REPORT NO. DOT-TSC-RSPD-78-8,I

NETWORK AGGREGATION IN TRANSPORTATION PLANNING
Volume I: Summary and Survey

Donald W. Hearn

Mathtech, Inc.
P.O. Box 2392
Princeton NJ 08540

APRIL 1978

FINAL REPORT

DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION

RESEARCH AND SPECIAL PROGRAMS DIRECTORATE

Office of Transportation Programs Bureau

Office of Systems Engineering
Washington DC 20590

NOTICE

This document is disseminated under the sponsorship of the Department of Transportation in the interest of information exchange. The United States Government assumes no liability for its contents or use thereof.

NOTICE

The United States Government does not endorse products or manufacturers. Trade or manufacturers' names appear herein solely because they are considered essential to the object of this report.

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT-TSC-RSPD-78-8, I		

4. Title and Subtitle	5. Report Date
NETWORK AGGREGATION IN TRANSPORTATION PLANNING Volume I: Summary and Survey	April 1978
	6. Performing Organization Code
7. Author(s) Donald W. Hearn	8. Performing Organization Report No. DOT-TSC-RSPD-78-8, I
9. Performing Organization Name and Address Mathtech, Inc.* P.O. Box 2392 Princeton NJ 08540	10. Work Unit No. (TRAIS) OS850/R8526
	11. Contract or Grant No. DOT-TSC-1232-1
12. Sponsoring Agency Name and Address U.S. Department of Transportation Research and Special Programs Directorate Office of Transportation Programs Bureau Office of Systems Engineering Washington DC 20590	13. Type of Report and Period Covered FINAL REPORT July 1976-July 1977
	14. Sponsoring Agency Code

15. Supplementary Notes
*Under contract to: U.S. Department of Transportation
Transportation Systems Center
Kendall Square
Cambridge MA 02142

16 Abstract

This, the first of two volumes, summarizes research on network aggregation in transportation models. It includes a survey of network aggregation practices, definition of an extraction aggregation model, computational results on a heuristic implementation of the model, and related mathematical results.

Volume II defines a new algorithm for the network equilibrium model that works in the space of path flows and is based on the theory of fixed point methods.

Volume II has 76 pages.

17. Key Words	18. Distribution Statement
Aggregation Duality Networks Traffic Assignment Fixed Point Method	DOCUMENT IS AVAILABLE TO THE U.S. PUBLIC THROUGH THE NATIONAL TECHNICAL INFORMATION SERVICE, SPRINGFIELD, VIRGINIA 22161

19. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of Pages	22. Price
Unclassified	Unclassified	100	

Form DOT F 1700.7 (8-72) Reproduction of completed page authorized

Preface

This report summarizes the research on Aggregation in Transportation Networks conducted by MATHTECH, Inc. under contract DOT-TSC-1232 for the U.S. Department of Transportation. Funding for the work was provided under Project TARP (OST/TST) with additional support provided by UMTA. The goals of this study were broadly defined as the identification of aggregation practices and the development of a framework for studying these practices. These goals have been accomplished by, (a) conducting a survey of aggregation practices, (b) formulating an aggregation model, (c) conducting a computational study, (d) deriving mathematical programming formulations aimed at making steps of the model precise, and (e) programming and testing a particular algorithm proposed in earlier MATHTECH research.

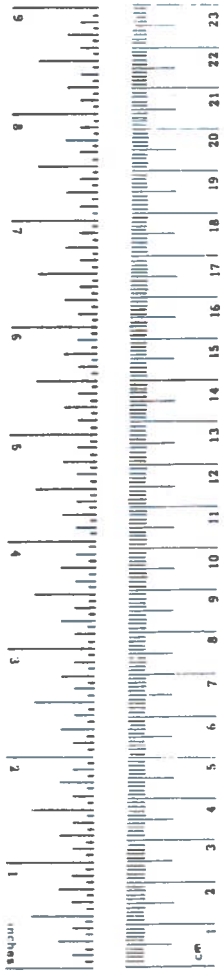
A number of people were very helpful in the survey stage of this project. In particular, thanks are due to William S. Mann, Washington Council of Governments Transportation Planning Board, Morris J. Rothenberg, JHK Associates, Bob Dial of UMTA, Raphael Kedar and Tom Bouvé of FRA for the time taken to discuss their aggregation schemes.

Principal consultant on this project was Professor Harold W. Kuhn, Princeton University, who offered many helpful suggestions throughout the study. A number of the ideas explored here originated in the MATHTECH report "Aggregation in Net work Models for Transportation Planning" (DOT-TSC-883) by Harold W. Kuhn and Daniel E. Cullen. The research was guided throughout by Dr. Edwin J. Roberts and Mr. Michael Nienhaus of TSC and by Mr. Robert Crosby of the Office of the Secretary.

METRIC CONVERSION FACTORS

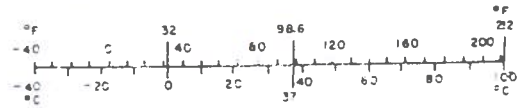
Approximate Conversions to Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
in	inches	2.5	centimeters	cm
ft	feet	30	centimeters	cm
yd	yards	0.9	meters	m
mi	miles	1.6	kilometers	km
		AREA		
in^2	square inches	6.5	square centimeters	cm^2
ft^2	square feet	0.09	square meters	m^2
yd^2	square yards	0.8	square meters	m^2
mi^2	square miles	2.6	square kilometers	km^2
	acres	0.4	hectares	ha
		MASS (weight)		
oz	ounces	28	grams	g
lb	pounds	0.45	kilograms	kg
	short tons (2000 lb)	0.9	tonnes	t
		VOLUME		
tsp	teaspoons	5	milliliters	ml
Tbsp	tablespoons	15	milliliters	ml
fl oz	fluid ounces	30	milliliters	ml
c	cups	0.24	liters	l
pt	pints	0.47	liters	l
qt	quarts	0.95	liters	l
gal	gallons	3.8	liters	l
ft^3	cubic feet	0.03	cubic meters	m^3
yd^3	cubic yards	0.76	cubic meters	m^3
		TEMPERATURE (exact)		
°F	Fahrenheit temperature	5/9 (after subtracting 32)	Celsius temperature	°C



Approximate Conversions from Metric Measures

Symbol	When You Know	Multiply by	To Find	Symbol
		LENGTH		
mm	millimeters	0.04	inches	in
cm	centimeters	0.4	inches	in
m	meters	3.3	feet	ft
m	meters	1.1	yards	yd
km	kilometers	0.6	miles	mi
		AREA		
cm^2	square centimeters	0.16	square inches	in^2
m^2	square meters	1.2	square yards	yd^2
km^2	square kilometers	0.4	square miles	mi^2
ha	hectares (10,000 m^2)	2.5	acres	
		MASS (weight)		
g	grams	0.035	ounces	oz
kg	kilograms	2.2	pounds	lb
t	tonnes (1000 kg)	1.1	short tons	
		VOLUME		
ml	milliliters	0.03	fluid ounces	fl oz
l	liters	2.1	pints	pt
l	liters	1.06	quarts	qt
l	liters	0.26	gallons	gal
m^3	cubic meters	35	cubic feet	ft^3
m^3	cubic meters	1.3	cubic yards	yd^3
		TEMPERATURE (exact)		
°C	Celsius temperature	9/5 (then add 32)	Fahrenheit temperature	°F



CONTENTS

Section			Page
1.	Introduction		1-1
2.	Survey of Some Aggregation Practices		2-1
	2.1	Introduction	2-1
	2.2	Shirley Highway Dedicated Lane Study	2-2
	2.3	Dial's Origin Aggregation	2-7
	2.4	Wilson's Load Node Concept	2-10
	2.5	Mann's Short Trip/Long Trip Aggregation	2-15
	2.6	Aggregation in a Railway Study	2-18
	2.7	Summary	2-22
		APPENDIX Criteria for Prototype Study	2-27
		REFERENCES	2-31
3.	Computational Study of Extraction Aggregation		3-1
	3.1	Introduction	3-1
	3.2	Given Network and Trip Table	3-2
	3.3	Links of Interest and Subnetwork	3-4
	3.4	Selection of Pseudo-Centroids	3-7
	3.5	Transfer of the Trip Table	3-7
	3.6	Flowing the Subnetwork and Measuring	3-8
	3.7	Additional Heuristics	3-12
	3.8	Conclusions	3-19
		REFERENCES	3-23

CONTENTS (Continued)

Section			Page
4.	Bounding Error in the Traffic Assignment Problem		4-1
	4.1	Introduction	4-1
	4.2	Problem Formulation and Notation	4-1
	4.3	Background	4-6
	4.4	Proofs of Equivalences	4-8
	4.5	Geometrical Interpretation	4-13
	4.6	Improving the Bound	4-16
	4.7	Minimizing the Gap	4-19
		REFERENCES	4-21
5.	Mathematical Programming and Extraction Aggregation		5-1
	5.1	Introduction	5-1
	5.2	Transfer and Flow by Convex Programming	5-1
	5.3	Comparison by Linear Programming	5-10
		REFERENCES	5-12

LIST OF FIGURES

Figure		Page
2-1	Aggregated Network of Dedicated Lane Study	2-4
2-2	Dedicated Lanes - Allowed Accesses and Exits November 1975	2-5
2-3	Dial's Origin Aggregation	2-8
2-4	Zone Centroid vs. Load Node	2-11
2-5	Adjacent Zone Trip Assignment - Zone Centroid vs. Load Node	2-12
2-6	Long Trip/Short Trip Aggregation	2-16
2-7	Example of Circuity	2-19
2-8	Segment Aggregation of FRA Network	2-20
2-9	Extraction Aggregation with Feedback	2-26
3-1	Elimination of an Outlying Subtree	3-5
3-2	Sketch of Subnetwork	3-6
3-3	Lifting of Flows to Given Network	3-9
3-4	Minimum Path Tree on the Subnetwork	3-13
4-1	Six Node Network	4-3
4-2	Geometrical Interpretation of Convexity Bound	4-14
4-3	Interpretation of G(x) in the Frank-Wolfe Algorithm	4-15
4-4	Obtaining the Minimax Bound	4-17
5-1	Attachment of Centroids to Pseudo Centroids	5-3
5-2	Non-Uniqueness of Trip Table for the Extracted Network	5-6

LIST OF TABLES

Table		Page
2-1	Comparison of Network Detail for Phoenix Network	2-1
2-2	Aggregation in Urban Highway Studies	2-24
2-3	Aggregation in a Railway Study	2-25
3-1	Flows on Shirley Highway Links (Northbound)	3-10
3-2	NLP Comparisons of Benchmark with Aggregation	3-11
3-3	Flows on Shirley Highway Links (Northbound)(Cont.)	3-15
3-4	NLP Comparisons of Benchmark with Other Aggregations	3-16
3-5	Aggregation Solutions as Advance Start for TRAFFIC	3-17
3-6	Gross Measures -- Aggregation versus Benchmark	3-18

1. Introduction

This study centers around practices of network aggregation that exist in transportation planning models and mathematical models which might improve these practices. Aggregation of networks may involve either (a) abstraction, as when links or nodes are combined into a single abstract link or node; or (b) extraction of subnetworks.

In accordance with the defined scope of this project, we have interviewed several users of large transportation networks to determine what aggregation methods (if any) they employ. The practices encountered were solely of the extraction type and therefore had similar characteristics. One of these was chosen for computer simulation and some experimentation done with parameters. A tentative conclusion is that simple heuristics aimed at determining the appropriate portion of total demand utilizing the extracted subnetwork can yield good solutions which rival more sophisticated methods.

Finally, we have attempted to bring some mathematical theory to bear on the problem of extraction aggregation. Included herein is a rather detailed analysis of the "duality gap" of nonlinear programming and its role in estimating error in the traffic assignment problem. In addition, we suggest several models for further study and application to the extraction aggregation process.

The remainder of Part I of this report is organized as follows. Chapter 2 contains a summary of the aggregation practices encountered in the survey. These practices have a common structure which is identified

and explained in the form of a very general Extraction Aggregation Model. This terminology emphasizes that links are extracted from a large network and retain their basic characteristics, as opposed to being combined into abstract links. Chapter 3 describes a large scale computational study of the Extraction Aggregation Model. This study was designed to be realistic and use actual data so that conclusions could be drawn about the validity of aggregation practices and thereby the areas where research results are most needed. Chapter 4 concentrates on theoretical results, especially with regard to the measurement of aggregation error, and Chapter 5 continues this thrust by posing mathematical programming models aimed at improving the extraction aggregation process. Both Chapter 4 and Chapter 5 contain a number of new research problems, and it is the intent to spur the development of alternative models.

Part II of this report (under separate cover) defines a new algorithm (named PATHFIX) for the traffic assignment problem. Program documentation and the results of several test runs are included.

2. Survey of Some Aggregation Practices

2.1 Introduction

It is generally conceded that aggregation practices abound in transportation planning, but that little record is made of just what these practices are. To partially remedy this situation and provide a framework for mathematical research, MATHTECH, with the assistance of its technical monitors, has conducted an *ad hoc* survey of DOT agencies and certain other sources in a search for aggregation practices sufficiently well defined to document. The survey consisted of site visits to the following groups (by mode):

Mode	Group
Highway	FHWA, COG, NBS
Rail	FRA, NBS, OST/TST-13
Urban Highway	UMTA, COG, JHK, NBS
Multimodal Freight	OST/TST-13, TSC
Pipeline	TSC
Air	OST/TPI-10
Water	TSC

Note: COG = Washington Council of Governments Transportation Planning Board.

JHK = JHK and Associates, Alexandria, Virginia

The goal of this survey was not only to record aggregation practices, but also to choose a prototype for study and analysis. The Appendix by Harold Kuhn describes the criteria employed in the prototype selection. The remaining sections detail the aggregation schemes encountered. In presenting these, we suggest that savings from aggregation of transportation networks are of two forms.* The first is savings of computational effort (usually computer time) and the second is savings of space (such as computer storage). If we assume that the primary purpose of having the network is to simulate flows on it, then it follows that the computational effort is likely to be proportional to the time required to compute minimum path trees. For a network of ℓ links this is an $O(\ell)$ operation per tree [1].+ If m minimum path trees must be computed to flow the network, then the time effort, t, takes the form

$$t \sim m\ell . \qquad (2.1.1)$$

The space required, s, is simply proportional to the number of links in the network:

$$s \sim \ell . \qquad (2.1.2)$$

We will use (2.1.1) and (2.1.2) to estimate the potential savings of the aggregation practices discussed.

2.2 Shirley Highway Dedicated Lane Study

The Shirley Highway is I95 south from Washington, D. C. and is the primary artery for Virginia commuters who work in the District (see Figure 2-1). What was formally the median strip is now two limited-

---

* Another very important savings, not evident in these examples, is that of data collection.

\+ For dense networks $\ell \doteq n^2$ where n is the number of nodes. Transportation networks, however, are very sparse.

access, dedicated lanes from just south of the Capitol Beltway to the north end of the 14th Street Bridge. These (reversible) lanes are available only to buses and four person car pools. Furthermore, while buses may use all access ramps, as of November 1975, the four person car pools could only use certain ones. The situation at that time is shown in Figure 2-2.

JHK and Associates of Alexandria, Virginia, conducted a study (Contract DOT-FH-11-8242) in 1976 to determine whether allowing car pools to use the additional ramps and/or reducing the required size of a car pool from four persons to three would overcrowd the dedicated lanes. The results of their study are in reference [2].

As a first step in their analysis, JHK performed a traffic assignment procedure to determine traffic flows on major arteries in the a.m. rush. Although their study consisted of many other important aspects, it is just the traffic assignment procedure and the preceding aggregation which we consider here. Following are the steps of the process:

1. JHK obtained trip tables for the a.m. rush period and the COG network of Metropolitan Washington.
2. The study area consisted of the Virginia suburbs and the downtown part of the District. For this area the COG network consists of approximately 9,000 links, 3,000 nodes and 700 zone centroids. From this the network shown in Figure 2-1 was extracted. (The dedicated lane was ignored in this network.) Constant travel times were assigned to the extracted links.

