REPORT NO. DOT-TSC-RSPD-78-7

AGGREGATION IN NETWORK MODELS FOR TRANSPORTATION PLANNING

Daniel Cullen
Harold Kuhn
Marguerite Frank

Mathematica Inc.
P.O. Box 2392
Princeton NJ 08540

FINAL REPORT

FEBRUARY 1978

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

Prepared for

U.S. DEPARTMENT OF TRANSPORTATION
OFFICE OF THE SECRETARY
Office of the Assistant Secretary for
Systems Development and Technology
Office of Systems Engineering
Washington DC 20590

Technical Report Documentation Page

1. Report No.	2. Government Accession No.	3. Recipient's Catalog No.
DOT-TSC-RSPD-78-7		

4. Title and Subtitle	5. Report Date
AGGREGATION IN NETWORK MODELS FOR TRANSPORTATION PLANNING	February 1978
	6. Performing Organization Code
7. Author(s)	8. Performing Organization Report No.
Daniel Cullen, Harold Kuhn, and Marguerite Frank	DOT-TSC-RSPD-78-7
9. Performing Organization Name and Address	10. Work Unit No. (TRAIS)
Mathematica Inc.* P.O. Box 2392 Princeton NJ 08540	OS750/R7502
	11. Contract or Grant No.
	DOT-TSC-883
	13. Type of Report and Period Covered
12. Sponsoring Agency Name and Address	Final Report June 1974 - October 1975
U.S. Department of Transportation Office of the Secretary Office of the Asst. Sec. for Sys. Dev. and Tech. Office of Systems Engineering Washington DC 20590	14. Sponsoring Agency Code

15. Supplementary Notes

*Under Contract to: U.S. Department of Transportation
Research and Special Programs Administration
Transportation Systems Center
Cambridge MA 02142

16. Abstract

This report documents research performed on techniques of aggregation applied to network models used in transportation planning. The central objective of this research has been to identify, extend, and evaluate methods of aggregation so as to improve the capabilities of the transportation planner by better computational methods, by more flexible models, and by increased confidence in the results obtained from aggregated models.

The results of the research are presented in a series of papers. Papers 1 and 2 address the question of error bounding in the lifted (disaggregated) solution using the application of duality theory. Paper 3 reviews the potential performance of a new algorithm for the equilibrium model using path extraction aggregation. Paper 4 addresses potential savings of aggregation. Paper 5 presents results on aggregation test networks and attempts to formulate rules for equilibrium models. Papers 6 and 7 present specific mathematical results on two aggregation-related problems.

17. Key Words	18. Distribution Statement
Aggregation, Networks, Transportation Planning, Duality, Equilibrium Model.	DOCUMENT IS AVAILABLE TO THE 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	136	

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

PREFACE

The research described in this report was conducted under the Transportation Network Analysis portion of the Transportation Advanced Research Project (TARP) sponsored by the Office of the Assistant Secretary for Systems Development and Technology. The network analysis portion of the program is a broad research effort aimed at developing (1) computational techniques for transportation network models, and (2) control-theoretical tecniques for real-time management of large scale transportation systems. Other research efforts in network analysis cover decomposition methods for large scale networks, real-time control of urban freeway networks, and computational methods for large scale freight and fleet routing problems.

The contract under which this work was performed was guided throughout by Dr. Edwin J. Roberts, TSC-213, and by Dr. John J. Fearnsides and Dr. Robert J. Ravera 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

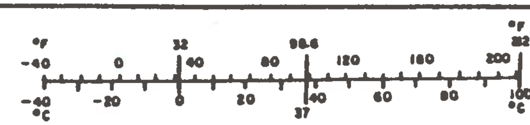


TABLE OF CONTENTS

A. SUMMARY GUIDE TO FINAL REPORT

1. Introduction

This section provides a summary guide to the research performed by MATHEMATICA on techniques of aggregation applied to network models used in transportation planning. The central objective of this research has been to identify, extend, and evaluate methods of aggregation so as to improve the capabilities of the transportation planner by better computational methods, by more flexible models, and by increased confidence in the results obtained from aggregated models.

