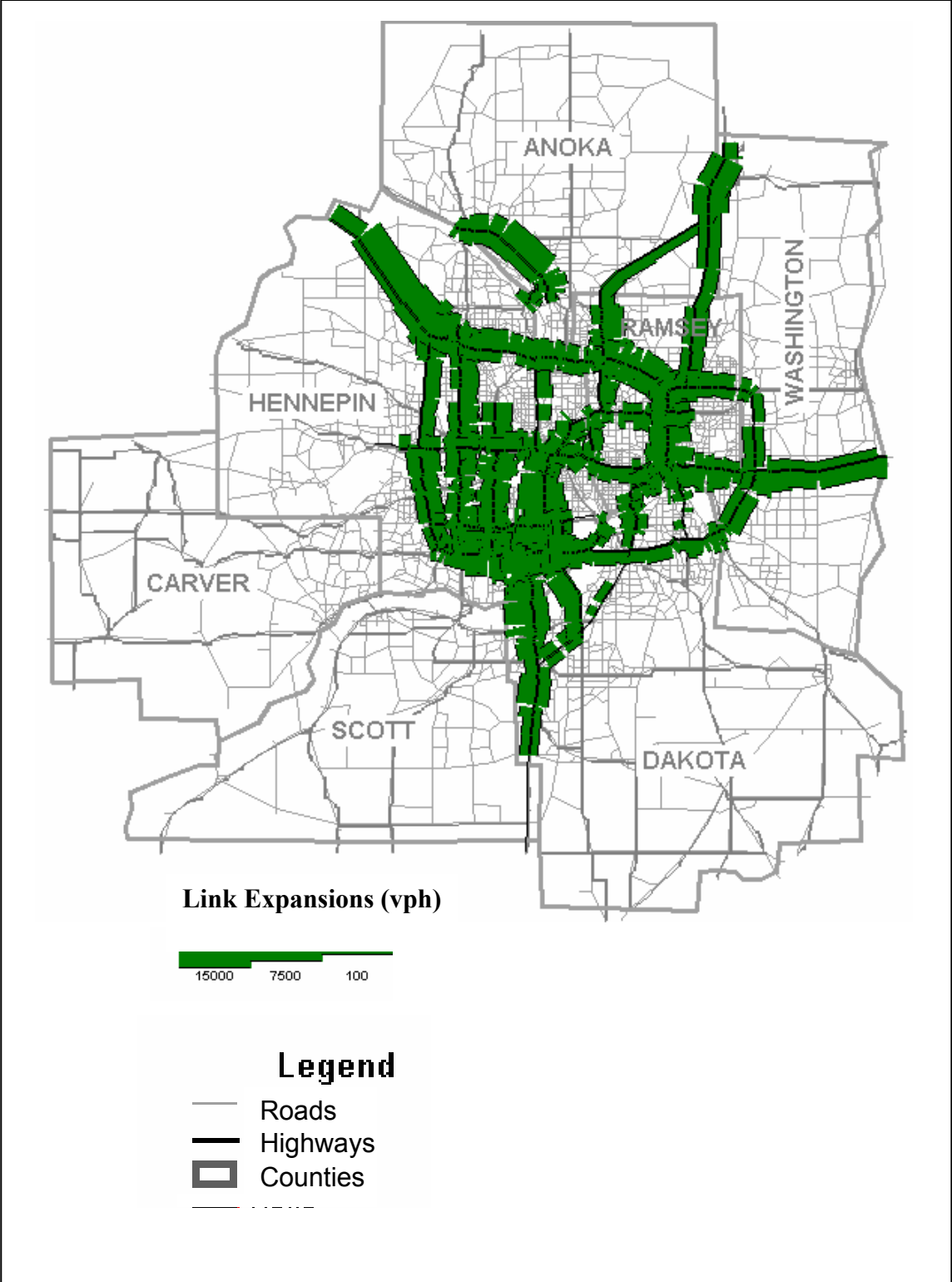


Using the procedure outlined in Chapter Four, the minimal set of capacity expansions that satisfied the constraints were determined. To support that this is an optimum, the procedure was restarted from a number of different initial conditions. This is a commonly used technique in optimization and was been used in other NDP literature, for example Friesz's (1985), who defined stability as: "A network equilibrium will be stable if convergence of the flow pattern to equilibrium from arbitrary initial conditions is guaranteed". The initial start point used adopted zero links flows and expansions. Taking the link flows and doubling the link expansions from the first result was another start point. Finally, link expansions between 0 to 1000 were randomly generated, then holding these constant the equilibrium link flows were found and these constituted the next starting point.

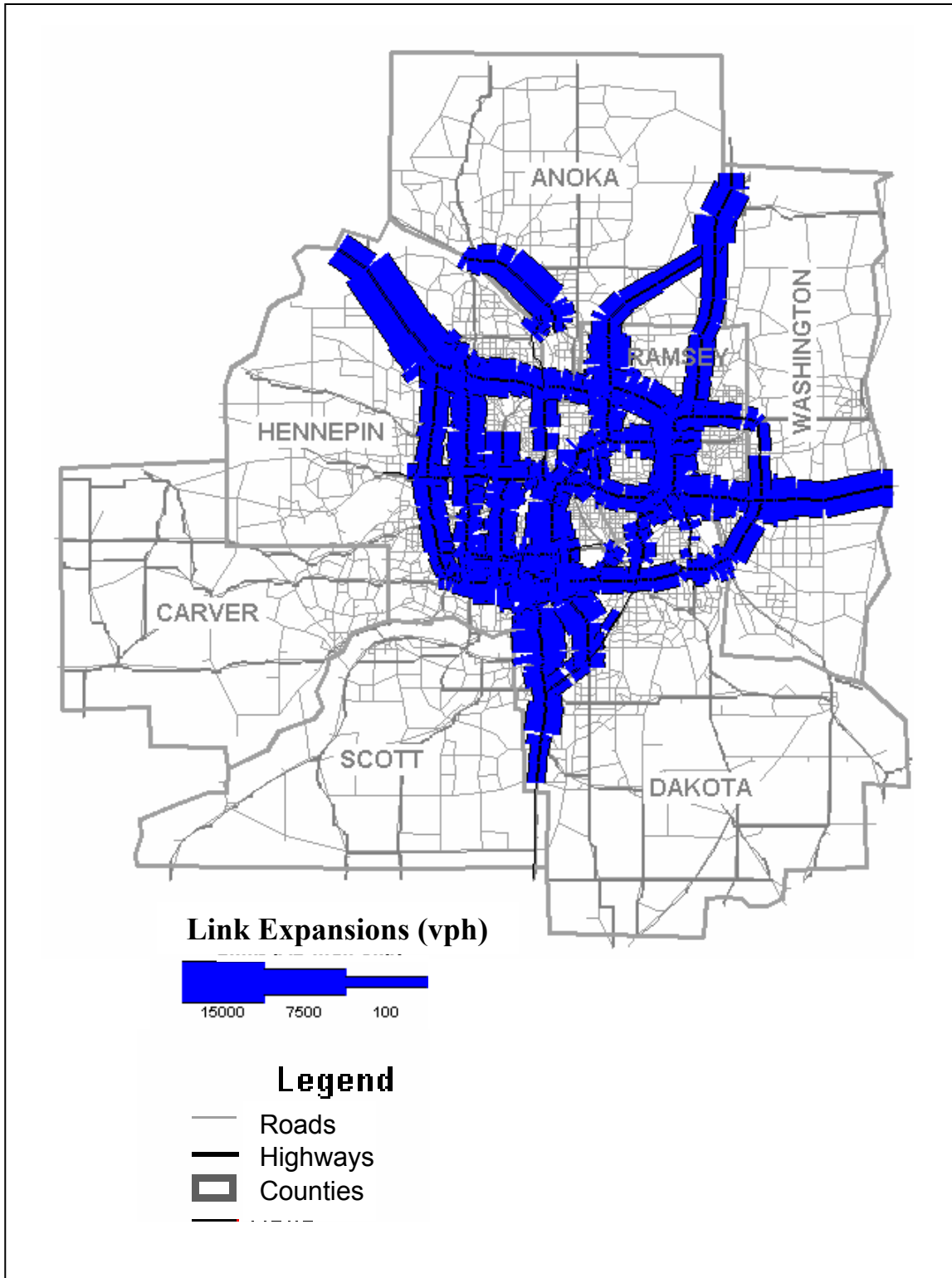
The capacity expansion results were found for both the AM and PM peak hours, as shown in Figures 5.6 and 5.7 respectively. These figures display the required expansion for the Twin Cities freeway system as a scaled theme map, the thicker lines indicate higher amounts of capacity expansion. Expansion shown is directional, so that the centerline of a freeway is a thin black line, with the directional expansion displayed on either side. The interesting points to note are the directionality of the expansion. Figure 5.6 shows the morning peak hour and therefore much of the expansion is in the direction moving towards the Twin Cities, Figure 5.7 for the PM peak shows the reverse case.