Figure 2-1: Aggregated Network of Dedicated Lane Study

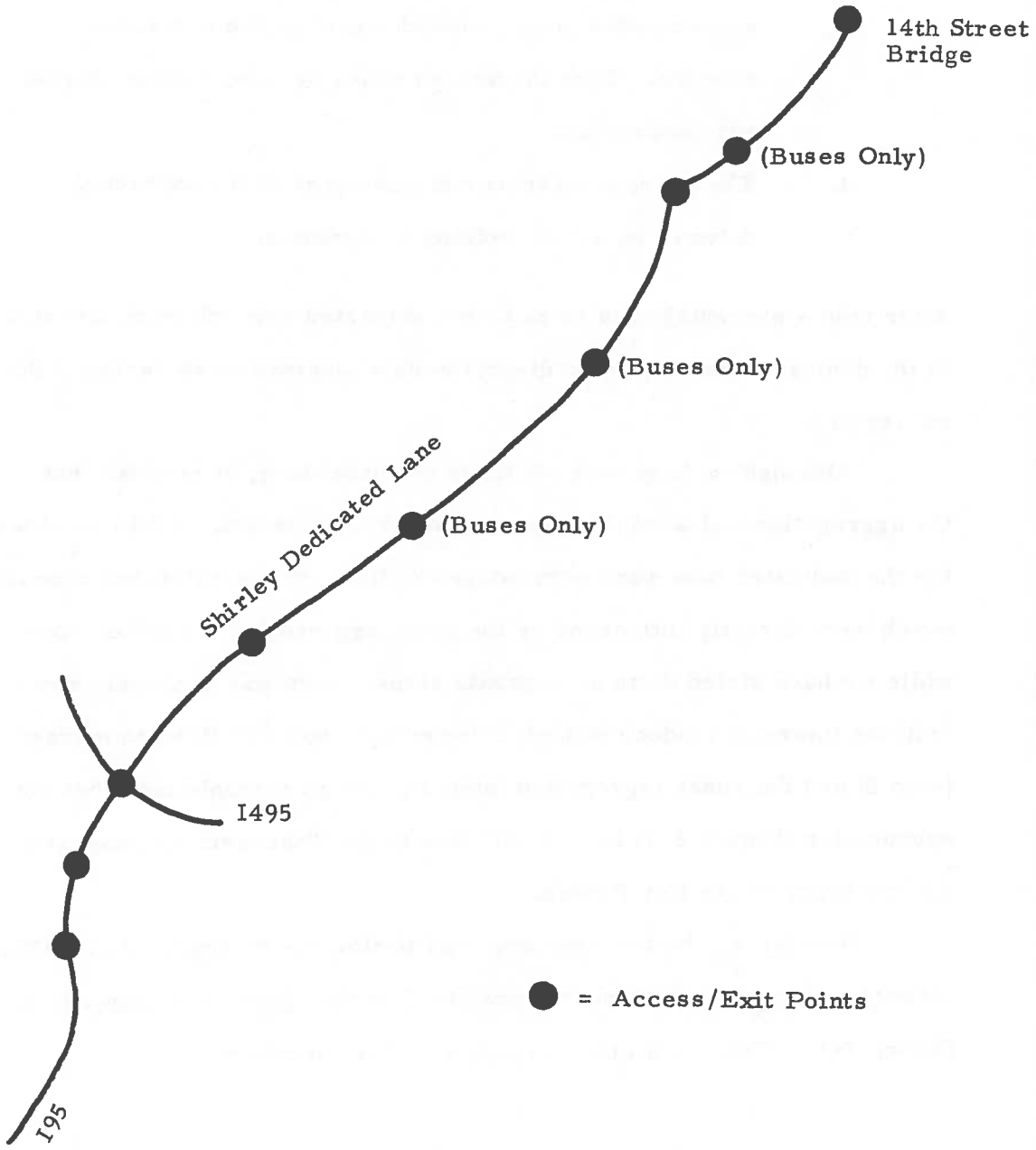


Figure 2-2: Dedicated Lanes - Allowed
Accesses and Exits November 1975

3. The 700 zones were aggregated by hand to 42 and, in effect, the aggregate trip demands were assigned to access nodes (e.g., interchanges) on the extracted network. Thus the access nodes became pseudo origins and destinations.

4. The aggregated trips were assigned to the extracted network by all-or-nothing assignment.

After step 4 percentages of trips on the extracted network were diverted to the dedicated lane based on diversion data obtained in an earlier NBS survey [3].

Although we have only sketched the procedure, it is clear that the aggregation had a major impact on the study results. All flows obtained for the dedicated lane were percentages of flows on the extracted network which were directly influenced by the zonal aggregation. Furthermore, while we have stated them as separate steps, there was obviously some delicate interplay (undocumented) between the choice of links to extract (step 2) and the zonal aggregation (step 3). As an example note that the aggregation (Figure 2-1) forces all trips to the Wisconsin Avenue area (Georgetown) to use Key Bridge.

Now let t_D be the time required to flow the disaggregated (COG) network and t_A be the time required to flow the aggregated network of Figure 2-1. Thus, from (2.1.1), we have the formulas:

$$\frac{t_D}{t_A} = \frac{700 \cdot (9,000)}{42 \cdot (1,000)} \doteq 150$$

and from (2.1.2)

$$\frac{s_D}{s_A} = \frac{9,000}{1,000} \doteq 9$$

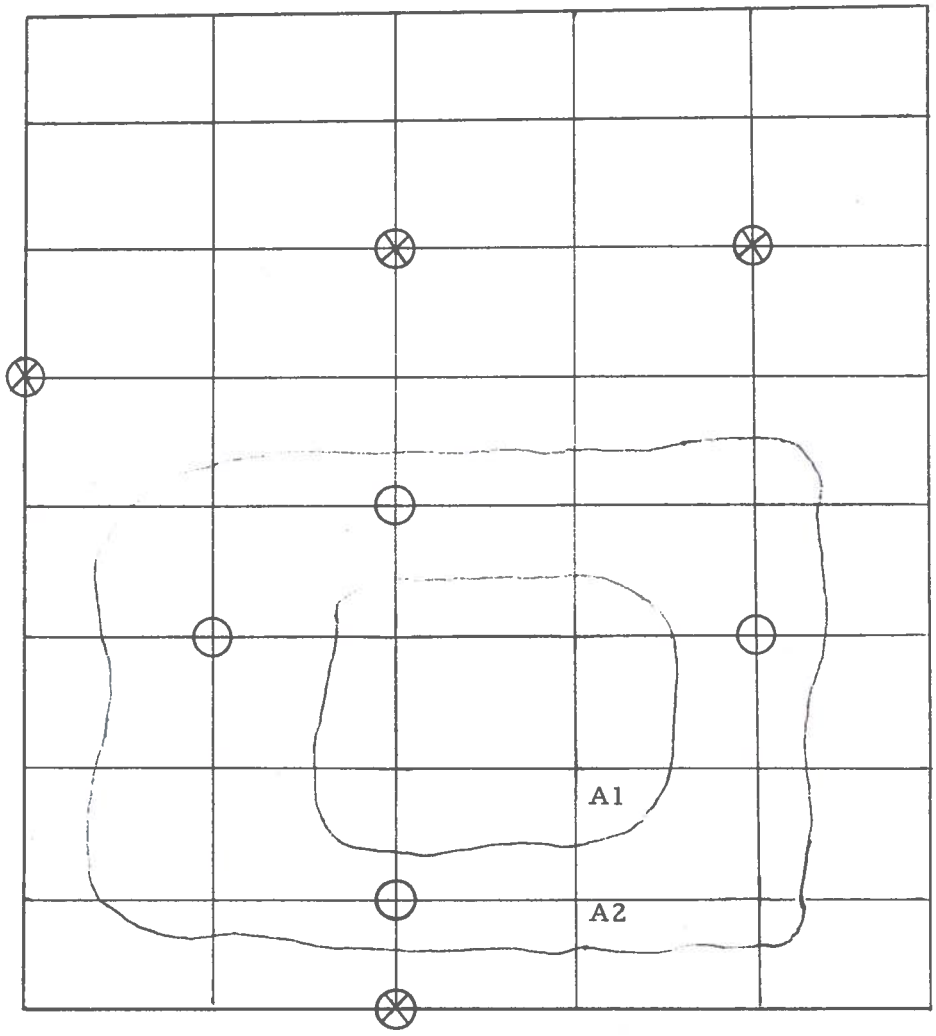
Hence the potential savings is between two and three orders of magnitude with respect to time and about one with respect to space.

2.3 Dial's Origin Aggregation

Bob Dial of UMTA has an aggregation scheme in an experimental traffic assignment code which is designed to work in conjunction with sub-area focusing. The test networks are very similar to typical urban highway networks, the chief difference being that points of origin and destination are at actual network nodes; there are no centroids.

The aggregation scheme (see Figure 2-3) is as follows:

1. A network and trip table are assumed given.
2. The user identifies two types of nodes: (i) candidate district centers and (ii) candidate super district centers.
3. The user also identifies a contiguous subset of the network as being the focus area (A1) of interest in which all network detail is to be preserved. He may also identify a nearby area (A2), usually a "ring" about A1.



Legend

⊕ Candidate District Center

⊛ Candidate Super District Center

(Assume any intersection may be an origin and/or destination.)

Figure 2-3: Dial's Origin Aggregation

4. The code then assigns every origin node in area A2 to some candidate district center. That is, the trips from the origin node are treated as being from the district center; the origin node remains in the network just as a node.

5. Similarly, the origin nodes not in A1 or A2 are assigned to super district nodes.

6. An equilibrium assignment of trips from the new origins to the destinations is made.

The assignment of origin nodes to district or superdistrict nodes is done by finding the nearest such in terms of link lengths (not travel times). Note also that candidate centers to which no origins are assigned are treated just as nodes in the assignment process.

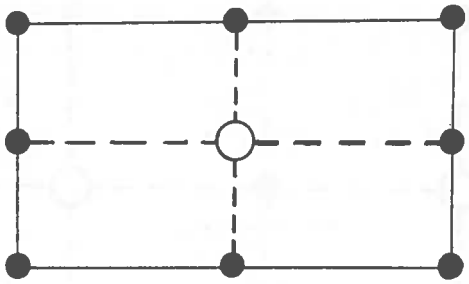
This scheme has shown promise when comparison is made against an equilibrium flow obtained without the origin aggregation. The only link flows considered in this comparison are those in area A1.

Dial makes the point that this aggregation can be thought of as transforming a square (say) OD matrix to a rectangular one. Since the assignment process consists of constructing minimum spanning trees from each origin, there would be negligible savings by aggregation of destinations in the same manner. Our measure of time savings (2.1.1) is in agreement with this observation. Since links are not deleted from the network, the reduction is just in the quantity m in our formula. There are no space savings since the entire network is retained.

2.4 Wilson's Load Node Concept

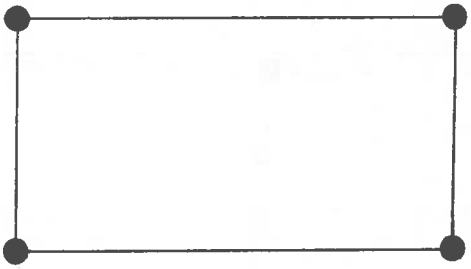
The dissertation of Wilson [4] and the subsequent paper by Wilson, Matthias and Betz [5] presents an aggregation method similar in spirit to the two already discussed. The idea, intended for traffic assignment for urban networks, is to remove all zone centroids and centroid connectors. Trip demands to and from each zone are "loaded" on nearby nodes. Figure 2-4 (from [4]) gives an illustration of maximum potential savings. The idea is facilitated by the fact that zone boundaries are usually streets.

Wilson mentions two methods for implementing the idea. One is to redefine zones so that actual network nodes are near the centers and hence can play the role of centroids. This concept, however, requires a fundamental change in the trip generation process and was not pursued. The method employed was to simply transform the trip table so that all trips originate and end at network nodes. The trips originating or terminating at a zone centroid were assigned to nodes on the zone boundaries in inverse proportion to the (straight line) distances from the centroid to the nodes. (For further details on how irregular zones, e.g., those not bounded by streets, were handled, see the references.) One important difference between the two methods is illustrated by Figure 2-5. In the zone centroid method trips between adjacent zones often are assigned just to the centroid connectors. This cannot occur with the load node method. Wilson argues that, with respect to adjacent zone trips between corners of the zones, the load node concept is more realistic.



4 Connector Links
8 Street Links
8 Nodes
1 Centroid Node

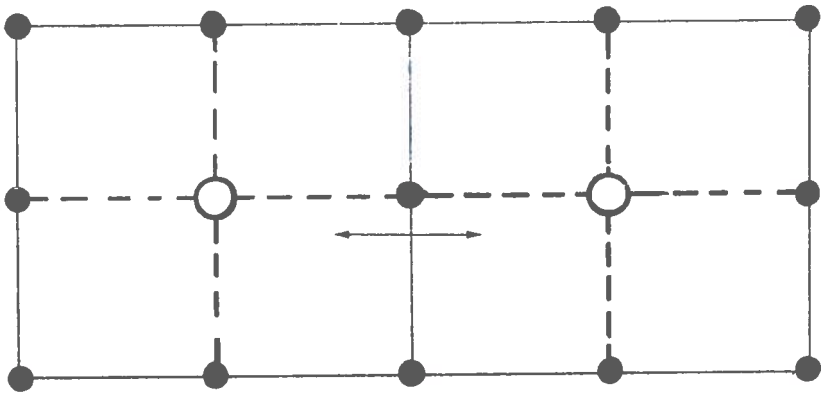
Zone Centroid Concept



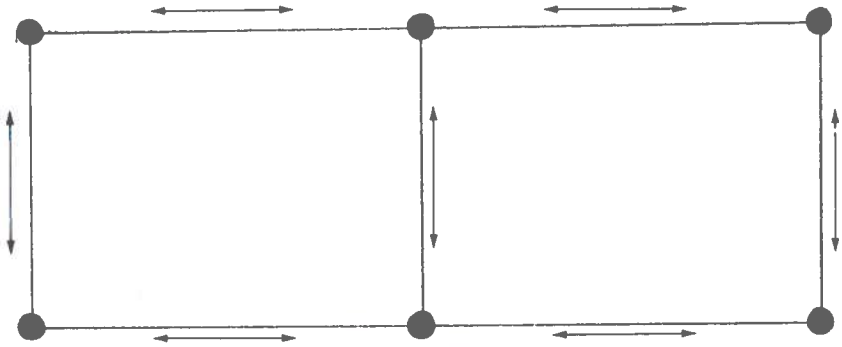
4 Load Nodes
4 Street Links

Load Node Assignment

Figure 2-4: Zone Centroid vs. Load Node



Trips Assigned to Connectors



Trips Assigned to Network Links

Figure 2-5: Adjacent Zone Trip Assignment - Zone Centroid vs. Load Node

The load node concept was applied to the Metropolitan Phoenix network for 1990. The changes in network detail are shown in Table 2-1. Using the formulas (2.1.1) and (2.1.2) we obtain

$$\frac{t_D}{t_A} = \frac{635\ (10000)}{736\ (3994)} \doteq 2.5$$

and

$$\frac{s_D}{s_A} \doteq \frac{10000}{3334} \doteq 3$$

(were the subscripts D and A are as before).

Wilson estimates time savings of 40% - 60% in agreement with the above ratio. To estimate differences between link volumes obtained by the two methods he derives the regression formula (for daily trip assignment)

$$v_D = -.922 + 1.01\ v_A$$

where

v_A = link volume (in thousands) obtained using the aggregated method (load node)

v_D = link volume (in thousands) obtained using the disaggregated method (zone centroid).

	Load-Node Concept	Zone Centroid Concept	% Change
Trip Origins or Destinations	736	635	+15.9
Other Nodes	259	3120	-94.9
Total Nodes	995	3755	-73.4
Total Links	3334	10000	-66.8

Table 2-1: Comparison of Network Detail for Phoenix Network

Thus the load node method yielded an increase of about 900 trips per day on the individual links. As mentioned earlier, this increase is primarily due to adjacent zone trips which are assigned (in Wilson's method) to network links rather than to centroid connectors.

2.5 Mann's Long Trip/Short Trip Aggregation

Bill Mann, Chief of Systems Planning for the COG (Washington, D. C.) Transportation Planning Board, has devised a dynamic aggregation of centroids which shows promise of substantially reducing traffic assignment costs for their network.