To set the background of this research, we shall give a quite general statement of the current status of network models used in transportation planning. The structure of this statement parallels the structure of the comprehensive bibliography prepared as part of our project. Concise descriptions are given for elements in the three categories that are central to our research:

(i) network models;

(ii) mathematical techniques;

(iii) aggregation methods.

This provides a precise setting for the results achieved up to now and for promising directions of future research, since the important questions to be answered all involve the interaction between these three categories.

This interaction can be explained in terms of a simple graphical device. Figure 1, following, represents an ideal listing of all aggregation methods (as rows) and all network models (as columns).

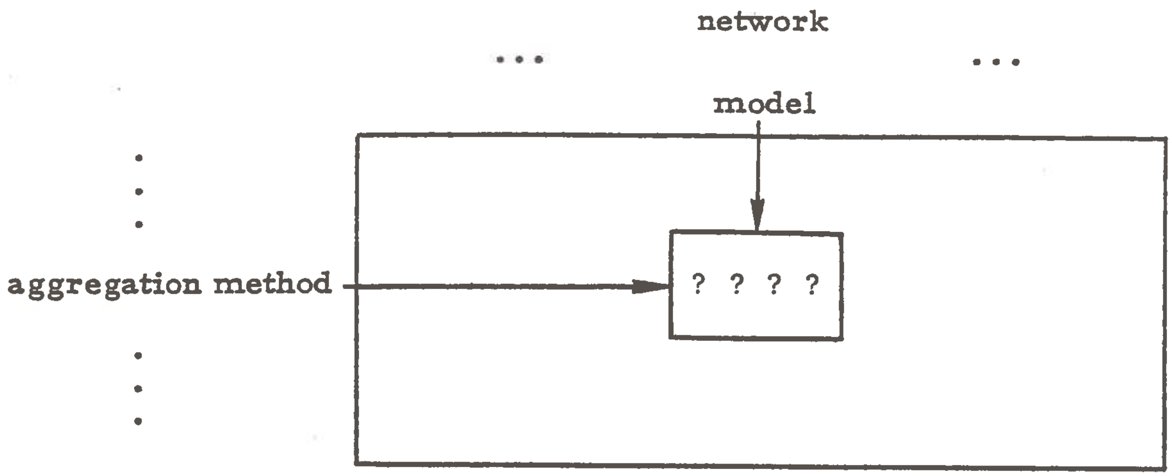


FIGURE 1. REPRESENTATION OF AGGREGATION METHODS AND MODELS

The combination of method and model raises a set of questions (to be elaborated elsewhere) about the results obtained from applying the aggregation method to the network model. Many of these questions depend on the mathematical technique used to solve the model (in either aggregated or disaggregated form). This interaction of aggregation method applied to network models using mathematical techniques is central to our research and the main object of this introduction is to make this interdependence precise by explicit definition and example. The sections that follow execute this objective in a self-explanatory style.

Before this exposition is begun, one missing element must be discussed. The bibliography referred to before has one other crucial category:

(4) modal applications.

Of course, this refers to actual planning for the real modes of transportation (road, rail, air, sea, etc.) and to real planning objectives. While we believe that the previous analysis <u>should</u> depend on this element of the classification to the same extent as the other three, at the present state of knowledge this is more pious hope than accomplishment. However, we shall indicate in general terms how this fourth category conditions the results to be sought and the answers that have been found using the other three, more theoretical categories.

2. Network Models

In this section, we shall give a concise description of the categories of network models that have been identified as being relevant to transportation planning. These follow the classification of our bibliography and so are indexed as P_1, P_2,... (P = Problem so as to distinguish them from M = Modal).

P_1: Equilibrium Models: An equilibrium (traffic assignment) problem asks for the assignment of flows to the network so that the constraints are satisfied (constraints may include demand requirements, capacity limitations, etc.) and so that users' costs cannot be improved by noncooperative actions (a user optimized or descriptive solution) or so that total cost is minimized (a system optimized or normative solution).