**Figure 5.6** Year 2020 Twin Cities AM Capacity Expansion ( $\theta = 0.2$ ,  $V/C = 0.63$ )



**Figure 5.7** Year 2020 Twin Cities PM Capacity Expansion ( $\theta = 0.2$ ,  $V/C = 0.63$ )



**Figure 5.8** Year 2020 Twin Cities AM and PM Capacity Expansion

The calculated set of capacity expansions is a continuous variable in units of vehicles per hour. For the candidate set, the expansions were then converted into equivalent numbers of additional lanes, using the existing capacities and number of lanes. Table 5.7 shows the amount of expansion required in lane-mileage and the number of kilometers in the system that would be affected by expansion. The four rows shown in this table indicate the range of lane values depending on methods used to convert the continuous expansion variable into lanes. This is done based on the capacity of the existing lanes, so it is possible that a resulting calculation could be, for example, one and a half lanes. The first row shows calculated values, which is the direct conversion, the second row is rounded to the nearest integer, the third row is chopped to the integer below and the fourth row is increased to the next largest integer. The table shows first the results calculated using the AM peak hour demand, then the PM peak hour demand and finally the maximum capacity expansion calculated for each link forms the solution which satisfies both AM and PM demands.

**Table 5.7** Twin Cities Capacity Expansion

	<b>AM Lane km</b>	<b>AM km</b>	<b>PM Lane km</b>	<b>PM km</b>	<b>AM/PM Lane km</b>	<b>AM/PM km</b>
<b>Calculated</b>	819	518	1387	761	1844	879
<b>Round</b>	811	-	1390	-	1852	
<b>Floor</b>	591	-	1025	-	1427	-
<b>Ceiling</b>	1107	-	1786	-	2306	-

To gain an appreciation for the numbers of lanes required, these calculations were grouped into the number of lane expansion amounts, along with the sum of the associated mileage of each of these groups, as shown in Table 5.8. The network shows a substantial amount of kilometers of lane expansion. In total 879 kilometers (546 miles) of the freeway system, out of 1,033 kilometers (642 miles), would be subject to expansion. An additional 1,844 lane-kilometers (1,146 lane-miles) would be added to the 2,588 lane-kilometers (1,608 lane-miles) of limited access roads in the base network.

**Table 5.8** Twin Cities Lane Groups of Capacity Expansion.

<b>Lane Groups</b>	<b>AM km</b>	<b>PM km</b>	<b>AM/PM km</b>
> 0 - 1	201	252	199
> 1 - 2	147	227	265
> 2 - 3	106	127	204.5
> 3 - 4	32	92	117
> 4 - 6	29	59	87.5
> 6 - 8	3	3	5
> 8 - 10	0	1	1
> 10	0	0	0
<i>Total</i>	<i>518</i>	<i>761</i>	<i>879</i>

## **CHAPTER SIX**

### **CONCLUSIONS**

The impetus behind this project was to provide a numerical answer to the question--'what would it take to build our way out of congestion?'. This research question is important because answering it requires the selection and implementation of a procedure to solve a large-scale NDP. Solving large-scale, realistically sized, NDP's had remained an unsolved challenge due to the computational complexities involved. Additionally the solution to the Twin Cities NDP that identifies the minimal set of capacity expansions to accommodate future travel at acceptable mobility levels would provide analytical data to guide the debate concerning the provision of highway capacity.

The identification of a procedure capable of solving a large scale NDP was described in Chapter Four. This novel procedure re-poses the NDP as one of minimizing total lane-miles of capacity expansion, subject to LOS and equilibrium constraints. It then uses a modification of the MSA algorithm to solve the NDP. Although the convergence of this method may be slow, it has several advantages. Firstly, given certain conditions, it is guaranteed to converge and secondly it is a relatively simple algorithm to program compared to many other minimization algorithms. Although the MSA is not considered to be numerically efficient (Ortuzar and Willumsen 1994), it is the only procedure that has been successfully implemented to solve a problem of this size--the

speed of an algorithm is a relative measure, so since there is currently no viable competitor, this is perhaps an unfair criticism.

The modified NDP procedure was then implemented to find a solution to the Twin Cities NDP, the results of which are presented in Chapter Five. As previously discussed, the largest 'realistic' NDP solved and presented in the literature to date has been the 'Sioux Falls' planning network. At only 76 links, this is far smaller than the Twin Cities network, which has 20,380 links, and is at the very lower limit of size in terms of practical application. In the work presented here, solutions were found for AM and PM peak hours for the Twin Cities forecast 2020 travel demands. The sum of the maximums of all the links for both peak periods provides an indication of the expansions required to 'build our way out of congestion', and this amounted to an additional 1,844 lane-kilometers (1,146 lane-miles) of capacity expansion. Currently the limited access highway system in the Twin Cities is 2,588 lane-kilometer (1,608 lane-miles), so this expansion represents approximately a seventy percent increase.

The numerical solution to this problem isn't expected to quell debate about the relative merits of highway capacity expansions, but provide additional evidence to guide debate towards realistic expectations regarding the provision of highway capacity. The techniques developed here can also be applied to similar large problems, and so used to investigate and compare more realistic solutions to the increased travel demands.

In summary, the contributions of new knowledge that this research project makes are:



1. Selection and demonstrated successful implementation of a procedure to solve to large-scale NDP--largest known.
2. Numerical answer to what it would take to build our way out of congestion, thereby adding to this debate.

### **Future Research**

Two important sub-models were employed in the implementation of this procedure, the shortest-path and assignment routines. This research implemented some of the most commonly used routines, the Pape-Moore shortest-path and the Dial Assignment. Alternative routines could be implemented to compare solution times with this method, such as Bell's (1995) matrix-inversion assignment and Zhan and Noon's (1998) work on shortest-path routines for real road networks.

This research selected as its candidate set for expansion the limited access highway system of the Twin Cities. Investigation of alternative candidate sets would be of interest, perhaps next considering arterials in addition. Similarly this research solved this problem for level of service 'C', which was argued to be a conservative threshold congestion. Solution for different levels of service and candidate sets would help test the sensitivity of these solutions to the various scenarios.

Congestion is a major area of concern for many metropolitan areas, and therefore using this procedure to solve NDP's for other cities might help guide their debates.

The solution procedure selected and implemented is a new algorithm and its proof of necessity, in that the set of solutions found by the algorithm contains the set of

Kuhn-Tucker points, is given in Appendix A. However, this is a somewhat weak result, and while numerical tests support that this procedure gives good results a stronger proof would be preferred.

Network Design Problems have been recognized by many scholars as inherently complex and difficult problems to solve. The development of an efficient algorithm capable of handling realistic networks has remained a significant and important challenge for researchers. Under a very specific NDP formulation, this research project has demonstrated the solution to a large-scale NDP. This large-scale NDP solution also represents a significant contribution to the regional Twin Cities debate concerning the capabilities and limitations of 'building our way out of congestion'.



## REFERENCES

Abdulaal, M. and L. J. LeBlanc (1979) Continuous Equilibrium Network Design Models. *Transportation Research*. 13B, pp. 19-32.

Bell, M.G.H, (1995) Alternatives to Dial's Logit Assignment. *Transportation Research*. 29B, pp. 287-295.

Bell, M.G.H, and Y. Iida (1997) *Transportation Network Analysis*, Wiley-Interscience: Chichester, UK.

Ben-Akiva, M. and S.R. Lerman (1985) *Discrete Choice Analysis: Theory and Application to Travel Demand*. Cambridge, MA and London, UK: The MIT Press.

Davis, G. A. (1994) Exact Local Solution of the Continuous Network Design Problem via Stochastic User Equilibrium Assignment. *Transportation Research*, 28B, pp 61-75.

Friesz, T. L. (1985) Transportation Network Equilibrium, Design and Aggregation: Key Developments and Research Opportunities. *Transportation Research*, 19A, pp. 413-427.

Friesz, T. L., H-J. Cho, N. J. Mehta, R. L. Tobin, and G. Anandalingam (1992) A Simulated Annealing Approach to the Network Design Problem with Variational Inequality Constraints. *Transportation Science*, vol. 26, no.1, pp. 18-26.