The COG zone level network consists of approximately 18000 links, 5000 nodes and 1207 zone centroids. A less detailed district level network of major arterials only is also available. It has 1200 links, 700 nodes and 185 district centroids. Each district is thus composed of about 6 or 7 zones on the average. Mann's aggregation scheme is to superimpose the district centroids on the zone network, assigning "long" trips on shortest paths between district centroids and "short" trips between zone centroids. The cutoff between short trips and long trips is an input parameter (say, 20 minutes).

Figure 2-6 is a flowchart of the procedure. Note that it is an incremental method. In the aggregated loop, the district trip table used is, in effect, the summed zone trip table. The potential savings comes from the fact that just one long path is constructed for all zones in one district to all zones in another district. In addition, in the disaggregated loop, the minimum path trees do not have to "reach" past those zones nearby. It is not possible to apply the formula (2.1.1) except

Inputs - Zone Network, District and Zone Centroids, Zone trip table, T = Parameter defining long trips and L = Parameter indicating when to adjust link travel times.

Long Trip (Aggregated) Loop

- Select District D_k
- Construct Min Time Paths to D_j
- If min time < T save the D_k to D_j trips
- Otherwise load the D_k to D_j trips on min path and zero the entries in the zone trip table
- If k is a multiple of L update link travel times
- k = k + 1

Short Trip (Disaggregated) Loop

- Select Zone Z_k
- Construct Min Paths to Z_j for which trip table entry > 0
- Load Z_k to Z_j trips
- If k multiple of L update link travel times
- k = k + 1

Figure 2-6: Long Trip/Short Trip Aggregation

on an average basis. To solve the problem at the detailed zone level

$$t_D \sim (1207) \cdot (10000).$$

Using Mann's procedure, 185 minimum path trees are constructed on the full network. If about half of these result in actual loadings and if trees built at the zone level need span only 10% of the network (i.e., just to links in nearby districts),

$$t_A \sim (185) \cdot (10000) + (600) \cdot (1000).$$

Therefore

$$\frac{t_D}{t_A} \doteq 5.$$

Space savings do not exist because the entire network is used.

2.6 Aggregation in a Railway Study

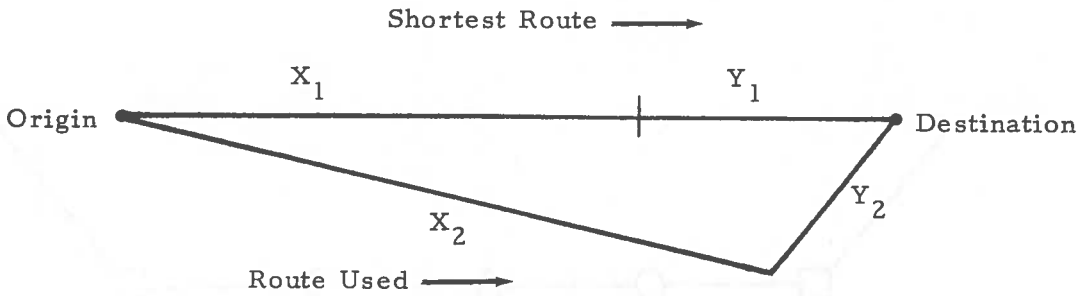
As a final example of the type of aggregation being discussed here, we consider a railway study being conducted at FRA by Tom Bouvé, Raphael Kedar and Carl Fisher.

The study is concerned with measuring the "circuity" of the rail freight system in the United States for the year 1974. Circuity, in this instance, is defined as

$$\text{Circuity} = \frac{\text{Total number of freight ton miles actually traveled}}{\text{Minimum number of freight ton miles needed}} .$$

The denominator in this expression is obtained, of course, by assuming all shipments are along minimum paths. While one might expect that the above ratio would be near one, in actuality it probably is not because of laws regarding how private railroads may bill customers for shipments. As a simple example, suppose a private railroad can send a single shipment over one of two routes. If the allowed charge for the shipment is independent of route, the railroad may use the longer route because their revenue is directly proportional to the fraction of total trip length on track owned by the railroad. See Figure 2-7.

A typical portion of the FRA rail network is shown in Figure 2-8(a). There are three types of nodes: junctions, centroids and dummy nodes. Junctions are points at which two or more tracks intersect. Centroids are nodes which represent origins or destinations (one station or an aggregation of several) and dummy nodes represent only the separation of centroid nodes. In the circuity study, the FRA personnel have aggregated to a "segment level" network as shown in Figure 2-8(b). The links (segments)



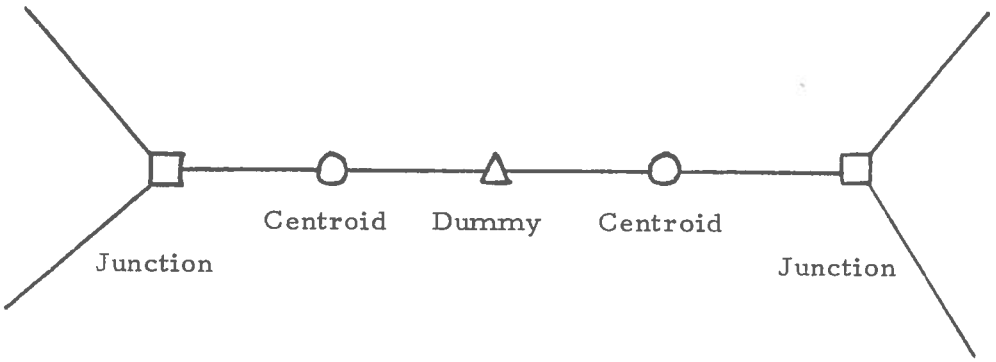
X_1, X_2 = Track owned by railroad X

Y_1, Y_2 = Track owned by railroad Y

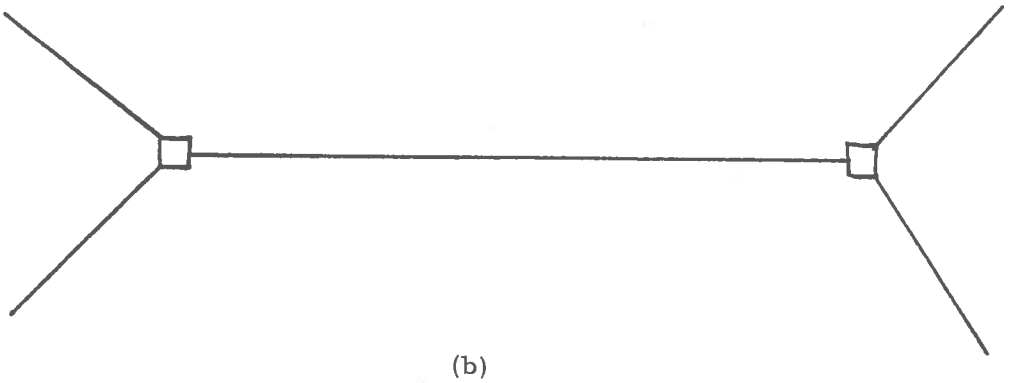
Assume $\frac{X_2}{X_2 + Y_2} > \frac{X_1}{X_1 + Y_1}$

Circuity = $\frac{X_2 + Y_2}{X_1 + Y_1}$

Figure 2-7: Example of Circuity



(a)



(b)

Figure 2-8: Segment Aggregation of FRA Network

of the aggregated network connect junction nodes; there are no centroids or dummy nodes. The trip table which accompanies the segment level network is constructed by aggregation of the centroid demands to the nearest junction nodes (with yards).

This aggregation scheme, then, closely agrees in spirit with that of Dial. Here the "centers" are the junction nodes which have railyards. To estimate the potential savings by the formulas, we have the approximate values

$$m_D = 16000 \qquad \ell_D = 20000$$

$$m_A = 8800 \qquad \ell_A = 4000 \ .$$

Therefore

$$\frac{t_D}{t_A} = \frac{16000 \cdot (20000)}{8800 \cdot (4000)} \doteq 9$$

and

$$\frac{s_D}{s_A} \doteq 5$$

2.7 Summary

The previous sections describe different aggregation schemes employed by different people working on different problems. We submit, however, that a common pattern exists which includes several of the schemes. In this section we will describe this pattern, which we call extraction aggregation, and how it encompasses the schemes of section 2.2 and 2.4.

In all of the methods encountered, there exists an underlying detailed network and associated trip table. These, then, we take as given. For reasons of computational effort and/or space, the network is aggregated prior to determining flows. Therefore, we assume that flowing at least part of the network is the objective. The centroid aggregation pattern is summarized as follows:

1. Given - A network and trip table.
2. Identify - Links of interest for which flows are required.
3. Extract - Some connected subset of the network which contains the links identified in step 2.
4. Select - Nodes of the extracted network which can serve as pseudo centroids.
5. Transfer - The original trip table to the extracted network using the pseudo centroids.
6. Flow - The extracted network using the trip table constructed in step 5.
7. Measure - The flows on the links of interest versus the ideal flows obtainable by flowing the entire network.
8. Use - The flows obtained on the links of interest.

Several comments are in order:

a) The links of interest in step 2 may be all links of the given network.

b) Similarly, the subnetwork of step 3 may be the entire network, whether or not (a) holds.

c) Step 7 was inserted as a needed addition; except for experimental work where flows obtained are compared with those resulting from flowing the entire network, this step does not exist in practice.

Table 2.2 shows how the practices described in sections 2.2 through 2.5 relate to the steps of extraction aggregation. Table 2.3 relates the railway aggregation to the same steps.

Despite the commonality suggested by the extraction aggregation pattern, it is incomplete. Part of this has been mentioned above - the lack of step 7, measurement. Just as important is the need for <u>feedback</u> in the aggregation process (see Figure 2.9). While it is difficult to imagine a perfect convergent process such as suggested by the flow chart of Figure 2.9, this is where mathematical attention should focus. Obviously, many questions can be raised. One such question is suggested by the flow chart: Where should the corrective portion of the loop begin?

Aggregation Steps	JHK Shirley Study	Dial's Origin Aggregation	Wilson's Load Node	Mann's Long Trip Loop	Mann's Short Trip Loop
1. Given Network	Virginia and D.C. Zone Network	8,000 Node Experimental Network	Phoenix Network	COG Zone Network (including District centroid)	Same
2. Identify Links	Figure 2.1	Area A1 of Figure 2.3	All Links	All Links	Same
3. Extract Subnetwork	Figure 2.1	Entire Network	Entire Network	Entire Network	Same
4. Select Pseudo Centroids	Access Points in Figure 2.1	District Centers and Super Centers	Intersection Nodes Surrounding Zones	District Centroids	Zone Centroids
5. Transfer Trip Table	Manually	Nearest Center	Inversely Proportional to Straight Line Distance	Transfer is made Dynamically in Step 6	Trip Table is Fixed
6. Flow Subnetwork	All-or-Nothing (Manual)	Equilibrium Assignment	Capacity Restraint Assignment	Incremental Assignment of Long Trips	Incremental Assignment of Short Trips

Table 2-2: Aggregation in Urban Highway Studies

Aggregation Steps	FRA Circuity Study
1. Given Network	FRA Network
2. Identify Links of Interest	All Links
3. Extract Subnetwork	Entire Network (Segments)
4. Select Pseudo Centroids	Nodes with Yards
5. Transfer Trip Table	Near Yard
6. Flow Subnetwork	All-or-Nothing

Table 2-3: Aggregation in a Railway Study

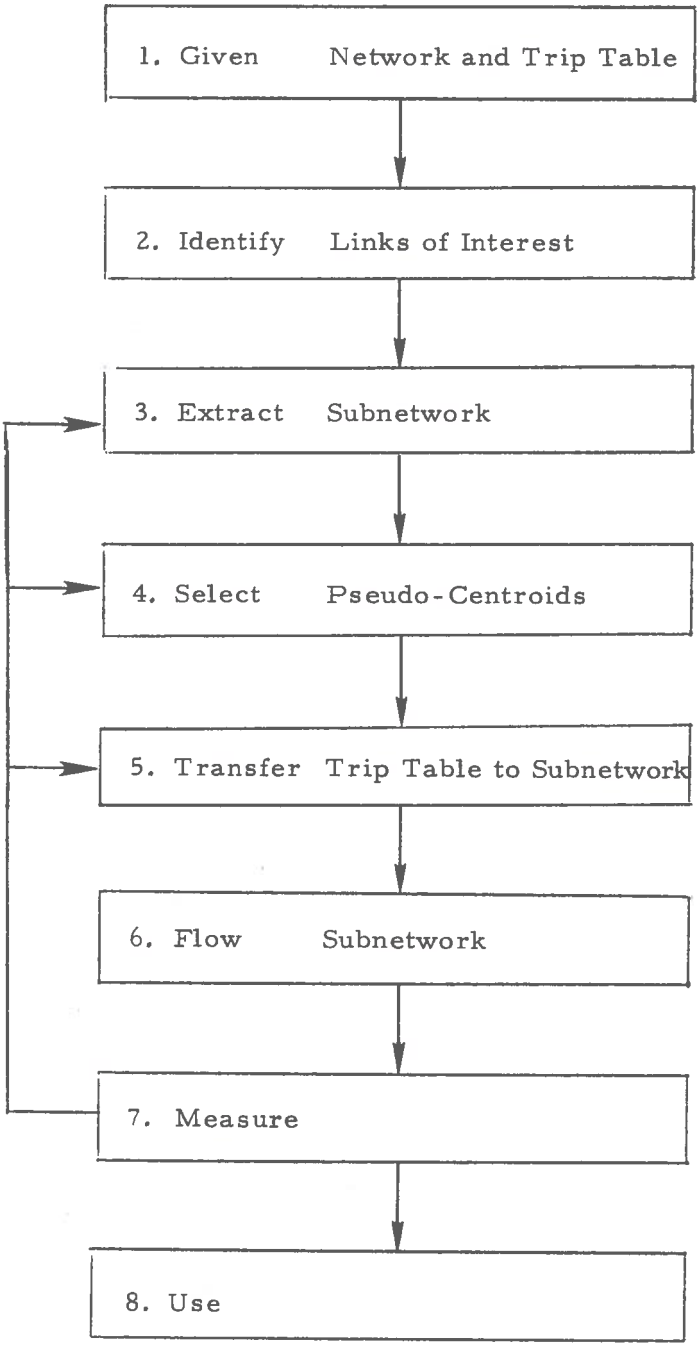


Figure 2-9. Extraction Aggregation with Feedback

APPENDIX

CRITERIA FOR PROTOTYPE STUDY

Harold W. Kuhn

In our search for an application, we formulated a set of criteria that we considered to be most important for the choice of this study. By having these criteria explicitly stated, we brought a degree of consistency to our investigation of the applications that were proposed. These were stated as a set of questions to be put regarding each candidate for the study. These questions were organized into five subareas. Some representative answers from our preliminary investigations have been given to make the intent of the questions clearer.

1. Decision Goals

   a. Does the network application have one or more clearly defined decision goals?

   Example: The various "dedicated-lane" or "special-lane" programs proposed or underway have as their objective the promotion of bus travel or car-pooling to cut urban traffic and air pollution.

   b. Can the elements that enter these decision goals be quantified?

   Example: Some measures that have been suggested for the "special-lane" programs are: vehicle-miles on the highway, bus ridership, number of accidents, travel times, smog abatement.

2. Mathematical Model

a. Does one of the transportation network models (e.g., Hitchcock-Koopmans problem, shortest path, traffic assignment, equilibrium) appear as an essential part of evaluating the decision goals?

Example: In almost all of the applications considered, some traffic assignment model is used to describe or predict traffic flow under different conditions. For the "special-lane" problem, the measures of the decision goals can be connected directly to these flows.

b. Is computational experience available for this network model on this application?

Example: For the models of Debanne at the University of Ottawa used to study pipeline expansion, considerable computational experience seems to be available.

3. Network

a. Does a non-trivial network appear as an essential feature of the application?

Example: Of course, the answer to this question must be affirmative. However, it may not be as simple as it seems. For example, the inland waterway network of the Mississippi is a tree with 3 extra links (one in Kentucky and two in the delta). However, the traffic assignment problem for this "trivial" network seems difficult to handle satisfactorily.

As a second example, the Santa Monica Freeway, which is the subject of a "special-lane" application is clearly a trivial network but is not when feeder roads (or some abstraction) is adjoined.

b. Is the network and the relevant characteristics of its elements available? Is it currently coded or easily coded for computer treatment? Are the data on the elements readily available for computer treatment?