Garber, N. J. and L.A. Hoel (1999) *Traffic and Highway Engineering*. Brooks/Cole Publishing Company, Pacific Grove, CA.

Huang, H-J and M. G. H. Bell (1999) *Continuous Equilibrium Network Design Problem With Elastic Demand: Derivative-Free Solution Methods*. *Transportation Networks: Recent Methodological Advances*. Selected Proceedings of the 4<sup>th</sup> EURO Transportation Meeting, pp. 175-193.

LeBlanc, L. J. (1975) An Algorithm for the Discrete Network Design Problem. *Transportation Science*, 9(3), pp.183-199.

Leurent, F. M. (1995) Contributions to Logit Assignment Model. *Transportation Research Record*. 1493, pp. 207-212.

Metro State University (2000) *Civic Confidence Survey Executive Summary*. Metro State University, Minnesota.

MetCouncil (2000) *Executive Summary of the Transportation Policy Plan*. Metropolitan Council, Minnesota.

MnDOT (2000) *Draft Transportation System Plan*. Minnesota Department of Transportation Metro Division, Roseville, Minnesota.

Moore, T. and P, Thorsnes (1994) *The Transportation / Land Use Connection*. APA, Chicago Il.

Murtagh, M. and M. Saunders (1998). *MINOS USERS GUIDE*. Systems Optimization Laboratory, Department of Operations Research, Stanford University.

Netlib (2000) Code for Moore-Pape algorithm obtained July 2000, from [www.netlib.org/toms/562](http://www.netlib.org/toms/562).

Oppenheim, N. (1995) *Urban Travel Demand Modeling*. Wiley-Interscience, NY.

Ortuzar, J. de D., and L.G. Willumsen (1994) *Modelling Transport*. Chichester, UK: John Wiley & Sons Ltd.

Roess, R.G., W.R. McShane and E.S. Prassas (1998) *Traffic Engineering*. New Jersey, Prentice Hall.

Sheffi Y. (1985) *Urban Transportation Networks*. Prentice Hall, Englewood Cliffs, NJ.

Small, K.A., C.A. Winston and C. A. Evans. (1989) *Road Works*. Brookings Institution, Washington, D.C.

Steenbrink, P.A (1974) *Optimization of Transport Networks*, Wiley, New York.

Suwansirikul, C., T. L. Friesz, and R. L. Tobin (1987) Equilibrium decomposed optimization: A heuristic for the continuous equilibrium network design problem. *Transportation Science*, 21, pp. 254-263

Texas Transportation Institute (2001) *Urban Mobility Study*. Information & Technology Exchange Center/Publications, Texas Transportation Institute.

Van Vliet, D. (1981) Selected node-pair analysis in Dial's assignment algorithm. *Transportation Research*, 15B, pp. 65-68

Yang, H. and Bell, M. G. H. (1998) Models and algorithms for road network design: a review and some new developments. *Transport Reviews*, vol. 18, no.3, pp. 257-278

Yang, H. and Bell, M. G. H. (2001) Transport bilevel programming problems: recent methodological advances.. *Transport Research*, 35B, pp.1-4.

Zhan, F. B. and C. E. Noon (1998) Shortest Path Algorithms: An Evaluation using Real Road Networks. *Transport Science*, vol. 32, no.1, pp.65-73.

**APPENDIX A**  
**NDP MODIFIED MSA PROOF**



**NDP\*:**

minimize

$$\bar{\mathbf{I}}^T \vec{y} = \sum_{i=1}^m l_i y_i$$

subject to

$$-\tilde{\mathbf{C}}\vec{y} + \vec{x}(\vec{y}) - \tilde{\mathbf{C}}\vec{z} \leq \vec{0} \quad \text{i.e. } \frac{x_i(\vec{y})}{z_i + y_i} \leq \tilde{c}_i \quad i = 1, \dots, m$$

$$-\vec{y} \leq \vec{0} \quad \text{i.e. } y_i \geq 0 \quad i = 1, \dots, m$$

where

$$\tilde{\mathbf{C}} = \text{diag}\{\tilde{c}_i\} \text{ and the candidate set is indexed } i=1, \dots, m$$

and the functions  $x_k(\vec{y})$  are defined implicitly

$$x_k - \sum_j d_j q_k^j(\vec{x}, \vec{y}) = 0 \quad k = 1, \dots, n$$

We want to show:

If  $\vec{y}$  is a local solution to NDP\*, then for all  $i = 1, \dots, m$  either

$$y_i = 0 \text{ or } y_i = \frac{x_i(\vec{y})}{\tilde{c}_i} - z_i$$

Suppose  $\vec{y}$  is a local solution to NDP\*. Then  $\vec{y}$  satisfies KKT conditions, and by Theorem 4.2.15 (Nonlinear Programming, Bazaraa, Sherali and Shetty 1993, p. 153), this is equivalent to  $\vec{y}$  being a solution to the approximating linear program:

minimize

$$\bar{\mathbf{I}}^T \vec{y} + \bar{\mathbf{I}}^T (\vec{y} - \vec{y})$$

subject to

$$\begin{aligned}
& -\tilde{c}\bar{y} + \bar{x}(\bar{y}) - \tilde{c}\bar{z} + [J_x(\bar{y}) - \tilde{c}][\bar{y} - \bar{y}] \leq \bar{0} \quad \text{where } \left( J_x(\bar{y}) = \begin{bmatrix} \frac{\partial x_k}{\partial y_i} \end{bmatrix}_{\bar{y}} \right) \\
& -\bar{I}\bar{y} - I(\bar{y} - \bar{y}) \leq 0
\end{aligned}$$

This can be simplified as

minimize

$$\bar{1}^T \bar{y}$$

subject to

$$\begin{aligned}
[J_x(\bar{y}) - \tilde{C}]\bar{y} &\leq \tilde{C}\bar{y} - \bar{x}(\bar{y}) + \tilde{C}\bar{z} + J_x(\bar{y})\bar{y} - \tilde{C}\bar{y} \\
&\leq \tilde{C}\bar{z} + J_x(\bar{y})\bar{y} - \bar{x}(\bar{y}) = \bar{b} \\
\bar{y} &\geq 0
\end{aligned}$$

Introducing the slack variable  $\bar{w}$ , we can then write this in a "standard" form

**LP\*:**

minimize

$$\bar{1}^T \bar{y}$$

subject to

$$\begin{aligned}
[J_x(\bar{y}) - \tilde{C}]\bar{y} + I\bar{w} &= \tilde{C}\bar{z} + J_x(\bar{y})\bar{y} - \bar{x}(\bar{y}) \\
\bar{y} &\geq 0 \\
\bar{w} &\geq 0
\end{aligned}$$

The extreme points of the polyhedral region:

$$\begin{aligned}
[J_x(\bar{y}) - \tilde{C}:I] \begin{bmatrix} \bar{y} \\ \bar{w} \end{bmatrix} &= \bar{b} \\
\bar{y} &\geq 0 \\
\bar{w} &\geq 0
\end{aligned}$$

Can be expressed, by re-arranging the order of the elements of  $\bar{y}$  and  $\bar{w}$  in a form

$$\begin{bmatrix} \bar{y}_1 \\ \bar{0} \\ \bar{0} \\ \bar{w}_1 \end{bmatrix} \text{ with } \begin{array}{l} \bar{y}_1 \geq 0 \\ \bar{w}_1 \geq 0 \end{array}$$

In particular, since  $\bar{y}$  solves LP there is a corresponding extreme point:

$$\begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \\ \bar{w}_1 \\ \bar{w}_2 \end{bmatrix} \text{ with } \bar{y}_2 = 0 \text{ and } \bar{w}_1 = 0$$

which satisfies  $\bar{y}_1 \geq 0$ ,  $\bar{w}_1 \geq 0$  and

$$\begin{bmatrix} J_x^1(\bar{y}) - \tilde{C}_1 & J_x^2(\bar{y}) & | & I & 0 \\ J_x^3(\bar{y}) & J_x^4(\bar{y}) - \tilde{C}_2 & | & 0 & I \end{bmatrix} \begin{bmatrix} \bar{y}_1 \\ 0 \\ 0 \\ \bar{w}_2 \end{bmatrix} \\ = \begin{bmatrix} \tilde{C}_1 & 0 \\ 0 & \tilde{C}_2 \end{bmatrix} \begin{bmatrix} z_1 \\ z_2 \end{bmatrix} + \begin{bmatrix} J_x^1(\bar{y}) & J_x^2(\bar{y}) \\ J_x^3(\bar{y}) & J_x^4(\bar{y}) \end{bmatrix} \begin{bmatrix} \bar{y}_1 \\ 0 \end{bmatrix} - \begin{bmatrix} x_1(\bar{y}) \\ x_2(\bar{y}) \end{bmatrix}$$

writing this out

$$\begin{aligned} (J_x^1(\bar{y}) - \tilde{C}_1)\bar{y}_1 &= \tilde{C}_1 z_1 + J_x^1(\bar{y}_1) - x_1(\bar{y}) \\ J_x^3(\bar{y})\bar{y}_1 + \bar{w}_2 &= \tilde{C}_2 z_2 + J_x^3(\bar{y})\bar{y}_1 - x_2(\bar{y}) \\ \bar{y}_1 &\geq 0 \\ \bar{w}_2 &\geq 0 \end{aligned}$$

which simplifies to

$$\begin{aligned} -\tilde{C}_1 \bar{y}_1 &= \tilde{C}_1 z_1 - x_1(\bar{y}) & \bar{y}_1 &\geq 0 \\ \bar{w}_2 &= \tilde{C}_2 z_2 - x_2(\bar{y}) & \bar{w}_2 &\geq 0 \end{aligned}$$

and the top of these two equation implies

$$\begin{aligned} \bar{y}_1 &= -\tilde{C}_1^{-1}(\tilde{C}_1(z_1) - x_1(\bar{y})) \\ &= \tilde{C}_1^{-1}x_1(\bar{y}) - z_1 \end{aligned}$$

i.e.

$$\bar{y}_i = \frac{x_i(\bar{y})}{\tilde{c}_i} - z_i \geq 0 \quad y_i \in \bar{y}_1$$

and

$$\bar{y}_2 = 0 \quad \Rightarrow \quad \bar{y}_i = 0, y_i \in \bar{y}_2$$

This shows that the set of solutions to

$$x_k = \sum_0 d_j q_k^0(\bar{x}, \bar{y})$$
$$y_i = \max\left(0, \frac{x_k(\bar{y})}{\tilde{c}_k} - z_k\right)$$

contains the set of KKT point, which contains the set of local solutions.



# **APPENDIX B**

## COMPUTER CODE

```
*****  
*****          START OF MAIN PROGRAM          *****  
*****
```

PROGRAM        MAIN

C THIS PROGRAM IMPLEMENTS THE SOLUTION PROCEDURE TO SOLVE A  
C NETWORK DESIGN PROBLEM OF THE TYPE:  
C MIN ( LINK EXPANSIONS )  
C SUBJECT TO:  
C   -LINK FLOWS SATISFYING EQUILIBRIUM AT SOLUTION  
C   -POSITIVE LINK EXPANSIONS  
C   -VOLUME TO CAPACITY RATIO IS LESS THAN/EQUAL TO  
C    SOME SPECIFIED LEVEL OF SERVICE (FROM HCM)  
C THE SOLUTION IS BASED ON DAVIS (2001)MODIFIED  
C METHOD OF SUCCESSIVE AVERAGES PROCEDURE.

IMPLICIT        DOUBLE PRECISION (A-H,O-Z)

C LIMIT IS A PARAMETER USED TO SIZE ARRAYS, IT IS BASED ON  
C THE MAXIMUM EXPECTED NUMBER OF LINKS IN THE NETWORK.

INTEGER LIMIT  
PARAMETER (LIMIT=21000)

C NPOSSLST IS A LIST OF CANDIDATES FOR EXPANSION.

LOGICAL NPOSSLST(LIMIT)

C THE COMMON BLOCKS ARE USED TO PASS VARIABLE VALUES THAT  
C ARE COMMON TO A NUMBER OF SUBROUTINES.

C NNODE IS THE NUMBER OF NODES IN THE NETWORK.  
C NUMORG IS THE NUMBER OF CENTROIDS IN THE NETWORK.  
C NEXT IS THE  
C NPRED  
C NTONODE - A LINK IS IDENTIFIED FROM THE START NODE AND  
C ITS END NODE, THIS IS ITS END NODE.  
C NFMNODE IS THE START NODE OF A LINK.  
C XGUESS IS THE CURRENT LINK FLOW.

COMMON /KSINPUT1/ NNODE, NUMORG,  
& NEXT(7400), NPRED(7400),

```
& NTONODE(20400),NFMNODE(20400),XGUESS(20400),  
& GLINK(20400,2),SLINK(20400,4), CLINK(20400,6),NIPE(7400)
```

```
C D(X,Y) IS THE ORIGIN - DESTINATION DEMAND ARRAY.  
C NLINK IS THE NUMBER OF LINKS IN THE NETWORK.
```

```
COMMON /KSINPUT2/ ABI(7400),  
& D(1200,7400),NLINK, NPOINT(7400)
```

```
C THETA IS A PARAMETER WHICH SCALES THE PERCEIVED TRAVEL TIMES
```

```
COMMON /KSINPUT2A/ THETA  
COMMON /KSINPUT3/ IDERIV, NPOSS, NPOSSLST
```

```
C NUMB IDENTIFIES WHICH NDP IS BEING SOLVED.  
C NAME IS THE NAME OF THE NDP BEING SOLVED.
```

```
COMMON /KSINPUT4/ NUMB, NAME
```

```
C SUBROUTINE MOOREPAPE READS IN DATA, SETS VARIABLES  
C AND SETS UP INITIAL FEASIBLE SOLUTION TO START OPTIMIZER.
```

```
CALL MOOREPAPE
```

```
C SUBROUTINE MSA CARRIES OUT MODIFIED METHOD OF  
C SUCCESSIVE AVERAGES TO FIND EQUILIBRIUM NDP SOLUTION.
```

```
CALL MSA
```

```
* -----
```

```
STOP  
END
```



```
*****
****          START OF MOOREPAPE SUBROUTINE          ****
*****
```

SUBROUTINE MOOREPAPE

C THIS READS FROM ALL THE INPUT FILES, AND SETS UP  
C INITIAL PROBLEM BEFORE IT GETS PASSED TO THE  
C OPTIMIZER.

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

INTEGER LIMIT, LIM3  
PARAMETER (LIMIT=21000,LIM3=8000)

LOGICAL NPOSSLST(LIMIT)  
INTEGER IT(LIM3),  
& LISTY(LIMIT), DEMAND, LKS(10), TYPES

CHARACTER NAME\*20

DOUBLE PRECISION XT(LIMIT), ZERO, DIFF,  
& ERRMAX, ERRCRIT, DA(LIMIT), FIT, LOS

COMMON /KSINPUT1/ NNODE, NUMORG,  
& NEXT(7400), NPRED(7400),  
& NTONODE(20400),NFMNODE(20400),XGUESS(20400),  
& GLINK(20400,2),SLINK(20400,4), CLINK(20400,6),NIPE(7400)

COMMON /KSINPUT2/ ABI(7400),  
& D(1200,7400),NLINK, NPOINT(7400)  
COMMON /KSINPUT2A/ THETA  
COMMON /KSINPUT3/ IDERIV, NPOSS, NPOSSLST  
COMMON /KSINPUT4/ NUMB, NAME

C OD CONTAINS THE ORIGIN-DESTINATION DEMAND MATRIX

OPEN (UNIT=1,FILE='OD',STATUS='OLD')

C NET CONTAINS ALL THE NETWORK PARAMETERS OF THE LINKS

OPEN (UNIT=2,FILE='NET',STATUS='OLD')

C OUT IS OPENED AS AN OUTPUT FILE TO KEEP TRACK OF  
C THE ERROR AT EACH ITERATION.

```
OPEN (UNIT=3,FILE='OUT',STATUS='UNKNOWN')
```

```
THETA = 2.0D-01
```

```
ZERO = 0.0D+0
```

C \*\*\* READ IN NON-ZERO DEMAND VALUES,  
C \*\*\* 'DEMAND' IS THE NO. OF PEOPLE WANTING TO TRAVEL FROM A->B.

```
READ (1,*) DEMAND  
DO 40 I=1,DEMAND  
  READ(1,*) J,K,D(J,K)
```

```
40  CONTINUE
```

C \*\*\* INPUT TITLE OF PROBLEM TO BE SOLVED,  
C \*\*\* THIS SETS WHICH PARAMETERS ARE TO BE READ.

```
READ(2,*) NUMB  
READ(2,*) NAME
```