Example: Various rail networks seem readily available with demand data. The inland waterway networks are also available with very complete data.

c. What is the size of the network?

Example: TSC has given the following estimates of the networks involved in the study of intercity freight systems: water, 400 links; rail, 3,500 links (NBS aggregated mainline); highway, 4,300 links (Federal aid highways). The number of transportation zones seems to be of the order of 500.

4. Aggregation

a. Has aggregation been used in the formulation or previous analysis of the model?

Example: Most rail networks have urban areas highly aggregated and some station aggregation. The Ottawa pipeline model of Debanné was highly aggregated before the equilibrium calculation was done. Most airline analysis has been done on quite small extracted (and abstracted) networks.

b. If the model has been analyzed in disaggregate form, is there the possibility of aggregation?

Example: We believe that aggregation is appropriate for macro goals (such as total vehicle miles) but less so for micro goals (such as link flows).

5. Future Interest and Implementation

a. Is this an important application with the prospect of continuing interest in the future?

Example: Many of the applications involve estimating the effect of various regulatory configurations, a subject which is clearly of continuing interest. Some applications, such as rail line abandonment, seem to have had a flurry of activity then interest waned. Another class of applications which promise an affirmative answer are the environmental impact studies.

b. What is the time scale of desired implementation of the analysis?

c. Is there a likely prospect that aggregation techniques will facilitate implementation of the analysis?

Example: This question is likely to have an affirmative answer in those applications for which aggregation will make possible the investigation of a wide range of parameters and configurations.

References

1. S. E. Dreyfus, "An Appraisal of Some Shortest-Path Algorithms," Operations Research, 17, 3 (May-June 1969), pp. 395-412.
2. JHK and Associates, "Evaluation of Alternative Traffic Operation Plans for the Commuter Lanes on the Shirley Highway in Virginia," April 1976, Contract DOT-FH-11-8242.
3. J. T. McQueen, D. M. Levinsohn, R. Waksman, G. K. Miller, "Evaluation of the Shirley Highway Express-Bus-On Freeway Demonstration Project -- Final Report," August 1975, Prepared for UMTA by Technical Analysis Division, Nat. Bureau of Standards.
4. E. M. Wilson, "The Load Node Concept of Traffic Assignment for Urban Areas," Ph. D. Dissertation, Arizona State University, September 1972.
5. E. M. Wilson, J. S. Matthias, M. J. Betz, "A Traffic Assignment Planning Model: The Load-Node Concept," Transportation Research, Vol. 8 (1974), pp. 75-84.

3. Computational Study of Extraction Aggregation

3.1 Introduction

As part of contract DOT-TSC-1232, MATHTECH has conducted a computational study of the aggregation scheme described in Chapter 2 (Figure 2-9). The objective of this study was to provide insight to the following questions:

(a) How good are aggregation practices? That is, how much faith can be placed in the results of studies where aggregation is done in an ad hoc and heuristic manner?

(b) Are some parts of the aggregation process more critical than others with respect to yielding correct solutions?

(c) Can the tools of nonlinear programming (Chapter 4) be employed in the MEASURE step to bound aggregation error?

(d) Are actual computational savings through aggregation in agreement with the formulas of Chapter 2?

No single study can answer these questions completely, of course, but to make the conclusions as believable as possible, MATHTECH and the technical monitors agreed that the study should center around a large, realistic network and that it should capture as many elements as possible of an actual aggregation practice. For these reasons the Shirley study described in Section 2.2 was chosen as prototype. It was not possible, or

necessary, to duplicate the JHK aggregation exactly. At every step (Figure 2-9) of the process it was essential that exceptions to their scheme be taken. There were many reasons for these exceptions, but the two most prevalent were (a) computational feasibility and (b) missing or incomplete data.

In normal transportation studies the calculated flows of a network model are measured (calibrated) against observed network flows. For this study we have chosen as a benchmark the flows obtained on the given (disaggregated) network using the computer code TRAFFIC [1] developed at the University of Montreal. This code calculates flows for a given network and trip table according to Wardrop's user-equilibrium principle [2]. Implicitly then, the aggregation scheme yields "good" results if the resulting flows on the links of interest (Figure 2-9) agree closely with the benchmark link flows. While this obvious criteria may be sufficient to claim a scheme is "good," it is not necessary if, for example, all that is desired are gross network measures (total vehicle miles, etc.) as might be required in a pollution study.

In the following sections descriptions of the implemented aggregation schemes are given. The final section summarizes conclusions of the study.

3.2 Given Network and Trip Table

The Washington Council of Governments (COG) Transportation Planning Board kindly provided their zone-level highway network for 1972. The network has approximately 19,000 links and 6,000 nodes including 1,207 centroids. While this could have been the chosen network, it would have

required excessive computer storage and costs in establishing benchmark flows using TRAFFIC. The reduction to more manageable size (itself an aggregation) was accomplished by deletion of all COG links for which both origin and destination nodes lie in the state of Maryland. The resulting Virginia plus D. C. network, which was chosen as the given network, had 9386 links and 3027 nodes including 700 centroids.

COG also provided a trip table based on 1968 data which represented home-to-work trips in the D. C. metropolitan area. Components of the trip table for trips from or to Maryland centroids were deleted and the remaining ones were "factored" to reflect all peak hour trips in the Virginia-D. C. area. This adjustment of the data was based on a COG study [3] on estimating peak hour traffic. The resulting trip table had 109,706 trips which, according to COG experts, approximated the number of peak hour trips in the area.

(In the JHK study the trip table for the subnetwork was constructed from data in a previous NBS study of the Shirley Highway. Their data base indicated 30,000 vehicles crossing a cordon line in northern Virginia. From this a trip table of 10,000 trips, which were assumed to use the major Virginia arteries, was developed. While this was made available to MATHTECH in both aggregated and disaggregated form, it was not used because to do so would involve making an unwarranted assumption that <u>all</u> trips in the disaggregated trip table used the subnetwork. As explained in the summary, the computational results show that the aggregation results are very sensitive to this transfer step.)

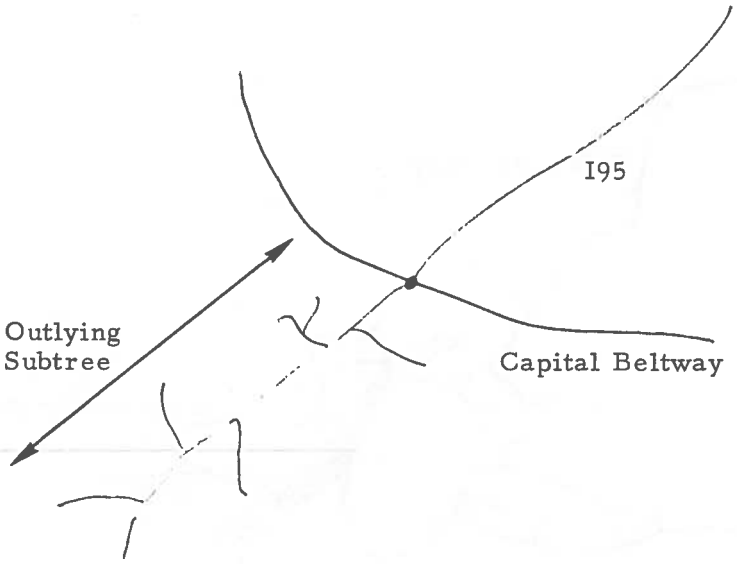
3.3 Links of Interest and Subnetwork

In the Shirley study the links of interest were the major arteries of Virginia. These and major streets of D. C. formed the extracted subnetwork. A precise description was provided by JHK.

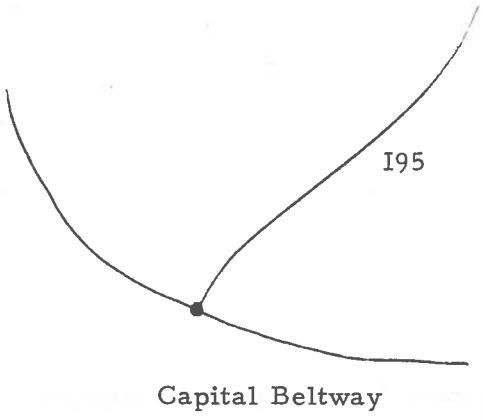
After studying the JHK subnetwork, we decided that, for the purposes of the MATHTECH research, it was unnecessarily large, and it was incomplete. To reduce the size, "outlying subtrees" were eliminated from the JHK subnetwork. This term is best explained graphically. In Figure 3-1(a) a sketch illustrates the JHK subnetwork detail on I95 south of the Capital Beltway. This portion of the subnetwork is a subtree which was eliminated in the MATHTECH subnetwork [see part (b) of the figure]. The justification, other than computational savings, is that on the subtree the paths between nodes are unique.

The JHK subnetwork was incomplete in the sense that certain links which seemed absolutely essential were omitted. The most obvious example is the Theodore Roosevelt Memorial Bridge, which JHK considered to be unimportant in the Shirley study, but which seemed necessary for the more general MATHTECH aggregation research. This issue was resolved by including any links that were part of shortest (uncongested) time paths between the nodes of the subnetwork.

In summary, the extracted subnetwork was guided by the JHK subnetwork but modified substantially in detail as judgement dictated. The result was a network of 228 nodes and 483 links. (The subnetwork did not include any centroids or centroid connectors of the given network.) A rough sketch, with major arteries identified, is in Figure 3-2.



(a)



(b)

Figure 3-1: Elimination of an Outlying Subtree

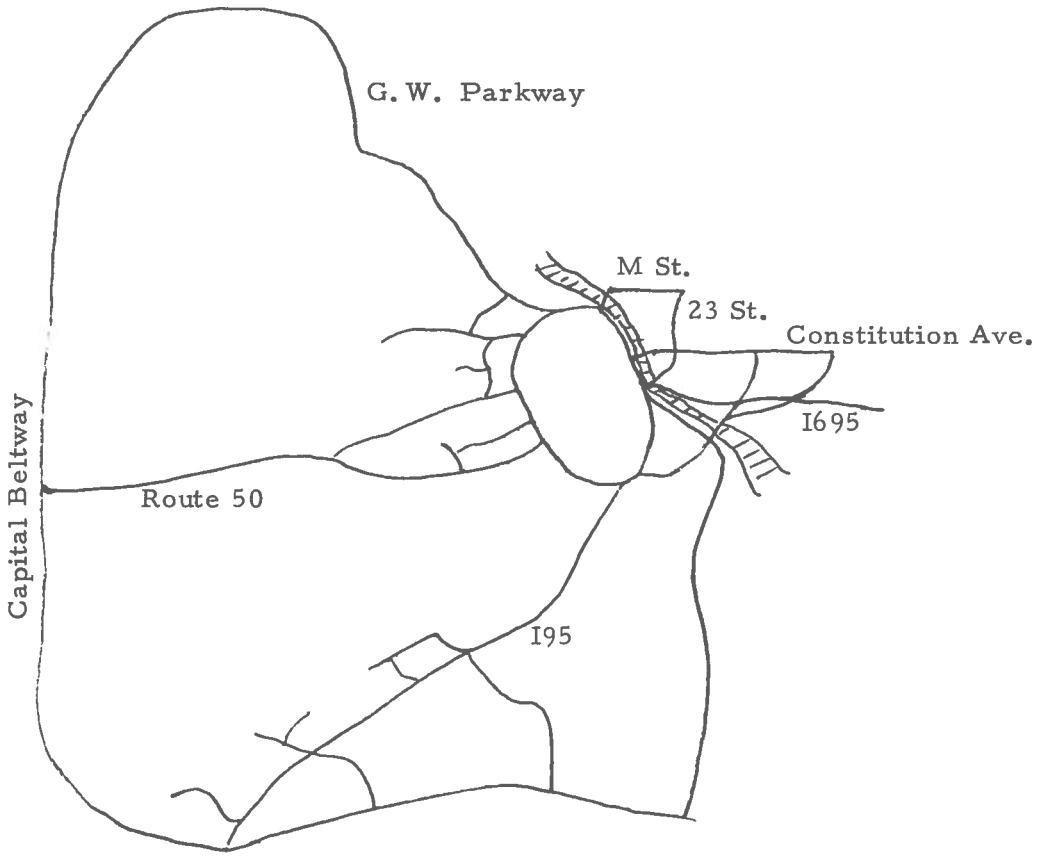


Figure 3-2: Sketch of Subnetwork

3.4 Selection of Pseudo-Centroids

The JHK study had 42 nodes of the subnetwork designated as pseudo-centroids. In the MATHTECH subnetwork there were only 28 because of the elimination of outlying subtrees described in the previous section. In Figure 3-1(a), for example, the subtree had seven pseudo-centroids, one at each "tip" of the subtree. With elimination of this subtree, the single node shown at the intersection of I95 and the Beltway became a pseudo-centroid.

3.5 Transfer of the Trip Table

The construction of an aggregated trip table from the disaggregated trip table is, we feel, the most critical step of the aggregation process if the subnetwork and pseudo-centroids are fixed. As discussed in Section 3.2, this problem was solved in the Shirley study by first making the decision of what trips would use the subnetwork, independent of the flowing of the subnetwork. Thus the two trip tables totaled the same. Similarly, the other aggregation schemes of Chapter 2 were involved with the transfer of an entire trip table. More generally, the aggregation process requires some portion of the full trip table transferred to the subnetwork.

In the computational study a simple automated heuristic was employed for the trip transfer. First, each original centroid i was assigned to one of the 28 pseudo-centroids, A_i, based on the JHK aggregation. A minimum (uncongested travel time) path tree was built from centroid i until A_i became labeled. If centroid j was part of the minimum path tree so constructed, then the i to j trips were <u>not</u> included in the aggregated trip table. Otherwise the i to j trips were

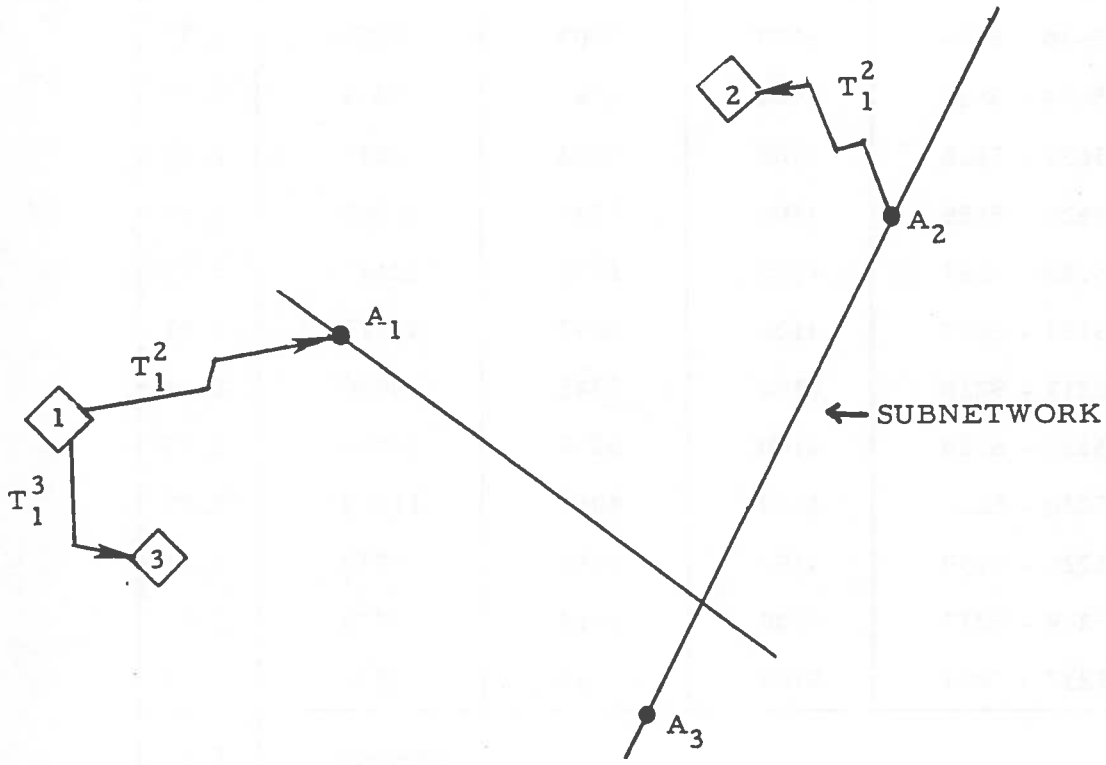
assumed to originate at A_i and terminate at A_j. The process was repeated for all i. Thus trips became part of the aggregated trip table if and only if the uncongested travel time from an origin to its (assigned) pseudo-centroid was less than the travel time to the destination.