C \*\*\* INPUT OF THE NUMBER OF NODES NNODE

```
1  READ(2,*) NUMORG,NNODE,NLINK
```

C \*\*\* INPUT OF THE NETWORK

C \*\*\* READ START AND END NODES FOR LINKS &

C \*\*\* READ IN DATA ASSOC W LINK CHARACTERISTICS.

```
C  CLINK-1=FFS TVL TIME (HRS)
```

```
C  CLINK-2=0.15 (BPR COEFF)
```

```
C  CLINK-3=CAPACITY
```

```
C  CLINK-4=BPR TVL TIME
```

```
C  SLINK-1=FFS (MPH)
```

```
C  SLINK-2='FUNCTIONAL CLASS'
```

```
C  SLINK-3=AREA TYPE
```

```
C  SLINK-4=#-LANES
```

```
C  GLINK-1=LINK LENGTH (MI)
```

```
C  GLINK-2=LOS
```

```
NPOSS=0
```

```
READ(2,*) TYPES,(LKS(L),L=1,TYPES)
```

```
READ(2,*) LOS
```

```
DO 4 I=1,NLINK
```

```
C *** FOR PROBLEMS 1 OR 2 DATA INPUT IS SLIGHTLY DIFFERENT
```

```
    IF (NUMB.EQ.1) THEN
      READ(2,*) NFMNODE(I),NTONODE(I),CLINK(I,3),
& CLINK(I,1)
      DO 33 L=1,TYPES
        IF (CLINK(I,1).GE.LKS(L)) THEN
          NPOSS=NPOSS+1
          LISTY(NPOSS)=I
          GLINK(I,2) = LOS
        ENDIF
33    CONTINUE
      ENDIF
      IF (NUMB.EQ.4) THEN
        READ(2,*) NFMNODE(I),NTONODE(I),CLINK(I,3),
& CLINK(I,1), SLINK(I,2)
        DO 34 L=1,TYPES
          IF (SLINK(I,2).EQ.LKS(L)) THEN
            NPOSS=NPOSS+1
            LISTY(NPOSS)=I
            GLINK(I,2) = LOS
          ENDIF
34    CONTINUE
        ENDIF
        IF (NUMB.EQ.2) THEN
          READ(2,*) NFMNODE(I),NTONODE(I),CLINK(I,3),
& GLINK(I,1), SLINK(I,1)
          DO 31 L=1,TYPES
            IF (SLINK(I,1).EQ.LKS(L)) THEN
              NPOSS=NPOSS+1
              LISTY(NPOSS)=I
              GLINK(I,2) = LOS
            ENDIF
31    CONTINUE
          ENDIF
        ENDIF
      ENDIF
```

```
C *** FOR PROBLEMS 3(TC) DATA INPUT IS DIFFERENT
```

```
    IF (NUMB.EQ.3) THEN
      READ(2,*) NFMNODE(I),NTONODE(I),
& SLINK(I,2), GLINK(I,1),SLINK(I,1),
& SLINK(I,3),SLINK(I,4), CLINK(I,3)
```

```

DO 32 L=1, TYPES
  IF (SLINK(I,2).EQ.LKS(L)) THEN
    NPOSS=NPOSS+1
    LISTY(NPOSS)=I
    GLINK(I,2) = LOS
  ENDIF
32  CONTINUE
ENDIF

C *** USE SLINK(K,2) FOR ASS GRP OR AREA TYPE FOR VC LOS.

CLINK(I,2)=1.5D-01
IF (NUMB.EQ.2.OR.NUMB.EQ.3) THEN
  IF (CLINK(I,3).EQ.0.0D+0) CLINK(I,3)=2.4D+03
  IF (SLINK(I,1).EQ.0.0D+0) SLINK(I,1)=1.5D+01

C *** CONVERT TIME TO MINUTES.

  CLINK(I,1)=6.0D+01*(GLINK(I,1)/SLINK(I,1))
ENDIF
4  CONTINUE

C IF LINK CAN BE EXPANDED THEN TRUE.

DO 35 K=1,NPOSS
  NPOSSLST(LISTY(K))=.TRUE.
35  CONTINUE

C *** FILL NPOINT WITH THE START OF NEW NODE NOS.

K=0
J=1
DO 44 I=1,NLINK

45  IF (NFMNODE(I).EQ.J) THEN
    K=K+1
    IT(J)=K
  ELSE
    K=0
    J=J+1
    IT(J)=K
    GOTO 45
  ENDIF
44  CONTINUE

```

```

      J=0
      DO 46 I=1,NNODE
        J=J+IT(I)
        NEXT(I)=J
46    CONTINUE

C ** FILL NPOINT WITH THE START OF NEW NODE NOS.

      NPOINT(1)=1
      DO 47 I=2,NNODE
        NPOINT(I)=NPOINT(I-1)+IT(I-1)
47    CONTINUE

      DO 93 K=1,NLINK
        XGUESS(K) = ZERO
        CLINK(K,4)=CLINK(K,1)
93    CONTINUE

C *** DONE WITH MOOREPAPE SUBROUTINE

      RETURN
      END

```

```

*****
***          START OF SHPHTL SUBROUTINE          ***
*****

```

```

SUBROUTINE SHPHTL(J0)

```

```

C THIS IS A MODIFIED VERSION OF THE PAPE ALGORITHM FROM NETLIB.
C THE TRAVEL TIMES FOR PATHS CONNECTING O-D PAIRS
C NEEDS TO BE DETERMINED TO ASSIGN TRIPS. SEE SHEFFI (P122) FOR
C A FULLER DESCRIPTION OF SHORTEST PATH ALGORITHMS.

```

```

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

```

```

INTEGER LIM3
PARAMETER (LIM3=8000)
INTEGER NJ(LIM3), J0
DOUBLE PRECISION INF,MJI,MJK
DATA INF /9999999/

```

```

COMMON /KSINPUT1/ NNODE, NUMORG,
& NEXT(7400), NPRED(7400),
& NTONODE(20400),NFMNODE(20400),XGUESS(20400),
& GLINK(20400,2),SLINK(20400,4), CLINK(20400,6),NIPE(7400)

```

```

COMMON /KSINPUT2/ ABI(7400),
& D(1200,7400),NLINK, NPOINT(7400)

```

```

C SHPHTL CALCULATES THE SHORTEST PATH LENGTHS (MJ) FROM A
C SPECIFIC NODE (J0) TO ALL OTHER (N-1) NODES IN A NETWORK
C (FLIST,DFLIST,KF).
C PREDECESSOR NODES ARE STORED IN WJ.
C NTONODE : FORWARD INDEX LIST
C CLINK(K,1) : DISTANCE LIST
C NEXT : POINTER LIST FOR FLIST AND DFLIST
C NNODE : NUMBER OF NODES
C CAB : ARRAY OF SHORTEST PATH LENGTHS
C J : INITIAL NODE, FIRST NODE OF SHORTEST PATH
C NPRED : ARRAY OF PREDECESSORS FOR SHORTEST PATH
C CONSTRUCTION
C NJ : DOUBLE ENDED QUEUE FOR NODE DISCUSSION
C INF : A LARGE NUMBER

```

```

        DO 1 I=1,NNODE
          ABI(I)=INF
          NPRED(I)=0
          NJ(I)=0
1      CONTINUE
        ABI(J0)=0.0D+0

C      I  : INDEX FOR NODE DISCUSSION, NODE UNDER DISCUSSION
C      NT  : POINTER TO THE END OF DEQUE NJ
C      MJI : LOCAL VARIABLE OF MJ(I)
C      KFI : LOCAL VARIABLE OF KF(I)
C      KFI1 : LOCAL VARIABLE OF KF(I)+1
C      IR  : INDEX FOR ARRAY DISCUSSION
C      K   : SUCCESSOR OF NODE I
C      MJK : LOCAL VARIABLE OF MJ(K)
C      NJI : LOCAL VARIABLE OF NJ(I), THE NEXT NODE OF NJ TO BE TAKEN
C          UNDER DISCUSSION

        NJ(J0)=INF
        I=J0
        NT=J0

C      OUTER LOOP
C      DISCUSSION OF NODES I

2      KFI=NEXT(I)
        MJI=ABI(I)
        IF (I.EQ.1) THEN
          KFI1=1
        ELSE
          KFI1=NEXT(I-1)+1
        ENDIF

C *** INNER LOOP
C *** DISCUSSION OF SUCCESSORS K

        IF (KFI1.GT.KFI) GO TO 6
        DO 5 IR=KFI1,KFI
          K=NTONODE(IR)
          MJK=MJI+CLINK(IR,1)

C *** NO DECREASE OF SHORTEST DISTANCES

        IF (MJK.GE.ABI(K)) GO TO 5

```

```

C *** DECREASE OF SHORTEST DISTANCES

      ABI(K)=MJK

C *** PREDECESSOR I OF NODE K

      NPRED(K)=I

C *** NODE K ALREADY IN THE DEQUE NJ ?

      IF (NJ(K)) 4,3,5

C *** NODE K ADDED AT THE END OF THE DEQUE NJ

3     NJ(NT)=K
      NT=K
      NJ(K)=INF
      GO TO 5

C *** NODE K ADDED AT THE BEGINNING OF THE DEQUE NJ

4     NJ(K)=NJ(I)
      NJ(I)=K
      IF (NT.EQ.I) NT = K
5     CONTINUE

C *** NODE I TAKEN FROM THE BEGINNING OF THE DEQUE NJ

6     NJI=NJ(I)
      NJ(I)=-NJI
      I=NJI
      IF (I.LT.INF) GOTO 2

      RETURN
      END

```



```

*****
*****          START OF INDEXX SUBROUTINE          *****
*****

```

SUBROUTINE INDEXX

C THIS IS FROM P331 C.U.P. NUMERICAL RECIPES

```

      IMPLICIT DOUBLE PRECISION (A-H,O-Z)
      INTEGER M,NSTACK,LIM3
      PARAMETER (M=7,NSTACK=50, LIM3=8000)

```

C INDEXES AN ARRAY ABI(1:NNODE), I.E., OUTPUTS THE ARRAY  
 C NIPE(1:NNODE) SUCH THAT ABI(NIPE(J)) IS IN ASCENDING ORDER FOR  
 C J = 1; 2; : : : ; NNODE. THE INPUT QUANTITIES NNODE AND ABI  
 C ARE NOT CHANGED.

```

      INTEGER I,INDXT,IR,ITEMP,J,JSTACK,K,L,
      & ISTACK(NSTACK)

```

```

      COMMON /KSINPUT1/ NNODE, NUMORG,
      & NEXT(7400), NPRED(7400),
      & NTONODE(20400),NFMNODE(20400),XGUESS(20400),
      & GLINK(20400,2),SLINK(20400,4), CLINK(20400,6),NIPE(7400)

```

```

      COMMON /KSINPUT2/ ABI(7400),
      & D(1200,7400),NLINK, NPOINT(7400)

```

```

      DO 11 J=1,NNODE
         NIPE(J)=J
11      CONTINUE
         JSTACK=0
         L=1
         IR=NNODE
1      IF(IR-L.LT.M)THEN
         DO 13 J=L+1,IR
            INDXT=NIPE(J)
            A=ABI(INDXT)
            DO 12 I=J-1,L,-1
               IF(ABI(NIPE(I)).LE.A) GOTO 2
               NIPE(I+1)=NIPE(I)
12      CONTINUE
            I=L-1
2      NIPE(I+1)=INDXT

```

```

13  CONTINUE
    IF(JSTACK.EQ.0)RETURN
      IR=ISTACK(JSTACK)
      L=ISTACK(JSTACK-1)
      JSTACK=JSTACK-2
    ELSE
      K=(L+IR)/2
      ITEMP=NIPE(K)
      NIPE(K)=NIPE(L+1)
      NIPE(L+1)=ITEMP
      IF(ABI(NIPE(L)).GT.ABI(NIPE(IR))) THEN
        ITEMP=NIPE(L)
        NIPE(L)=NIPE(IR)
        NIPE(IR)=ITEMP
      ENDIF
      IF(ABI(NIPE(L+1)).GT.ABI(NIPE(IR))) THEN
        ITEMP=NIPE(L+1)
        NIPE(L+1)=NIPE(IR)
        NIPE(IR)=ITEMP
      ENDIF
      IF(ABI(NIPE(L)).GT.ABI(NIPE(L+1))) THEN
        ITEMP=NIPE(L)
        NIPE(L)=NIPE(L+1)
        NIPE(L+1)=ITEMP
      ENDIF
      I=L+1
      J=IR
      INDXT=NIPE(L+1)
      A=ABI(INDXT)
3    CONTINUE
      I=I+1
      IF(ABI(NIPE(I)).LT.A) GOTO 3
4    CONTINUE
      J=J-1
      IF(ABI(NIPE(J)).GT.A) GOTO 4
      IF(J.LT.I) GOTO 5
      ITEMP=NIPE(I)
      NIPE(I)=NIPE(J)
      NIPE(J)=ITEMP
      GOTO 3
5    NIPE(L+1)=NIPE(J)
      NIPE(J)=INDXT
      JSTACK=JSTACK+2
      IF(JSTACK.GT.NSTACK)PAUSE 'NSTACK TOO SMALL IN INDEXX'

```

```
IF(IR-I+1.GE.J-L)THEN
  ISTACK(JSTACK)=IR
  ISTACK(JSTACK-1)=I
  IR=J-1
ELSE
  ISTACK(JSTACK)=J-1
  ISTACK(JSTACK-1)=L
  L=I
ENDIF
ENDIF
GOTO 1

RETURN
END
```

```

*****
*****          START OF STOCHO SUBROUTINE          *****
*****

```

C \*\*\* THIS SUBROUTINE SOLVES A DIAL ASSIGNMENT.

SUBROUTINE STOCHO(JX,X)

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

INTEGER LIMIT, LIM2, LIM3  
PARAMETER (LIMIT=21000, LIM2=1200, LIM3=8000)  
DATA INF /9999999/  
INTEGER BEGIN, JX  
LOGICAL NPOSSLST(LIMIT)

DOUBLE PRECISION LK(LIMIT),  
& W(LIMIT), SUM1(LIMIT),  
& X(LIMIT), XSUM(LIMIT), ZERO, TEMP

DOUBLE PRECISION P(LIM2,LIMIT), W1(LIM2,LIM2),  
& SUMJD(LIM2), WJTD(LIM2,LIMIT), WJD(LIMIT),  
& WOTI(LIMIT), WITJ(LIMIT)

COMMON /KSINPUT1/ NNODE, NUMORG,  
& NEXT(7400), NPRED(7400),  
& NTONODE(20400),NFMNODE(20400),XGUESS(20400),  
& GLINK(20400,2),SLINK(20400,4), CLINK(20400,6),NIPE(7400)

COMMON /KSINPUT2/ ABI(7400),  
& D(1200,7400),NLINK, NPOINT(7400)  
COMMON /KSINPUT2A/ THETA  
COMMON /KSINPUT3/ IDERIV, NPOSS, NPOSSLST

ZERO = 0.0D+00

C \*\*\* PRELIMINARIES FOR SETTING UP SHEFFI VERSION OF  
C \*\*\* SINGLE PASS OF DIAL ASSIGNMENT.

K = 1  
BEGIN = 1  
DO 70 I = 1,NNODE  
DO 69 J = BEGIN,NEXT(I)  
IF (ABI(I).LE.ABI(NTONODE(J))) THEN

```

        LK(K) = EXP(THETA*(ABI(NTONODE(J))-ABI(I)-CLINK(J,4)))
    ELSE
        LK(K) = ZERO
    ENDIF
        K = K + 1
69    CONTINUE
        BEGIN = NEXT(I) + 1
70    CONTINUE

C *****FIND LINK WEIGHTS - MODIFIED FORWARD PASS*****

        DO 77 I=1,NLINK
            W(I)=ZERO
            X(I)=ZERO
77    CONTINUE
        DO 78 I=1,NNODE
            SUM1(I)=ZERO
            XSUM(I)=ZERO
78    CONTINUE
            SUM1(JX)=1.0D+0
            DO 90 K=2,NNODE
                I=NIPE(K)

C ***** COMPUTE SUM(I) = SUM OF W(M,I)**
C ***** FIND LINKS GOING INTO I, ADD WTS.**

                DO 80 L=1,NLINK
                    IF (NTONODE(L) .EQ. I) THEN
                        W(L)=LK(L)*SUM1(NFMNODE(L))
                        SUM1(I) = SUM1(I) + W(L)
                    ENDIF
80    CONTINUE
90    CONTINUE

C CALCULATE LINK VOLUMES - MODIFIED BACKWARD PASS
C CONSIDER NODES IN DESCENDING ORDER OF DIST. FROM ORIGIN, JX

        TEMP=ZERO
        DO 110 K=0,(NNODE-2)

C ***** J = NODE NO.**

        J=NIPE(NNODE - K)

```

```

C ***** L = LINK NO.

      IF (SUM1(J).NE.ZERO) THEN
      TEMP=D(JX,J) + XSUM(J)
      IF (TEMP.NE.ZERO) THEN
      DO 105 L=1,NLINK
      IF ( NTONODE(L) .EQ. J. AND.
&      W(L).NE.ZERO ) THEN
      X(L)= ( TEMP ) * ( W(L)/SUM1(J) )
      XSUM(NFMNODE(L))=XSUM(NFMNODE(L))+X(L)
      ENDIF
105    CONTINUE
      ENDIF
      ENDIF
110  CONTINUE

      RETURN
      END

```

```
*****  
***** START OF MSA SUBROUTINE *****  
*****
```

SUBROUTINE MSA

C IN THIS SUBROUTINE THE MODIFIED MSA  
C IS IMPLEMENTED TO DETERMINE THE EQUILIBRIUM SOLUTION.