3.6 Flowing the Subnetwork and Measuring

The TRAFFIC code was used to flow the subnetwork and obtain approximate user-equilibrium flows.

To measure the results using the theory of Chapter 4, it was necessary to construct a flow pattern for the entire given network. This was done by assigning trips that were not part of the aggregated trip table to the minimum paths constructed in the transfer step (see the previous section). Also, trips of the aggregated network were assigned to the minimum paths between the centroids and the pseudo-centroids as defined in the transfer step. This process of constructing a full set of flows for the entire network is referred to as "lifting" the aggregated solution, or as "disaggregating" the solution. Figure 3-3 depicts the lifting process. Trips T_1^2 and T_1^3, from centroid 1 to centroids 2 and 3, are assigned to the links of the minimum path tree built from centroid 1. Also, the T_1^2 trips were assigned to the minimum path from A_2 to centroid 2.

The results of the aggregation as compared with the benchmark were, by any measure, very poor. As an example, Table 3-1 shows the flows on certain key links (all on the Shirley highway) obtained by both methods. The link volumes are off by factors ranging from 1.42 to 2.08, the average being 1.86.



A_i = Pseudo-centroid assigned to centroid i.

Figure 3-3: Lifting of Flows to Given Network

Shirley Highway [1] (COG Node Nos.)	Assigned "Capacity" [2] C	Benchmark Volume B	Aggregation Volume A	Ratio A/B
6001 - 6002	3900	3625	5347	1.47
6002 - 5640	3900	3603	6184	1.72
5640 - 5639	5300	3603	6184	1.72
5639 - 5631	3900	4287	8444	1.97
5631 - 5626	3900	5122	7276	1.42
5626 - 5185	4100	5934	11062	1.86
5185 - 5187	4100	6632	12113	1.83
5187 - 5217	4100	6657	13412	2.01
5217 - 5218	4100	6745	14056	2.08
5218 - 5220	4100	8035	18059	2.25
5220 - 5226	4100	4057	11329	2.79
5226 - 5239	4100	5514	9770	1.77
5239 - 5237	4100	5514	9770	1.77
5237 - 2903	5700	8009	11539	1.44
			Average	1.86

Notes: 1. Node 6001 is at Capital Beltway and Node 2903 is the north end of the 14th Street Bridge.

2. The "capacity" C is the value which appears in the COG volume delay formula $T = T_0 (1 + 0.5 (V/C)^4)$, where

T = Link Travel Time

T_0 = Uncongested Link Travel Time

V = Link volume.

COG practice is to set this value at approximately 1300 vehicles per lane per hour.

Table 3-1: Flows on Shirley Highway Links (Northbound)

(Column one of this table contains the COG node numbers [from / to] for each link and column two is the link capacity constant obtained from the network data.)

Other comparisons are given in Table 3-2. It is well-known that the user-equilibrium link flows can be obtained by solving a nonlinear programming problem (the code TRAFFIC uses this approach). In Chapter 4 we show how to compute the duality gap for this problem and that it leads to an intuitively appealing measure of how good any feasible set of flows is with respect to the user-equilibrium objective. Table 3-2 shows the results obtained in the benchmark run and from the aggregation flows. As before, the aggregation column shows poor results. The negative lower bound is effectively worthless since zero is an obvious lower bound.

(Table 3-2 also reflects a weakness in the lower bound. The benchmark run shows a lower bound of 30% relative error when, in fact, the objective value is probably within 5% to 10% of the true optimum. This theoretical point is discussed in Chapter 4.)

	Benchmark	Aggregation
NLP Objective Value	0.201×10^9	1.3669×10^9
Lower Bound	0.143×10^9	-4.27×10^9 *
Relative Duality Gap	30%	412% *

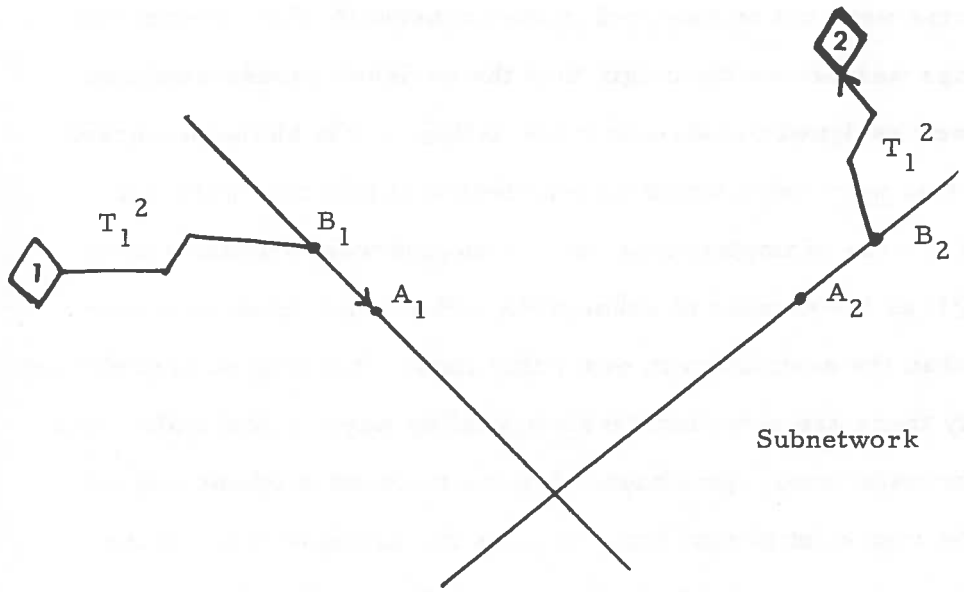
*Zero is a trivial lower bound.

Table 3-2: NLP Comparisons of Benchmark with Aggregation

3.7 Additional Heuristics for the Transfer Step

After obtaining the results described in the previous sections, it was decided to modify the transfer procedure to try to improve them. Of particular interest was whether improvement could be obtained by transferring fewer trips from the original trip table (step 5 of Figure 2-9). It can, of course, be argued that extraction of the subnetwork and/or selection of pseudo-centroids (Steps 3 and 4) are at least as important as the transfer step. We agree that this is true and feel that much research needs to be done on these steps and their interrelationships.

For the current project, we have limited our computational experimentation to simple modifications of Steps 4 and 5. For the first of these, selection of the pseudo-centroids, examination of the data revealed that in construction of the minimum path trees (Figure 3-3), it was often the case that the assigned pseudo-centroid A_i for centroid i was not the first labeled node of the subnetwork. Figure 3-4 demonstrates that the minimum path from centroid 1 to A_1 might contain links which are part of the subnetwork. Thus the trips originating at centroid 1 are possibly doubly assigned to the subnetwork, once in flowing it, and again in lifting flows in the Measure step. While this is not necessarily incorrect, depending on directions of flow, it seemed reasonable to alter the pseudo-centroids so that this would not occur. This was done by defining the pseudo-centroid for centroid i to be the node of the subnetwork nearest (in the sense of uncongested travel time) to centroid i. When this was done the number of pseudo-centroids increased to 142 from the 28 of the prior method.



A_i = Pseudo-Centroid Assigned to Centroid i

B_i = "Nearest" Subnetwork Node to Centroid i

Figure 3-4: Minimum Path Tree on the Subnetwork

The second modification concerned the transfer of fewer trips from the original trip table to the subnetwork. Recall that in the original scheme, trips were not transferred to the subnetwork if the destination of those trips was nearer the origin than the assigned pseudo-centroid. Given the new assigned pseudo-centroids defined in the above paragraph, it is clear that more trips would be transferred if this rule were not altered. For ease of implementation, the method was to define a parameter $k(\geq 1)$ as the number of subnetwork nodes which must be nearer the origin than the destination in order that those trips may be transferred. Undoubtedly there are more intuitively appealing ways to accomplish this important transfer step. (In Chapter 5 some rigorous methods are outlined.) It is important to note that, because the lifting of flows to the full network was not changed from the method described in Section 3-6, subnetwork links might be included in the alternate routes.

Changing only the value of k, computer runs were made using the same data base as before. For k = 40, 100, 150 the subnetwork trip tables contained 69,497, 46,681, and 31,943 trips, corresponding to 63%, 43%, and 29% of the total.

The results of these runs are summarized in Tables 3-3 and 3-4 which compare with Tables 3-1 and 3-2. In addition we have Tables 3-5 and 3-6 which show the results of using the lifted flows from all four aggregation schemes as an "advance" start in the TRAFFIC code. Normally this code begins with an "all-or-nothing" assignment, i.e., all trips by minimum time paths without regard to congestion. It has the provision, however,

Shirley Highway (COG Node Nos)	Benchmark Volume B	Aggregation Volumes			Ratios		
		k = 40 A_1	k = 100 A_2	k = 150 A_3	A_1/B	A_2/B	A_3/B
6001-6002	3625	5609	6221	5713	1.55	1.72	1.58
6002-5640	3603	5567	6242	5866	1.55	1.73	1.63
5640-5639	3603	5567	6242	5866	1.55	1.73	1.63
5639-5631	4287	7484	7290	6992	1.75	1.70	1.63
5631-5626	5122	7278	7825	8319	1.42	1.53	1.62
5626-5185	5934	11355	11068	10631	1.91	1.87	1.79
5185-5187	6632	12108	11844	11581	1.83	1.79	1.75
5187-5217	6657	13838	13197	12604	2.08	1.98	1.89
5217-5218	6745	14242	13550	12931	2.11	2.01	1.92
5218-5220	8035	13631	13297	12558	1.70	1.65	1.56
5220-5226	4057	8256	7667	7276	2.04	1.89	1.79
5226-5239	5514	7967	7473	7175	1.44	1.36	1.30
5239-5237	5514	7967	7473	7175	1.44	1.36	1.30
5237-2903	8009	12070	10151	9410	1.51	1.27	1.17
				Averages	1.71	1.69	1.61

Note: Node 6001 is at Capital Beltway and Node 2903 is the north end of the 14th Street Bridge

Table 3-3: Flows on Shirley Highway Links (Northbound) (Continued)

	Benchmark	Aggregation Results		
		k = 40	k = 100	k = 150
NLP Objective Value	0.201×10^9	0.645×10^9	0.355×10^9	0.300×10^9
Lower Bound	0.143×10^9	-1.545×10^9 *	-0.442×10^9 *	-0.255×10^9*
Relative Duality Gap	30%	340% *	225% *	185% *

* Zero is a trivial lower bound

Table 3-4: NLP Comparisons of Benchmark with Other Aggregations

	TRAFFIC	Original Aggregation	Other Aggregations		
			k = 40	k = 100	k = 150
Initial Assignment	0.371	1.367	0.645	0.356	0.300
Iteration 1	0.239	0.465	0.343	0.253	0.222
Iteration 2	0.219	0.377	0.275	0.226	0.213

Note: Table contains NLP objective values/10^9

Table 3-5: Aggregation Solutions as Advance Start for TRAFFIC

	Benchmark	Three TRAFFIC Iterations	Original Aggregation	Other Aggregations		
				k = 40	k = 100	k = 150
			Plus 2 TRAFFIC Iterations			
Total Vehicle Hours	44,860	61,703	160,583	92,990	51,061	51,573
Total Vehicle Miles	846,315	847,288	1,074,638	967,752	911,048	881,313
Average MPH	18.87	13.73	6.7	10.4	14.4	17.08

Table 3-6: Gross Measures -- Aggregation versus Benchmark

of starting from any assignment of trips. Since the aggregation schemes employed can yield an assignment at slightly less cost than the all-or-nothing assignment, we ran TRAFFIC two iterations starting with the aggregated solutions. These results are summarized in Tables 3-5 and 3-6. The most impressive results came from gross measures of the network flows. By gross measures we mean, in particular, Total Vehicle Hours and Total Vehicle Miles which can be translated, of course, to Average Miles per Hour. In studies of, say, energy consumption or environmental impact, such measures may be the only information required. In Table 3-6, the last aggregation of 31,934 trips produced very good results when compared with the benchmark results.

3.8 Conclusions

The evidence presented by these computational experiments leads us to the following conclusions (cf. the questions raised in Section 3-1):

(a) Our experiments are contrary to the informally reported experimental results of Dial and Mann. (They are, however, consistent with the predictions of Chan, et al. [4].) Since the crucial difference in the aggregation scheme is that the subnetwork does not flow all trips from the original trip table we must conclude that the extraction of a small subnetwork can only work well if planners have *a priori* information regarding the trip table transfer. The fact that our heuristics got better as the number of trips became fewer supports the view. In fact, it is encouraging that the final flowing of 31,934 trips to equilibrium on the subnetwork* resulted in a significantly better solution than the initial all-or-nothing solution of TRAFFIC (Table 3-5). This is the first solid

*Recall that the total number of trips is greater than 31,934, by the structure of the heuristic.

evidence that an automated aggregation scheme can be made to work, especially since the heuristic employed is quite crude. More refined methods such as those discussed in Chapter 4 can be expected to produce very good results.

(b) As we have defined extraction aggregation, the critical parts are (i) the defining of the subnetwork, (ii) selection of pseudo-centroids, and (iii) transfer of the trip table. Whether an algorithm can be developed that allows all three of these elements to vary is an open (and difficult) question. We feel that the correct approach is to first assume the subnetwork and pseudo-centroids fixed and attempt to make precise the transfer of trips and, at the same time, refine the tools of the Measure step so that a good solution can be recognized.

(c) The "duality gap" error bound used in the Measure step has proven to be a weak measure of aggregation error. The primary reason for this is that if, in the lifted solution, some links are heavily congested then the duality gap will be large unless those link flows are very near optimal. In other words, a lifted solution may be almost correct (with respect to the NLP objective value) but if a few congested links are not near their correct flow values, the duality gap will be artificially large. This points out, as predicted in earlier MATHTECH research, that the lifting of flows in the Measure step is absolutely critical. Crude lifting rules such as those we have employed will tend to overly congest some links and thereby yield a weak bound.

Despite the weakness of the bound, we should note that it did improve with the improvement of the aggregation.

(d) The computational savings through aggregation are considerable, and the formulas of Chapter 2 give estimates of the magnitude. In establishing the benchmark, each iteration of TRAFFIC required 221 seconds of 360/91 CPU time. The original aggregation scheme led to a subnetwork assignment that required 0.59 CPU seconds per iteration, also using TRAFFIC. Thus the ratio was 375. From the formula 2.1.1 we obtain the estimated ratio

$$\frac{t_D}{t_A} = \frac{(585)(9386)}{(28)(483)} = 406 \quad .$$

(The number 585 is used for m_D because that many centroids of the 700 were actual sources of trips.) Similarly, for the other three aggregations the estimated ratio is

$$\frac{t_D}{t_A} = \frac{(585)(9386)}{(142)(483)} = 80$$

and the actual ratio was 221/2.18 = 101.

These are only per iteration estimates based on the fundamental minimum path calculations. Of more importance are the rough estimates we obtained for the costs of all steps of the aggregation versus the benchmark run. These are summarized by the approximate formulae

$\doteq$	Total cost of aggregated assignment
=	Cost of transfer + subnetwork flowing
$\doteq$	Cost of one iteration of TRAFFIC

and

	Cost of measuring lifted flows
$\doteq$	Cost of one iteration of TRAFFIC.

Hence a good aggregation assignment plus measurement to prove how good it is can be accomplished for the cost of two iterations of TRAFFIC. Since ten to fifteen TRAFFIC iterations (sometimes more) are required to achieve good flows for the given problem, the potential savings are considerable. Furthermore, many transportation planning groups are currently using just three or four iterations to obtain their assignments, the constraint being that one such run can cost several hundred dollars for an urban network.

References

1. Nguyen, S. and James, L., "TRAFFIC - An Equilibrium Traffic Assignment Program," Centre de Recherche sur les Transports, Publication #17, University of Montreal, March 1975.

2. Potts, R. B. and Oliver, R. M., Flows in Transportation Networks, Academic Press, 1972.

3. Mann, W. S., "Estimating Peak Hour Automobile Travel," Technical Note #4, Transportation Planning Board, Metropolitan Washington Council of Governments, 1976.