IMPLICIT DOUBLE PRECISION (A-H,O-Z)

INTEGER LIMIT, LIM3  
PARAMETER (LIMIT=21000,LIM3=8000)

LOGICAL NPOSSLST(LIMIT)

INTEGER ITNUM, MAXMSA

DOUBLE PRECISION ZERO, DIFF,  
& ERRMAX, ERRCRIT, FIT, YGUESS(LIMIT),  
& DA(LIMIT),XT(LIMIT),CONST(LIMIT)

COMMON /KSINPUT1/ NNODE, NUMORG,  
& NEXT(7400), NPRED(7400),  
& NTONODE(20400),NFMNODE(20400),XGUESS(20400),  
& GLINK(20400,2),SLINK(20400,4), CLINK(20400,6),NIPE(7400)

COMMON /KSINPUT2/ ABI(7400),  
& D(1200,7400),NLINK, NPOINT(7400)  
COMMON /KSINPUT2A/ THETA  
COMMON /KSINPUT3/ IDERIV, NPOSS, NPOSSLST  
COMMON /KSINPUT4/ NUMB, NAME

ZERO = 0.0D+0  
MAXMSA = 32000  
ERRCRIT = 1.0D-01

C THE RESULTS FILE IS AN INPUT/OUTPUT FILE USED TO RECORD  
C THE CURRENT ITERATION SOLUTION AND IF THE PROGRAM HAS TO  
C BE INTERRUPTED THEN THIS ALSO SERVES AS THE INPUT FILE  
C TO RESTART THE PROGRAM FROM THAT POINT.

OPEN (UNIT=50,FILE='RESULTS',STATUS='OLD')  
READ(50,\*) ITNUM, ERRMAX

```

DO 111 K=1,NLINK
  READ(50,*) XGUESS(K), YGUESS(K)
111 CONTINUE
  CLOSE(50)

```

C THE INPUT FILES FOR THE DIFFERENT NDPS WERE SET UP SLIGHTLY  
C DIFFERENTLY, SO THE PROGRAM DETERMINES WHICH PROBLEM IT IS  
C AND SETS UP THE PARAMETERS ACCORDINGLY

C PROBLEM 2 IS WASECA.

```

  IF (NUMB.EQ.2) THEN
    DO 22 K=1,NLINK
      IF (NPOSSLST(K)) THEN
        CONST(K)= ((( 1 / CLINK (K,2) ) *
&      ( (SLINK(K,1)/GLINK(K,2)) - 1 ) ) ** (2.5D-01))
      ENDIF
22  CONTINUE
    ENDIF

```

C PROBLEM 1 IS LITTLE, AND 3 IS THE TWIN CITIES

```

  IF (NUMB.NE.2) THEN
    DO 23 K=1,NLINK
      IF (NPOSSLST(K)) THEN
        CONST(K)= GLINK(K,2)
      ENDIF
23  CONTINUE
    ENDIF

```

C THIS IS THE BEGINING OF THE METHOD OF SUCCESSIVE AVERAGES  
C ITERATIVE PROCESS. THIS RUNS EITHER UNTIL THE ERROR IS LESS  
C THAN THE TOLERANCE SPECIFIED, OR UNTIL A PRESET MAXIMUM NO.  
C OF ITERATIONS IS REACHED.

```

  ITNUM=0
100 IF (ITNUM.LT.MAXMSA) THEN
  ITNUM=ITNUM+1

```

C THIS UPDATES THE EXPANSION VARIABLE FOR EACH LINK, IF THE  
C LINK IS A CANDIDATE AND THE V/C RATIO IS GREATER THAN  
C LOS CONSTRAINT, THEN THE LINK IS EXPANDED, OTHERWISE  
C IT IS LEFT ALONE.



```

DO 5 K=1,NLINK
  IF(NPOSSLST(K))THEN
    TEMP=(XGUESS(K)/CONST(K))-CLINK(K,3)
    YGUESS(K)=MAX(ZERO,TEMP)
  ENDIF
5   CONTINUE

```

C NOW THE TRAVEL TIME IS ADJUSTED TO ACCOUNT FOR CURRENT  
C TRAFFIC ASSIGNMENT.

```

DO 110 K=1,NLINK
  CAP = CLINK(K,3)
  IF(NPOSSLST(K))THEN
    CAP = CAP + YGUESS(K)
  ENDIF
  VC = (XGUESS(K)/CAP)
  IF(NPOSSLST(K))THEN
    IF (YGUESS(K).GT.ZERO) THEN
      CLINK(K,4)=CLINK(K,1)*
&      (1.0D+0+1.5D-01*CONST(K)**4.0D+0)
    ELSE
      CLINK(K,4)=CLINK(K,1)*
&      (1.0D+0+1.5D-01*(VC)**4.0D+0)
    ENDIF
  ELSE
    CLINK(K,4)=CLINK(K,1)*
&      (1.0D+0+1.5D-01*(VC)**4.0D+0)
  ENDIF
  DA(K)=ZERO
  XT(K)=ZERO
110  CONTINUE

```

C USING THE CURRENT CONGESTED TVL TIMES, NEW SHORTEST PATHS  
C ARE CALCULATED AND A NEW TRAFFIC ASSIGNMENT CARRIED OUT,  
C THIS GIVES THE AUXILIARY LINK FLOW PATTERN.

```

DO 120 J=1,NUMORG
  CALL SHPTHL(J)
  CALL INDEXX
  CALL STOCHO(J,XT)
DO 115 K = 1, NLINK
115  DA(K) = DA(K) + XT(K)
120  CONTINUE
  ERRMAX = ZERO

```

```

FIT = FLOAT(ITNUM)

C THE NEW FLOW IS THEN FOUND (THIS IS THE 'MOVE' DIRECTION),

DO 130 K = 1, NLINK
  DIFF = DA(K) - XGUESS(K)
  ERRMAX = DMAX1(ERRMAX,DABS(DIFF))
  XGUESS(K) = XGUESS(K) + DIFF/FIT
130  CONTINUE

C WRITE TO OUT FILE TO KEEP A RECORD OF CONVERGENCE.

WRITE(3,*) 'ITNUM = ', ITNUM, ', ERRMAX = ', ERRMAX

C KEEPING TRACK OF CURRENT RESULT EACH ITERATION.
OPEN (UNIT=4,FILE='RESULTS',STATUS='UNKNOWN')
WRITE(4,*) ITNUM, ERRMAX
DO 12 K=1,NLINK
  WRITE(4,'(2E20.10)')XGUESS(K),YGUESS(K)
12  CONTINUE
CLOSE (4)

C IF CONVERGENCE IS ATTAINED THEN WE'RE DONE, OTHERWISE WE
C GO BACK AND DO ANOTHER ITERATION.
C *** ERRMAX IS CURRENT SOL'NS DIFF IN ASSIGNMENT FROM PREVIOUS,
C *** ERRCRIT IS DESIRED CONVERGENCE TOLERANCE.

IF (ERRMAX.GT.ERRCRIT) GOTO 100

ENDIF

C IF CONVERGENCE IS REACHED, WRITE FINAL RESULTS TO OUT.

WRITE(3,*) ITNUM, ERRMAX
DO 11 K=1,NLINK
CAP=CLINK(K,3)
ID=0
IF (SLINK(I,1).EQ.1.OR.SLINK(I,1).EQ.2) THEN
  ID=1
ENDIF
IF(NPOSSLST(K))THEN
  CAP=CAP+YGUESS(K)
ENDIF
VC = (XGUESS(K)/CAP)

```

```
      WRITE(3,'(3E14.5)')XGUESS(K),YGUESS(K), CLINK(K,4)
11    CONTINUE
```

```
C *** DONE WITH MSA SUBROUTINE
```

```
      RETURN
      END
```

**APPENDIX C**  
**NETWORK DATA**

**Table C-1 "Waseca" Link Parameters**

<b>Link</b>	<b>A - B</b>	<b>Capacity</b>	<b>Link Length</b>	<b>Free-flow speed</b>	<b>Link A - B</b>	<b>Capacity</b>	<b>Link Length</b>	<b>Free-flow speed</b>
01 - 48		2400	0.23	15	39 - 26	1200	0.74	40
01 - 54		2400	0.22	15	39 - 38	1200	0.32	40
01 - 57		2400	0.28	15	40 - 38	1200	0.62	40
02 - 31		2400	0.33	15	40 - 41	1200	0.24	40
02 - 32		2400	0.18	15	40 - 45	1200	0.11	40
02 - 33		2400	0.23	15	40 - 47	1200	0.75	40
02 - 54		2400	0.25	15	41 - 09	2400	0.21	15
03 - 55		2400	0.12	15	41 - 40	1200	0.24	40
03 - 56		2400	0.09	15	41 - 42	1200	0.51	40
03 - 69		2400	0.13	15	41 - 44	1200	0.85	40
04 - 47		2400	0.23	15	42 - 10	2400	0.08	15
04 - 48		2400	0.23	15	42 - 41	1200	0.51	40
05 - 51		2400	0.23	15	42 - 43	1200	0.27	40
05 - 58		2400	0.57	15	43 - 42	1200	0.27	40
05 - 59		2400	0.41	15	43 - 44	1200	0.37	40
06 - 21		2400	0.10	15	44 - 11	1200	1.68	40
07 - 27		2400	0.12	15	44 - 41	1200	0.85	40
07 - 36		2400	0.28	15	44 - 43	1200	0.37	40
08 - 46		2400	0.70	15	45 - 40	1200	0.11	40
08 - 51		2400	0.26	15	45 - 46	1200	0.38	40
09 - 41		2400	0.21	15	46 - 08	2400	0.70	15
10 - 42		2400	0.08	15	46 - 45	1200	0.38	40
11 - 44		1200	1.68	40	46 - 50	1200	0.67	40
12 - 67		1200	0.86	40	47 - 04	2400	0.23	15
13 - 63		1200	1.30	40	47 - 40	1200	0.75	40
14 - 18		1200	0.88	40	47 - 49	1200	0.23	40
15 - 25		1200	0.29	40	47 - 56	1200	0.17	40
16 - 39		2400	1.00	15	48 - 01	2400	0.23	15
17 - 66		2400	1.52	15	48 - 04	2400	0.23	15
18 - 14		1200	0.88	40	48 - 49	1200	0.22	40
18 - 19		1200	0.40	40	48 - 50	1200	0.22	40
19 - 18		1200	0.40	40	49 - 47	1200	0.23	40
19 - 20		1200	0.06	40	49 - 48	1200	0.22	40
19 - 24		1200	0.47	40	49 - 54	1200	0.27	40
20 - 19		1200	0.06	40	49 - 55	1200	0.16	40
20 - 21		1200	0.08	40	50 - 46	1200	0.67	40
20 - 22		1200	0.47	40	50 - 48	1200	0.22	40
21 - 06		2400	0.10	15	50 - 51	1200	0.60	40
21 - 20		1200	0.08	40	50 - 52	1200	0.12	40
21 - 34		1200	0.37	40	51 - 05	2400	0.23	15
22 - 20		1200	0.47	40	51 - 08	2400	0.26	15
22 - 23		1200	0.09	40	51 - 50	1200	0.60	40
22 - 68		1200	0.06	40	51 - 67	1200	0.60	40
23 - 22		1200	0.09	40	52 - 50	1200	0.12	40
23 - 24		1200	0.04	40	52 - 53	1200	0.10	40
23 - 26		1200	0.28	40	53 - 52	1200	0.10	40

24 - 19	1200	0.47	40	53 - 58	1200	0.39	40
24 - 23	1200	0.04	40	54 - 01	2400	0.22	15
25 - 15	1200	0.29	40	54 - 02	2400	0.25	15
25 - 26	1200	0.09	40	54 - 49	1200	0.27	40
26 - 23	1200	0.28	40	54 - 62	1200	0.25	40
26 - 25	1200	0.09	40	55 - 03	2400	0.12	15
26 - 27	1200	0.17	40	55 - 31	1200	0.15	40
26 - 39	1200	0.74	40	55 - 49	1200	0.16	40
27 - 07	2400	0.12	15	56 - 03	2400	0.09	15
27 - 26	1200	0.17	40	56 - 35	1200	0.11	40
27 - 28	1200	0.09	40	56 - 47	1200	0.17	40
28 - 27	1200	0.09	40	57 - 01	2400	0.28	15
28 - 29	1200	0.18	40	57 - 58	1200	0.35	40
28 - 35	1200	0.26	40	57 - 62	1200	0.18	40
29 - 28	1200	0.18	40	58 - 05	2400	0.57	15
29 - 30	1200	0.12	40	58 - 53	1200	0.39	40
29 - 68	1200	0.24	40	58 - 57	1200	0.35	40
30 - 29	1200	0.12	40	58 - 60	1200	0.21	40
30 - 31	1200	0.12	40	59 - 05	2400	0.41	15
30 - 32	1200	0.32	40	59 - 60	1200	0.52	40
31 - 02	2400	0.33	15	59 - 61	1200	0.50	40
31 - 30	1200	0.12	40	60 - 58	1200	0.21	40
31 - 55	1200	0.15	40	60 - 59	1200	0.52	40
31 - 69	1200	0.10	40	61 - 59	1200	0.50	40
32 - 02	2400	0.18	15	61 - 66	1200	0.79	40
32 - 30	1200	0.32	40	62 - 33	1200	0.22	40
32 - 34	1200	0.23	40	62 - 54	1200	0.25	40
33 - 02	2400	0.23	15	62 - 57	1200	0.18	40
33 - 34	1200	0.19	40	62 - 64	1200	0.93	40
33 - 62	1200	0.22	40	63 - 13	1200	1.30	40
34 - 21	1200	0.37	40	63 - 64	1200	0.15	40
34 - 32	1200	0.23	40	64 - 62	1200	0.93	40
34 - 33	1200	0.19	40	64 - 63	1200	0.15	40
35 - 28	1200	0.26	40	64 - 65	1200	0.44	40
35 - 37	1200	0.35	40	65 - 64	1200	0.44	40
35 - 56	1200	0.11	40	65 - 66	1200	0.49	40
35 - 69	1200	0.07	40	66 - 17	2400	1.52	15
36 - 07	2400	0.28	15	66 - 61	1200	0.79	40
36 - 37	1200	0.20	40	66 - 65	1200	0.49	40
36 - 38	1200	0.36	40	67 - 12	1200	0.86	40
37 - 35	1200	0.35	40	67 - 51	1200	0.6	40
37 - 36	1200	0.20	40	68 - 22	1200	0.06	40
38 - 36	1200	0.36	40	68 - 29	1200	0.24	40
38 - 39	1200	0.32	40	69 - 03	2400	0.13	15
38 - 40	1200	0.62	40	69 - 31	1200	0.10	40
39 - 16	2400	1.00	15	69 - 35	1200	0.07	40

**Table C-2 "Waseca" Travel Demand Matrix**

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	59	47	20	15	15	17	55	67	26	72	84	24	49	59	19	133	58
2	48	47	17	12	11	17	56	54	24	67	77	98	41	60	18	138	50
3	32	27	4	5	7	15	51	48	22	69	80	81	21	42	1	151	3
4	20	16	2	4	5	9	27	30	15	42	46	52	13	24	3	80	7
5	18	15	6	4	7	5	22	37	10	32	37	64	13	20	4	59	14
6	12	10	8	5	3	2	10	8	3	5	7	19	15	16	12	10	30
7	45	45	31	18	17	13	66	50	22	60	70	100	48	61	40	130	76
8	57	46	31	22	26	12	51	83	24	65	75	181	53	53	31	102	86
9	19	17	13	8	7	4	19	22	15	35	42	44	21	19	14	46	38
10	54	46	43	26	22	7	51	55	33	113	116	114	60	50	46	100	110
11	61	56	48	30	24	10	60	65	39	121	261	137	70	60	50	126	128
12	115	89	50	37	50	28	114	199	54	150	173	601	96	110	47	255	139
13	56	50	14	11	13	24	64	71	31	89	104	122	109	73	12	172	61
14	52	52	25	17	15	22	68	56	23	65	74	106	56	141	34	159	80
15	31	30	3	4	6	20	64	48	24	71	81	79	19	55	0	204	0
16	86	87	91	49	37	8	90	70	34	76	92	166	109	107	129	203	226
17	96	82	6	12	24	51	124	141	61	182	208	230	98	132	0	368	0