4. Chan, Y., Follansbee, K. G., Manheim, M. L., and Mamford, J. R., "Aggregation in Transport Networks: An Application of Hierarchial Structure," Volume VIII of Search and Choice in Transport Systems Planning, Dept. of Civil Engineering, M.I.T., Cambridge, Mass., 1968.

4. Bounding Error in the Traffic Assignment Problem

4.1 Introduction

In this chapter we show that various error measures proposed for the user-equilibrium assignment problem are equivalent. Geometrical interpretations are given along with some numerical examples. These results lead to a proposed method for improving bounds and to an alternative mathematical programming formulation of the problem.

4.2 Problem Formulation and Notation

The user equilibrium model of traffic flow for a given network is equivalent to the following mathematical programming problem [1].

$$(P) \qquad \min_{x} \sum_{kj} \int_0^{x_{kj}} c_{kj}(t)\, dt$$

$$\text{s.t.} \qquad B x^i = b_i \qquad \text{(flow conservation)}$$

$$x^i \geq 0 \qquad i \in D$$

where $(\cdot)_{kj}$ - the subscript kj is associated with the directed arc from node k to node j.

x^i_{kj} = flow to destination i on arc kj

$x_{kj} = \sum_i x^i_{kj}$ = total flow on arc kj

B = node - arc incidence matrix

x = vector of all x_{kj}^i

x^i = vector of all flows to destination i

$c_{kj}(x_{kj})$ = time per unit to traverse arc kj when the flow is x_{kj} - assumed convex

O = set of origin nodes

D = set of destination nodes

b_i = trip vector for destination i

T_k^i = trip table entry of required flow, $k \in O$, $i \in D$

T^i = all trips to destination i

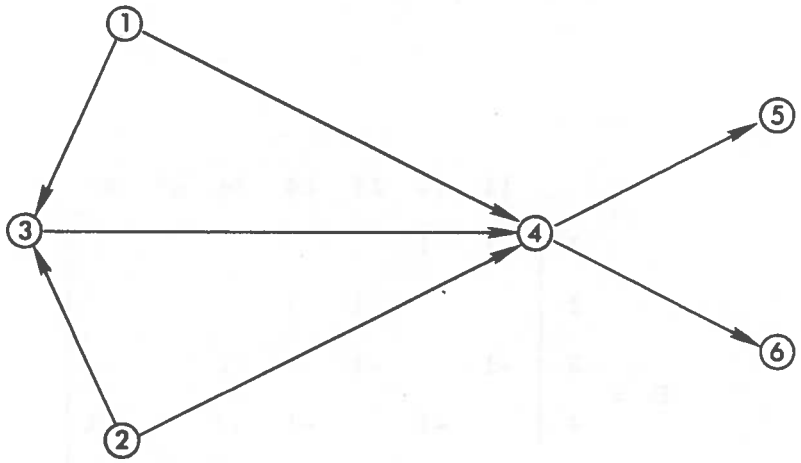
To illustrate, consider Figure 4-1. In this case problem (P) becomes

$$\text{Min} \quad \sum_{kj} \int_0^{x_{kj}} c_{kj}(t)\, dt$$

$$\text{s.t.} \quad B x^5 = b_5$$

$$B x^6 = b_6$$

$$x^5, x^6 \geq 0$$



Trip Table

	5	6
1	T_1^5	T_1^6
2	T_2^5	T_2^6

Figure 4-1: Six Node Network

where

$$kj \in \{13,\ 14,\ 23,\ 24,\ 34,\ 45,\ 46\} = \text{all arcs}$$

$$O = \{1, 2\} \qquad D = \{5, 6\}$$

$$B = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{array} \begin{array}{c} \begin{array}{ccccccc} 13 & 14 & 23 & 24 & 34 & 45 & 46 \end{array} \\ \begin{bmatrix} 1 & 1 & & & & & \\ & & 1 & 1 & & & \\ -1 & & -1 & & 1 & & \\ & -1 & & -1 & -1 & 1 & 1 \\ & & & & & -1 & \\ & & & & & & -1 \end{bmatrix} \end{array}$$

$$b_5 = \begin{bmatrix} T_1^5 \\ T_2^5 \\ 0 \\ 0 \\ -T^5 \\ 0 \end{bmatrix} \qquad b_6 = \begin{bmatrix} T_1^6 \\ T_2^6 \\ 0 \\ 0 \\ 0 \\ -T^6 \end{bmatrix}$$

(Note that a slight reduction in problem size results from $x_{46}^5 = 0$ and $x_{45}^6 = 0$ in the flow conservation equations. For simplicity of notation we leave these variables in the problem.)

For ease of exposition we will use the more general formulation of minimizing a continuously differentiable objective function subject to linear constraints:

$$\text{(C)} \qquad \text{Min} \quad f(x)$$

$$\text{s.t.} \quad Ax = b$$

$$x \geq 0.$$

To relate this to problem (P), make the following identifications:

A = a block diagonal matrix with blocks B,

b = vector of b_i,

x = vector of all x_{kj}^i,

$$f(x) = \sum_{kj} \int_0^{x_{kj}} c_{kj}(t)\, dt.$$

Furthermore note that the gradient vector of f, $\nabla f(x)$, has components $c_{kj}(x_{kj})$. Define for any x feasible to (P) the total network time for all users:

$$F(x) = \sum_{kj} c_{kj}(x_{kj})\, x_{kj}$$

and for $k \in 0$, $i \in D$

$s_k^i(x)$ = minimum time over all paths from k to i.

Finally, define

$$S(x) = \sum_{i \in D} \sum_{k \in 0} s_k^i(x)\, T_k^i$$

= Total time if all trips are made on minimum paths (with respect to x).

4.3 Background

Although the theoretical solution x^* to problem (P) can be shown to exist, computational procedures such as TRAFFIC calculate only an approximate solution x because of the excessive size and running time of real problems. Therefore it is important to compute a lower bound on the objective value of (P) in order to determine relative error.

By definition a set of flows x is at user-equilibrium if, for each OD pair, the path costs of utilized paths are equal and not greater than the cost of paths not used. Therefore the total cost of all shortest paths must equal the total cost of all paths (equivalently the total cost of all arcs). In the notation of the previous section,

$$F(x^*) = S(x^*). \qquad (4\text{-}1)$$

The equation appears first in the paper by Beckman [2] who proved that the user-equilibrium problem is equivalent to (P). Later, Murchland [3] used conjugate duality theory to prove that there is a dual problem to (P)

which has an optimal objective value equal to the optimal objective value of (P). Furthermore, for any x satisfying the constraints of (P) (i.e., feasible to (P)) the objective value, f(x), is bounded below by the corresponding dual objective value. The difference between these two values (expressed solely in terms of x) is known as the duality gap, G(x). Murchland proves that

$$G(x) = F(x) - S(x) \geq 0 \qquad (4\text{-}2)$$

hence $G(x^*) = 0$ if and only if x^* solves (P).

The function G(x) has obvious intuitive appeal, and, in fact, has been recognized for some time by designers of computer codes as a measure of how far x is from optimal. For example, it appears as an output of the UMTA program UROAD [4].

Another common output from programs such as UROAD and TRAFFIC is the "rate of change" of f(x) at each iteration. This negative quantity approaches zero (nonmonotonically) as x approaches x^*, and is the directional derivative of f in the direction of movement from one value of x to the next. In addition, these codes, based on the Frank-Wolfe algorithm of nonlinear programming, print at each iteration, a lower bound on the value of the objective function which is often referred as the "Frank-Wolfe bound."

It is our purpose here to show that all of these measures -- duality gap, rate of change, and Frank-Wolfe bound -- are effectively the same. Furthermore, we will show that G(x) is a convex function with easily computed subderivatives and therefore might itself be useful as an objective function.

4.4 Proofs of Equivalences

Our results in this section are in terms of problem (C), specialized when necessary to problem (P) as described earlier.

First we list the steps of the Frank-Wolfe algorithm:

Step 0. Choose x feasible to (C).

Step 1. Solve $\min_{y} \nabla f(x)\, y$

$$\text{s.t. } Ay = b$$

$$y \geqq 0$$

and call the solution $\bar{y}$.

Step 2. Solve $\min_{0 \leqq \lambda \leqq 1} f(\lambda x + (1 - \lambda \bar{y})$

and call the solution $\bar{\lambda}$.

Step 3. Replace x by $\bar{\lambda}x + (1 - \bar{\lambda})\bar{y}$.

Step 4. Go to 1.

Since f is convex and continuously differentiable, the fundamental inequality

$$f(x_1) \geqq f(x_2) + \nabla f(x_1)\,(x_2 - x_1) \tag{4-3}$$

holds for all x_1, x_2. Thus, in particular

$$f(x^*) \geqq f(x) + \nabla f(x)\,(x^* - x) \tag{4-4}$$

for any x. Furthermore,

$$\nabla f(x)\, x^* \geq \min_{y} \nabla f(x) y \qquad (4\text{-}5)$$

$$\text{s.t. } Ay = b$$

$$y \geq 0 .$$

Therefore,

$$f(x^*) \geq f(x) - \nabla f(x)\, x + \min_{y} \nabla f(x) y \qquad (4\text{-}6)$$

$$\text{s.t. } Ay = b$$

$$y \geq 0$$

The right hand side of (4-6) we call the convexity bound. It is obtained for any feasible x at the expense of solving a linear program. Note further that the linear program is exactly the subproblem in Step 1 of the Frank-Wolfe algorithm. Thus, while not indigenous to the method, the convexity bound arises naturally at no additional effort. Now assume $\bar{y}$ solves the linear program for a given x so that (4-6) becomes

$$f(x^*) \geq f(x) + f(x)(\bar{y} - x) \qquad (4\text{-}7)$$

The second term is clearly the directional derivative (rate of change) of f at x in the direction $(\bar{y} - x)$. Hence the convexity bound is f(x) plus this (nonpositive) quantity.

To see how the same bound arises from convex duality theory, we state the Lagrangian dual * [5] of (C) as:

---

* Murchland used conjugate duality theory. For our purposes conjugate duality and Lagrangian duality yield the same results.

(D) $$\max_{\lambda, u} \left[\min \; f(x) - \lambda^T (Ax-b) - u^T x\right].$$

$$\text{s.t.} \quad \nabla f(x) - A^T\lambda - u = 0$$

$$u \geqq 0.$$

Elimination of u and simplification of the objective yields

$$\max_{\lambda \in Y(x)} \left[\lambda^T b + \min_x f(x) - \nabla f(x)^T x\right] \tag{4-8}$$

where $Y(x) = \left\{\lambda \mid A^T\lambda \leqq \nabla f(x)\right\}$.

Thus, by the weak duality theorem [5]

$$f(x^*) \geqq \max_{\lambda \in Y(x)} \left[\lambda^T b + \min_x f(x) - \nabla f(x)^T x\right]. \tag{4-9}$$

Now, for fixed x, assume $Y(x) \neq \phi$. This gives

$$f(x^*) \geqq f(x) - \nabla f(x)^T x + \max_{\lambda \in Y(x)} \lambda^T b. \tag{4-10}$$

Observe that no assumption of x feasible to (C) was made. Therefore, for any x for which f and ∇f are defined we may bound $f(x^*)$ by solving the linear program

$$\max \quad \lambda^T b \tag{4-11}$$

$$\text{s.t.} \quad A^T\lambda \leqq \nabla f(x),$$

assigning a value of $-\infty$ if $Y(x) = \phi$.

The right hand sides of (4-6) and (4-11) are equal because the linear program of (4-11) is dual to the one in (4-6) and hence they have equal objective values.

It remains to relate the convexity bound to G(x). First, note that $F(x) = \nabla f(x)x$ for any x feasible to (P). Then consider (4-11) in the notation of problem (P):

$$\max_{\lambda} \sum_{i} \sum_{kj} (\lambda_k - \lambda_j) x^i_{kj} \tag{4-12}$$

$$\text{s.t.} \quad \lambda^i_k - \lambda^i_j \leq c_{kj}(x_{kj}) .$$

The objective is separable in i. For fixed $i \in D$ we have the linear program

$$\max \sum_{kj} (\lambda^i_k - \lambda^i_j) x^i_{kj} \tag{4-13}$$

$$\text{s.t.} \quad \lambda^i_k - \lambda^i_j \leq c_{kj}(x_{kj}) , \tag{4-14}$$

Summing the constraints (4-14) over any path, p, from $k \in O$ to i yields

$$\lambda^i_k - \lambda^i_i \leq \sum_{kj \in p} c_{kj}(x_{kj}) \tag{4-15}$$

Thus, for any $k \in O$, we have the bound

$$\lambda^i_k - \lambda^i_i \leq s^i_k(x) . \tag{4-16}$$

Furthermore, the objective (4-13) simplifies as follows:

$$\sum_{kj} (\lambda^i_k - \lambda^i_j)\, x^i_{kj} = \sum_{k \in O} (\lambda^i_k - \lambda^i_i)\, T^i_k \qquad (4\text{-}17)$$

by expansion and collection of terms and using the fact that the x^i_{kj} are feasible to (P). Thus, to maximize (4-17) subject to (4-16) let

$$\lambda^i_i = 0 \qquad i \in D$$

$$\lambda^i_k = s^i_k(x) \quad k \in O,\ i \in D$$

and, having attained the bound in (4-16), the linear program (4-13) - (4-14) is solved. Repeating for all $i \in D$ solves (4-12). This leads to

$$S(x) = \max \ \lambda^T b \quad \text{s.t.} \ A^T\lambda \leqq \nabla f(x) \qquad = \min_y \nabla f(x)\, y \quad \text{s.t.} \ Ay = b,\ y \geqq 0 . \qquad (4\text{-}18)$$

So we may write (4-6) as

$$f(x^*) \geqq f(x) - (F(x) - S(x)) = f(x) - G(x) . \qquad (4\text{-}19)$$

In summary, the duality gap, $G(x)$, is the difference between $f(x)$ and and the convexity bound. Furthermore the directional derivative (rate of change) arising at each iteration of the Frank-Wolfe algorithm is the negative of $G(x)$.

4.5 Geometrical Interpretation

The convexity bound can be interpreted geometrically by considering simple convex programs in one and two variables. In one variable consider

$$\begin{aligned} \min \quad & f(x) = x^2 \\ \text{s.t.} \quad & 1 \leq x \leq 3 \end{aligned} \tag{4-20}$$

which has the solution $x^* = 1$, $f(x^*) = 1$. Since $\nabla f(x) = 2x$, we may write

$$G(x) = 2x^2 - \min_{1 \leq y \leq 3} 2xy = 2x(x-1) \tag{4-21}$$

for all $x \geq 0$. At $x = 3$, $f(x) = 9$ and the bound is -3. In Figure 4-2 this is interpreted as the slope of the tangent to x^2 at $x = 3$ times $(x - \bar{y})$ where $\bar{y}\,(=1)$ is the solution of the linear program in (4-21).

A two-variable example relates the results of section 4.4 to the Frank-Wolfe algorithm. The pentagon in Figure 4-3 defines the constraint region of the problem (C). From the point x, the linear program of Step 1 returns the extreme point solution $\bar{y}$. The line search (Step 2) is conducted along the line from x to $\bar{y}$ and yields the next iterate $\bar{x}$. The inner product of $\nabla f(x)$ with the search direction $(\bar{y} - x)$ is the negative of $G(x)$ as we have proven. The inner product is negative, of course, because the angle between the vectors is obtuse.

The insight gained from these two examples with respect to problem (P) is that if the links are congested so that the objective function has a steep slope (Figure 4-2), or if the flow values x are far

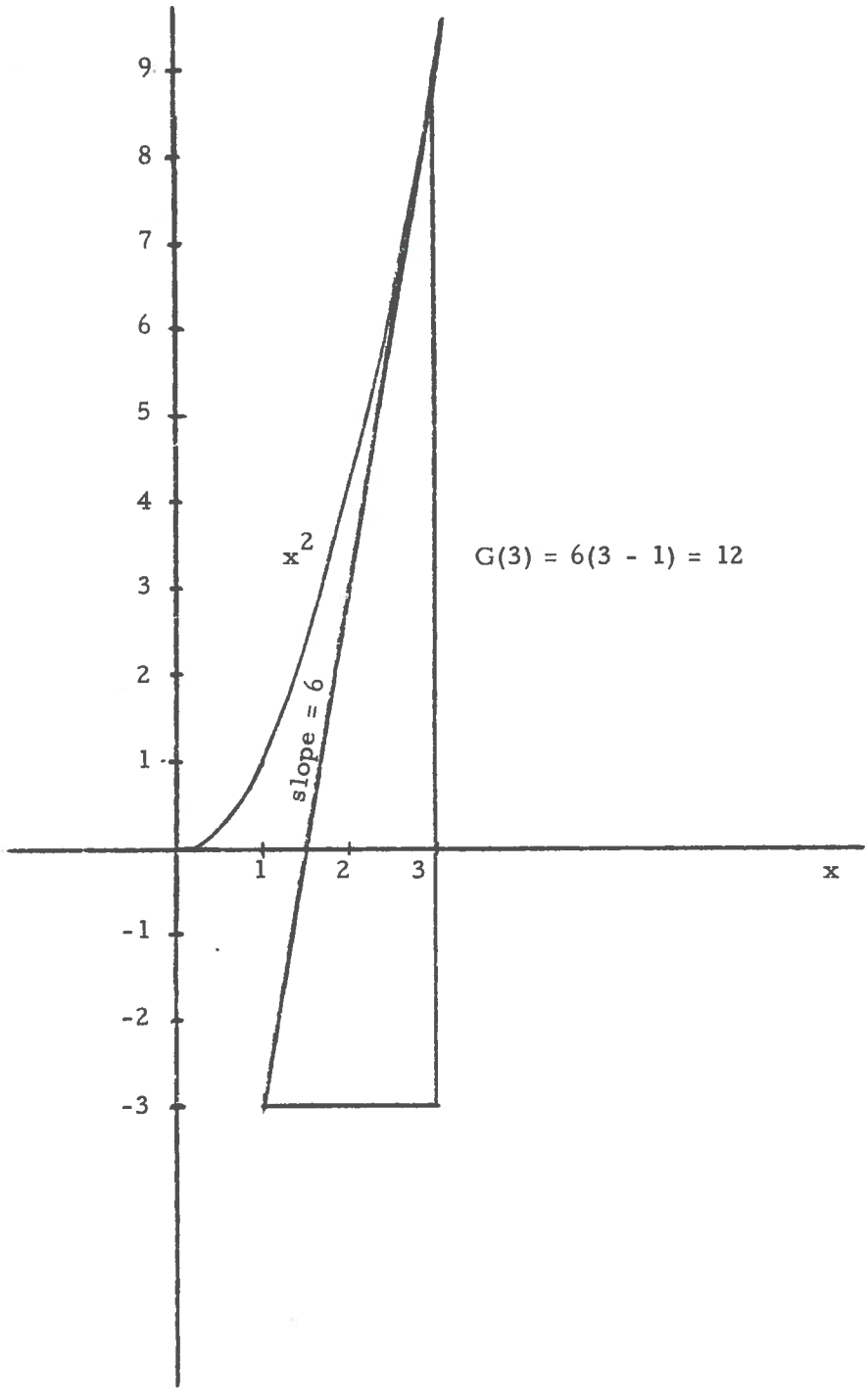
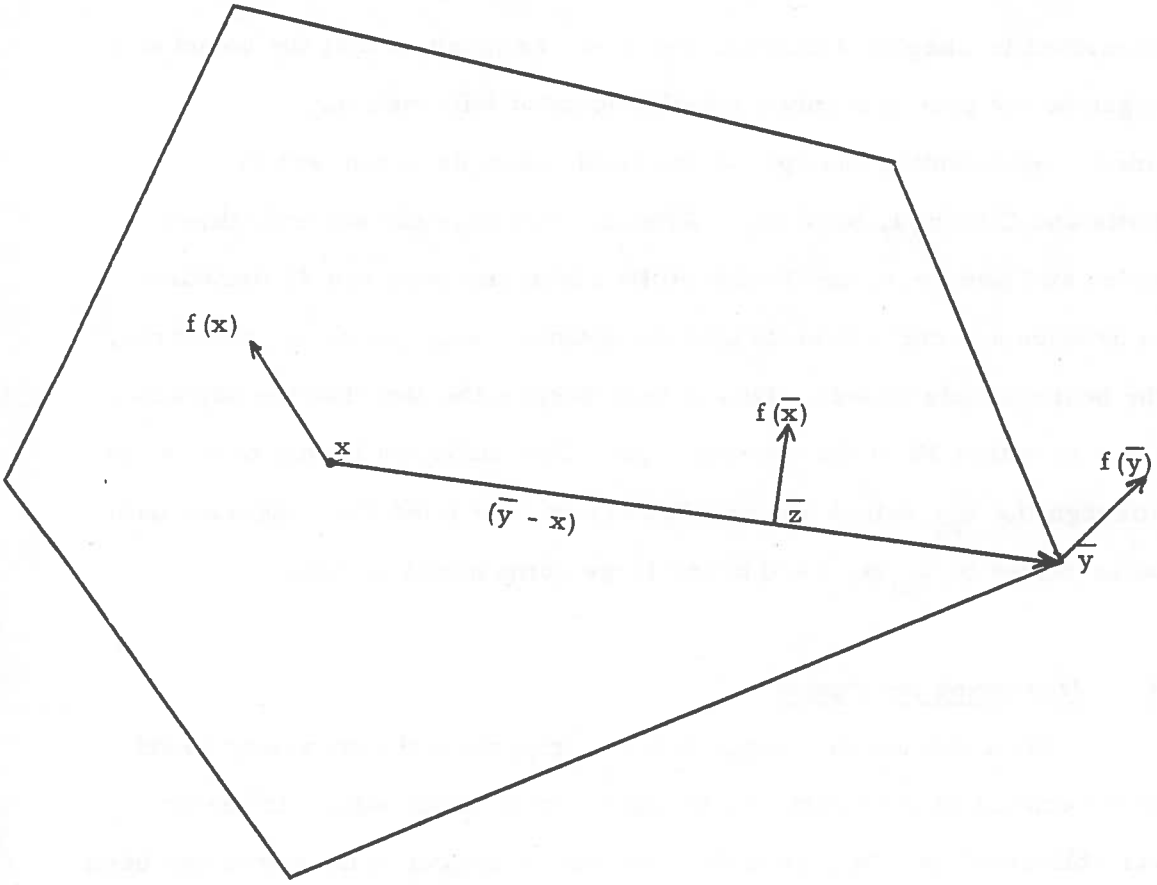


Figure 4-2. Geometrical Interpretation of Convexity Bound



$$G(x) = - f(x)\,(\bar{y} - x)$$

Figure 4-3. Interpretation of G (x) in the Frank-Wolfe Algorithm

from $\bar{y}$ (Figure 4-3) then G(x) will be large and the convexity bound will be weak. In practice, the bound has often proven to be quite weak for either or both of these reasons. In the aggregation experiments of described in Chapter 3 this was the case, so much so that the bound was negative and provided only a small amount of information.
Another outstanding example is the small example in the text by Potts and Oliver [1, page 96]. Although this example has only three nodes and five links, the Frank-Wolfe algorithm requires 73 iterations to produce a bound within 10% of the optimal value (which is, of course, the best possible bound). This is true despite the fact that the objective value is within 3% of the optimal value. The difficulty in this case is that although the x_{kj} values are nearly correct, the links are congested with large values of $c_{kj}(x_{kj})$ and hence large components of $\nabla f(x)$.

4.6. Improving the Bound

The most obvious suggestion for improving the convexity bound in the context of the Frank-Wolfe algorithm is illustrated with the one variable problem. In Figure 4-4 note that a tangent to the curve has been constructed at the point $\bar{y}$, the solution of the linear subproblem. The maximum of the two tangent lines is itself a convex function which underestimates f(x) and the minimum with respect to $1 \leq x \leq 3$ is at $x = 1$. This yields an improved bound of value 1, which we call the minimax bound. For this example the bound has been improved to the best possible value, but in general this need not occur, especially in multivariable problems. Another possibility is to choose x heuristically and employ (4-10) and (4-18).

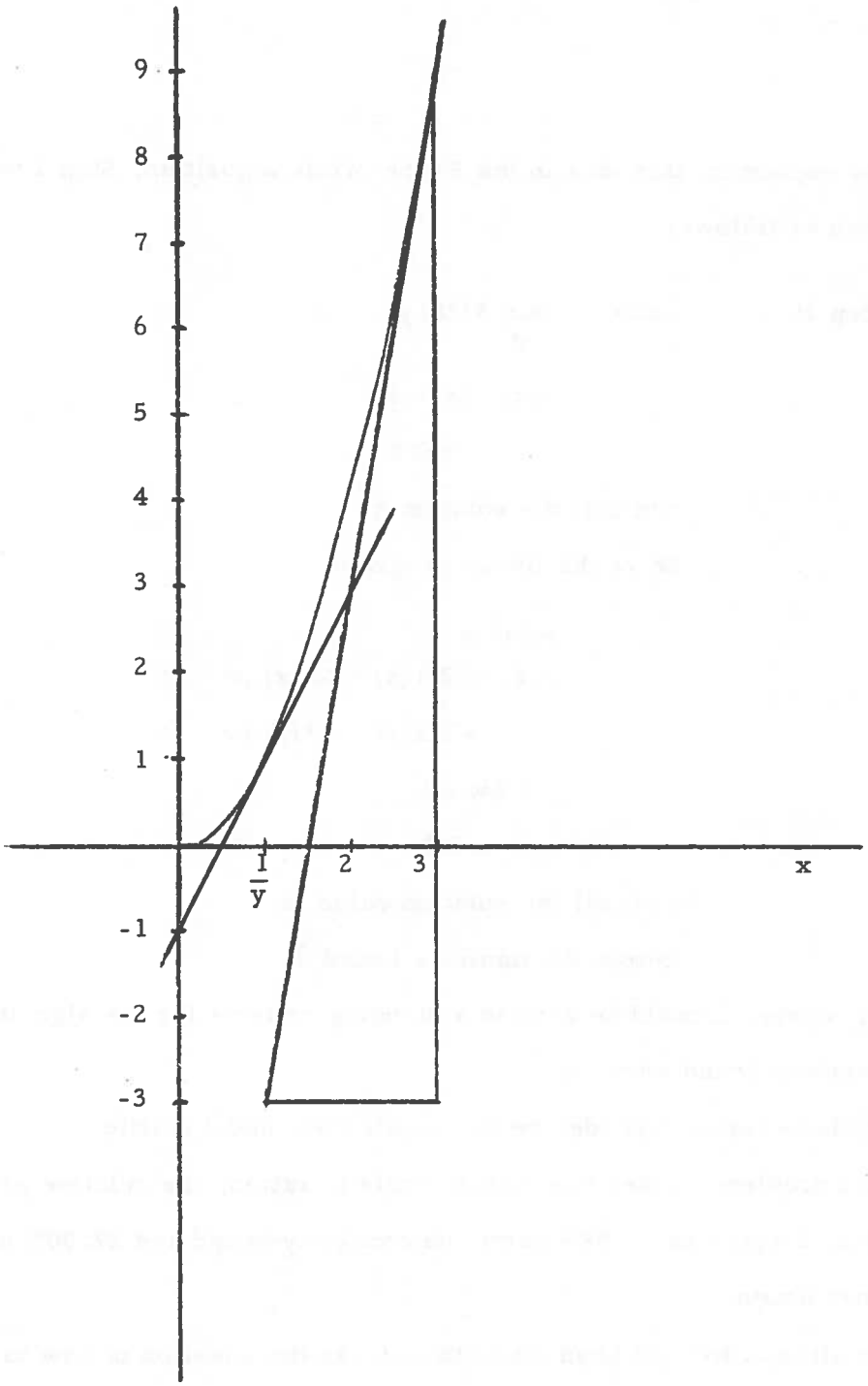


Figure 4-4. Obtaining the Minimax Bound

To implement this idea in the Frank-Wolfe algorithm, Step 1 would be modified as follows:

Step 1' Solve $\min_{y} \nabla f(x)\, y$

$$\text{s.t.} \quad Ay = b$$

$$y \geqq 0$$

and call the solution $\bar{y}$.

Solve the linear program

$$\min \quad z$$

$$\text{s.t.} \quad z \geqq f(x) + \nabla f(x)\,(w - x)$$

$$z \geqq f(\bar{y}) + \nabla f(\bar{y})\,(w - \bar{y})$$

$$Aw = b$$

$$w \geqq 0.$$

and call the solution value $\bar{z}$.

Output the minimax bound $\bar{z}$.

Of course, $\bar{z}$ could be used in a stopping criteria for the algorithm as the convexity bound often is.

We have tested this idea on one small (four node) traffic assignment problem. After one Frank-Wolfe iteration, the relative error in the value of $f(x)$ was 72.58% using the convexity bound and 27.90% using the minimax bound.

No attempt has yet been made to address the question of how to solve for $\bar{z}$ in large problems, but this appears to be an interesting research problem.

4.7 Minimizing the Gap

Another interesting research problem is to consider G(x) as an objective and design an algorithm which minimizes it directly to achieve the optimal set of flows x^*. The results which follow establish that this is valid in theory, at least if f(x) is quadratic.

Lemma 1. G(x) is a convex function for all x feasible to (P).

Proof. F(x) is convex under the usual assumptions on c(x). -S(x) is convex because S(x) is concave. This is most easily seen from (4-18), i.e., S(x) is the pointwise minimum of functions linear in y.

Consider now

$$(P') \qquad \min\ G(x)$$
$$\text{s.t.} \quad Bx^i = b_i$$
$$x^i \geqq 0 \qquad i \in D$$

Lemma 2. x^* solves (P') if and only if x^* solves (P).

Proof. (P') has optimal value $G(x^*) = 0$. But $G(x^*) = F(x^*) - S(x^*) = 0$ if and only if x^* is a user equilibrium flow.

In general G(x) is not differentiable but as a proper convex function it possesses subgradients denoted by $\partial G(x)$.

Lemma 3. The subgradients of G(x) are given by

$$\partial G(x) = \partial F(x) - \partial S(x)$$

where

$$\partial F(x) = \nabla F(x)$$

and

$$\partial S(x) = \left\{\text{all multipliers of } \max\ \lambda^T b \quad A^T\lambda \leqq \nabla f(x)\right\}$$

Proof. The first two equalities are immediate and the third follows from Theorem 4, page 471 of Lasdon [5].

Lemma 4. If $G(x)$ is differentiable

then $$\partial G(x) = \nabla G(x) = \nabla f(x) + \nabla^2 f(x)(x - \bar{y})$$

where $\bar{y}$ is as defined in Step 1 of the Frank-Wolfe algorithm.

Proof. Follows from Lemma 3 and application of the chain rule.

From a practical point of view, (P') has the disadvantage that its objective can only be evaluated by the solution of a linear program. Hence the question of typical line searches (as in Step 2 of the Frank-Wolfe algorithm) is raised. On the other hand, the second order terms in Lemma 4 suggest the possibility of improved convergence.

References

1. Potts, R. B. and R. M. Oliver, Flows in Transportation Networks, Academic Press, 1972.

2. Beckmann, Martin, Studies in the Economics of Transportation (with McGuire and Winsten), Yale University Press, New Haven, 1956.

3. Murchland, John D., "Road Network Distribution in Equilibrium," Presented at Mathematical Methods in the Economic Sciences Conference, Oberwolfach, October, 1969.

4. Dial, R. B., private communications, March 1977.

5. Lasdon, L. S., Optimization Theory for Large Systems, Macmillan, 1970.

5. Mathematical Programming and Extraction Aggregation

5.1 Introduction

In this chapter we concentrate on steps 5, 6 and 7 (Chapter 2) of extraction aggregation. We show how mathematical programming theory can serve to integrate this portion of the aggregation process and suggest computational procedures.

5.2 Transfer and Flow by Convex Programming

Steps 5 and 6 of extraction aggregation, transferring the trip table to the extracted network and flowing the extracted network, can be merged into a single operation and solved by the methods of convex programming. In this section we introduce two models for doing this. First we make the following assumptions -

Assumption 1	All trips of the original trip table will use the extracted network.
Assumption 2	The flowing of the extracted network can be accomplished by solving a convex program with flow conservation constraints.
Assumption 3	For each centroid of the original network, there is a known subset of pseudo centroids to which the trips can be transferred.

The first assumption is consistent with most practices of Chapter 2. The second assumes that the problem of step 6 is of the sort usually found in the traffic assignment literature. The final assumption is that each centroid to be aggregated can be "attached" to some subset of

pseudo centroids manually. This was done in Wilson's load-node method (Section 2.4) and is reasonable for studies such as the Shirley one (Section 2.2) because trips from a given zone would only use a few nearby accesses of the extracted freeway system. In the methods of Dial and FRA (Sections 2.3 and 2.6) the centroids are attached only to the nearest pseudo centroid. Hence all trips are transferred there. What we propose is closest to the long trip loop of Mann's method (Section 2.4); that the transfer be made dynamically as part of the flow phase.

Figure 5-1 depicts the idea. Each centroid (C_1, C_2, C_3) is attached via a pseudo connector directly to pseudo centroids of the extracted network. We assume that the decision of <u>where</u> to attach these has been made. With this in mind, we modify steps 5 and 6 of the aggregation process to be

5' Attach	-	Each centroid of the original network to a subset of the pseudo centroids
6' Flow	-	The extracted network including the original centroids and the pseudo connectors.

Thus the attach step replaces the transfer step and the transfer of trip demands occurs as part of the flow step.

Since step 6 is assumed to be a convex program, it is obvious that step 6' could be a convex program of the same form with fewer links, i.e., the deleted links of the network are simply replaced by pseudo connectors. To make this model complete, it is probably necessary to impede flows on the pseudo connectors as an (aggregate) simulation of impedances on the deleted links. Some methods for doing this have been introduced by Chan [1].

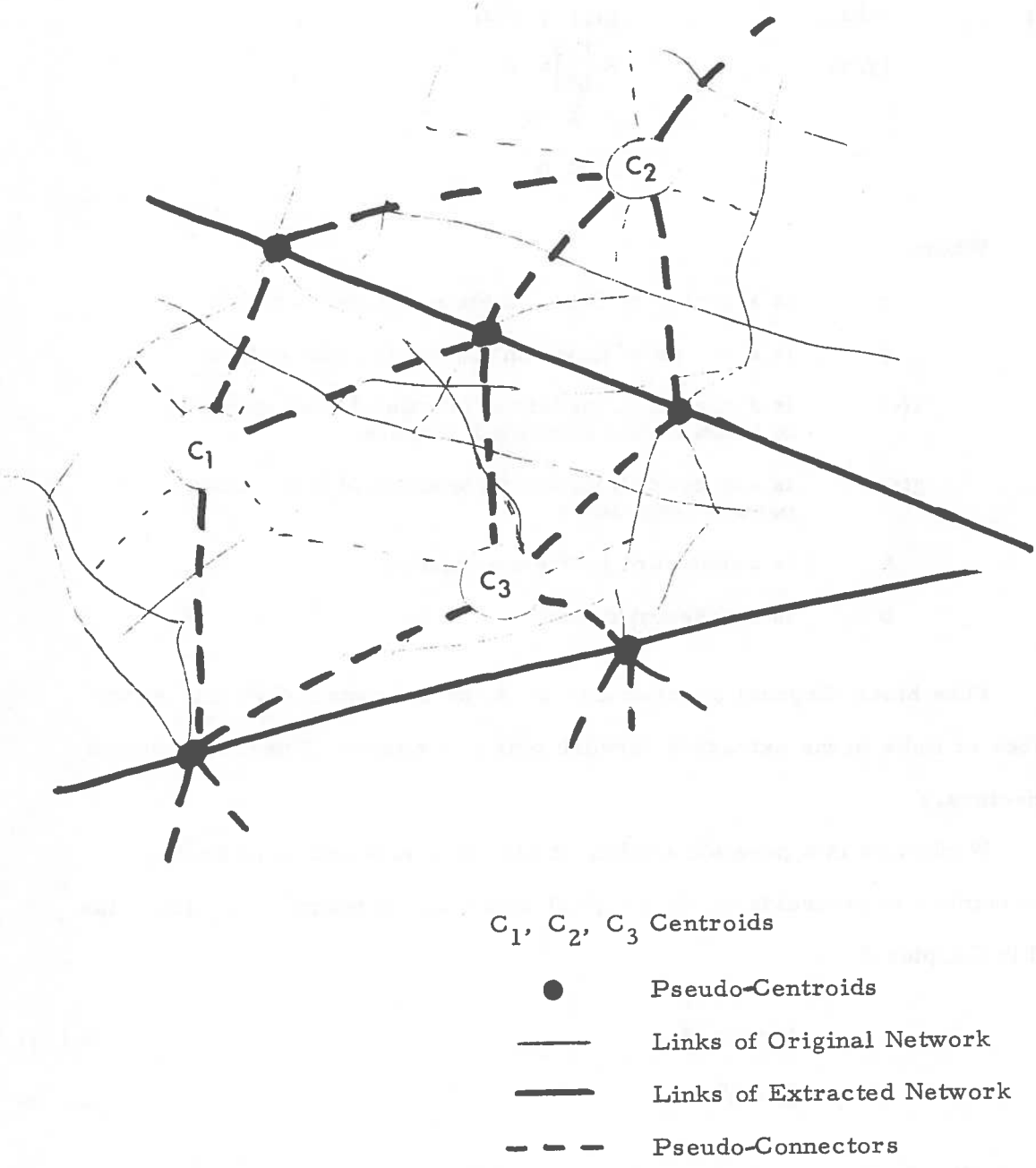


Figure 5-1. Attachment of Centroids to Pseudo-Centroids

Formally, we have

$$(M1) \qquad \min_{(y,x)} \quad g(y) + f(x)$$

$$A \begin{bmatrix} y \\ x \end{bmatrix} = b$$

$$y \geqq 0$$

$$x \geqq 0$$

Where

x	is a vector of flows on the extracted network
y	is a vector of flows on the pseudo connectors
$f(\cdot)$	is a convex (impedance) function (Assumption 2) of flows on the extracted network
$g(\cdot)$	is a convex (impedance) function of flows on the pseudo connectors
A	is a node-arc incidence matrix
b	is a constant vector

(The block diagonal submatrices of A have as many columns as the number of links in the extracted network plus the number of pseudo centroid connectors.)

While this is a possible model, it has the disadvantage of having the same number of centroids as the original network. In terms of the formulas used in Chapter 2

$$t \sim m \ell \tag{5.1.1}$$

$$s \sim \ell \tag{5.1.2}$$

the quantity ℓ is reduced (perhaps substantially) but the quantity m is unchanged. Also, the function $g(\cdot)$ is unknown and must be estimated. Examples indicate that in some cases, but not all, it can be identically zero.

We now suggest an additional model which attempts to reduce both m and ℓ in the above formulas.

The primary notion is that the flows on pseudo connectors are, under our assumptions, the components of a trip table (when summed at each pseudo centroid) for the extracted network. Put another way, if we knew a correct trip table we could flow just the extracted network. We emphasize a trip table because under Assumption 1, there are many trip tables which will induce flows on the extracted network that are the same as one would obtain by flowing the entire network.

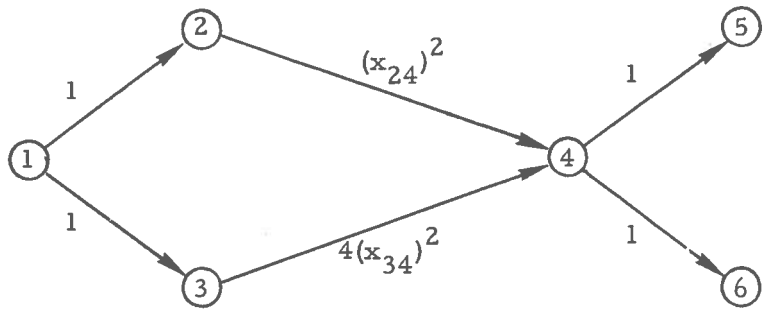
[As a numerical example, consider Figure 5.2. The quantities beside each arc represent "cost" per unit of flow - these are the constant 1 on all but the center arcs where the values increase as a function of flow. Assume the optimal solution for the full network is $x_{24} = 6$ and $x_{34} = 3$. (This is the case if the objective is the familiar user-equilibrium criteria for traffic assignment.) For the extracted network, any trip table with row and column sums as shown will yield the correct flows.]

The model is

$$
\begin{array}{lll}
\text{(M2)} & \min_{(x,\,z)} & f(x) + h(z) \\
& \text{s.t.} & Bx = T(z) \\
& & Dz \leqq c \\
& & z \geqq 0 \qquad x \geqq 0
\end{array}
$$

where

- x is a vector of flows on the extracted network
- z is a vector of the (unknown) trip table components for the extracted network

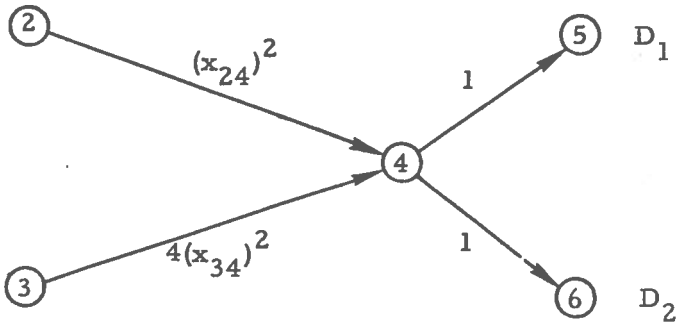


x_{kj} = flow on link kj

Trip Table

	5	6
1	5	4

a. Full Network



Trip Table

	5	6	Σ
2	?	?	6
3	?	?	3
Σ	5	4	9

b. Extracted Network

Figure 5.2. Non-Uniqueness of Trip Table for the Extracted Network

B	is a node-arc incidence matrix of just the extracted network
$T(\cdot)$	is a trivial linear transformation that converts the trip table z to the right hand side of the flow equations
$h(\cdot)$	a convex impedance function
D, c	are a constant matrix and vector that constrain the trip demands at each pseudo centroid.

The matrix D and the vector c are determined by step 5', e.g., in Figure 5.3b the row and column sums of the trip table are constained. As with model (M1) we have included an impedance function $h(\cdot)$ which must be estimated. (It should be thought of as node impedance at the pseudo centroids.) In some cases it may be identically zero, as before.

Models (M1) and (M2) are quite close, of course, but (M2) will generally be smaller, i.e., in terms of the quantities m and ℓ in formulas (5.1.1) and (5.1.2).

The question of solving (M1) or (M2), assuming the functions $g(\cdot)$ or $h(\cdot)$ are estimated, will now be addressed. We consider just (M2), but the same ideas could be employed with (M1).

First, we note that (M2) is a linearly constrained convex program which can be solved by the methods of nonlinear programming -- gradient projection, reduced gradient, convex simplex, etc. [2]. Second, we consider solving the problem iteratively. That is, fix the z variables and then solve for the x, etc. This is accomplished by <u>projection</u> [3] into the space of z variables. This yields a "master" problem in z and a subproblem in x:

$$\text{(M2M)} \qquad \min_{z} \quad w(z) + h(z)$$
$$Dz \leqq c$$
$$z \geqq 0$$

and

$$\text{(M2S)} \qquad w(z) = \min_{x} \quad f(x)$$
$$Bx = T(z)$$
$$x \geq 0$$

This is the method of "right hand side decomposition" discussed by Lasdon [4] and Geoffrion [3]. From the theory presented in these two references the following results can be proven:

a) $w(z)$ is a convex function

b) Subderivatives of $w(z)$ can easily be obtained from the multipliers of (M2S).

c) The iterative process of solving (M2M) by tangential approximation is globally convergent [3].

Whether the decomposition method of solving (M2M) is more efficient than a direct attack on (M2) is a question for numerical experimentation. It will only be successful if the subproblem (M2S) is solved quickly for z fixed. For highway traffic assignment, this subproblem is well-studied and many computational procedures exist which take advantage of its special structure. Generally, these are themselves slow to converge, but our computational experience shows that if the extracted network is simple enough the subproblem solutions can be obtained with relative ease.

It is interesting to note that a recent paper by Nguyen [5] utilizes an important special case of (M2). Nguyen addresses the problem of constructing a trip table (which is consistent with the user-equilibrium model discussed in Chapter 4) from data observable on an actual network. In the notation of the previous chapter, he assumes the travel time λ_k^i is known (observed) for $k \epsilon 0$ and $i \epsilon D$, and then proves that the solution of

$$(N) \qquad \min_{(x,\,z)} \; f(x) - \sum_{\substack{i \in O \\ k \in D}} \lambda_k^{\,i} z_k^{\,i}$$

$$Ax = T(z)$$

$$x \geqq 0$$

where f, A, x, λ, O and D are as in Chapter 4 and $z_k^{\,i}$ is the number of trips from $k \in O$ to $i \in D$,

yields a set of flows and a trip table satisfying the user-equilibrium principle. Problem (N) relates to problem (M2) by setting $h(z) = -\Sigma \lambda_k^{\,i} z_k^{\,i}$, B = A, D = 0 and c = 0.

Finally, to measure the effort of solving (M2) versus (M1) in a particular case, we again employ the formula (5.1.1). We do this for the Shirley study (Chapter 2). From the data introduced in Section 2.2, we have

$$t_{M1} \sim 700 \cdot (3100)$$

assuming (as seems reasonable) that each centroid is attached to three pseudo centroids. The quantity t_{M2S} is the t_A of Section 2.2

$$t_{M2S} \sim 42 \cdot (1000) .$$

Thus

$$\frac{t_{M1}}{t_{M2S}} \doteq 52 .$$

This difference of one to two orders of magnitude is based on comparing one iteration of M1 vs. M2S. The important question of how many total iterations there would be is not known.

5.3 Comparison by Linear Programming

Assuming that flows for the extracted network has been found by solving a reduced problem, (M1) or (M2), the question of measurement remains.

In Chapter 4 of this report we derive gross measures of error which require that feasible flows be known for all links of the network in question. To employ these then, it is necessary to "lift" the flows determined in step 6 to the given network.

This can be accomplished in a number of ways. We mention one here based on the assumption that (M2) has been solved. This being the case, the vector z is known.

Flow conservation equations for the links of the network not extracted in step 3 can be formulated with the pseudo centroids as destinations. The original trip table must be aggregated by destination in a (non unique) manner to agree with z, but this is easy to do. Having formulated these equations, they may be solved for the flows on the remaining network links. The simple structure of flow conservation equations makes this easy also. However, completely arbitrary flows will probably not be desirable as indicated in Chapters 3 and 4 -- overly congested links imply weak error bounds. One alternative would be to minimize maximum relative flow (on the individual links) subject to the flow conservation equations. Intuitively this would tend to spread the flows over the remaining links in an even fashion and thereby yield a good error bound as discussed in Chapter 4.

Formally, the model is

$$\min_{x_{kj}} \left[\max_{kj \in S} x_{kj}/c_{kj} \right]$$

$$\text{s.t.} \quad \bar{B}\, x^{\ell} = \bar{b}_{\ell}$$

$$x^{\ell} \geq 0$$

where
- S - areas not on the subnetwork
- ℓ - index of pseudo-centroids
- $\bar{B}$ - arc-node incidence matrix for the arcs in S
- $\bar{b}_{\ell}$ - trip vectors constructed from z.
- c_{kj} - capacity of link k_j

This problem is equivalent to a linear program and has the same structure as the problem posed in Chapter 4 for obtaining the minimax bound.

References

1. Chan, Y. "A Method to Simplify Network Representation in Transportation Planning," Transportation Research, Vol. 10, pp. 179-191, 1976.
2. Luenberger, David G., "Introduction to Nonlinear Programming," Addison-Wesley, 1973.
3. Geffrion, A. M. "Elements of Large-Scale Mathematical Programming," Management Science, 16, 11, July 1970.
4. Lasdon, L. S. Optimization Theory for Large Systems, McMillan, 1970.
5. Nguyen, S., "Estimating an OD Matrix from Network Data: A Network Equilibrium Approach," Centre de recherche sur les transports, Report #60, University of Montreal, February 1977.

110 Copies

U. S. DEPARTMENT OF TRANSPORTATION
TRANSPORTATION SYSTEMS CENTER
KENDALL SQUARE, CAMBRIDGE, MA. 02142

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
U. S. DEPARTMENT OF TRANSPORTATION
518