By variation of the constraints and the nature of the costs, many special problems (such as the Hitchcock-Koopmans Transportation Problem) fit this description, but because of their importance have been given their own category. As understood here, an equilibrium model refers to a traffic assignment problem in which the link costs to a user are (non-decreasing) functions of the flows on each link and the constraints specify origin-destination flows and (possibly) capacity bounds on some links.

P_2: Shortest Path Models: If we let "distance" stand for a general disutility function (cost, time, discomfort, etc.) then this category is one of the most important and appears as a major component of several other models (and mathematical techniques). In its simplest form, it asks for the shortest path from one fixed node to another node of a network. In one generalization, included in this category, it asks for all of the least cost paths. In another, more ambitious generalization, it asks for the k^{th} shortest path, for $k > 1$.

The importance of this model is underlined by the following quotation from Steenbrink[1]:

> "Unfortunately, the computation of a shortest path takes a relatively long computing time: for the road network for the Dutch Integral Transportation Study consisting of about 2,000 nodes and 6,000 links and with 351 origins and destinations, one shortest path computation for all origins and destinations followed by an assignment takes about 12 minutes on an IBM 360/65 computer, using an algorithm based on the algorithm of Moore/Ford/Bellman written in FORTRAN IV. So it has become clear now that the use of a good shortest-path algorithm is extremely important and may even be an important factor in the total costs of a whole transportation study. On the other hand, because study budgets are often more or less fixed in advance, the quality of the shortest-path algorithm used influences the quality of the transportation study a great deal."

P_3: Maximum Flow Models: This model asks for a maximum flow between a fixed origin-destination pair in a capacitated network. Various generalizations are possible which are closer to real-life measures of the real-life capacity of a network. However, the original problem has a highly developed mathematical structure and is often called the Ford-Fulkerson problem, due to their elegant algorithm and related max flow - min cut theorem.

As will be seen later, this model is a unique example of a model that yields to an exact aggregation method, due to Gomory and Hu.

P_4: Scheduling and Dynamic Flow Models: These models ask for flows that meet time requirements in addition to demand requirements and capacity constraints. Thus, links will have characteristics of

[1]Peter A. Steenbrink, Optimization of Transportation Networks, John Wiley & Sons, 1974, p. 150.

of capacity and/or (possibly congestion dependent) traversal times. The problem of determining maximum origin-destination flows within a specified time is often called the "dynamic max-flow problem." The related "scheduling problem" involves optimizing a function (such as cost) while meeting demand, capacity, and time related constraints.

P_5: Multi-models: These models involve multiple commodities, copies, modes, etc. that generalize other models. Examples are the multicommodity version of the Hitchcock-Koopmans Transportation Problem (P_7); traffic assignment for multiple classes of users (generalizing P_1); transportation infrastructures involving a choice of several modes; multiple copies of networks in dynamic analyses.

P_6: Stochastic Demand Models: These models ask for solutions to the other categories but with data that are probabilistic rather than deterministic. Examples are probabilistic demands, Markov decision models, queuing congestion problems, etc.

P_7: Hitchcock-Koopmans Transportation Model: This model (and its generalizations) asks for the flows from m origins to n destinations along links that may be capacitated, with costs which are either linear or convex in the flows, and so as to satisfy prescribed supplies and demands.

This category also includes the transshipment problem where several network nodes are transit points, the bottleneck transportation problem with consideration of congestion, and the fixed charge problem with positive link costs for zero flows.

P_8: Routing, Traveling Salesman, Longest Path Models: This category of network optimization problems involves combinatorial or graph theoretical considerations that place them outside the earlier categories. Either the specification of the constraints or the nature of the objective function requires complex combinatorial analysis.

Examples include routing problems such as the traveling salesman problem (minimizing total distance while visiting each node exactly once), maze problems, and finding the longest path between two nodes.

P_9: Socially Oriented Models: These models use networks as incidental to a related socially oriented objective. These include such questions as the optimality of network planning, the realism of network design, data reliability, traffic control, and investment policy. In general, network methodology is secondary in these models.

P_{10}: Mathematically Oriented Models: Although motivated by applied network problems, these models are "once removed" from transportation planning. As such they are formulated in purely mathematical terms. Included in this category are problems of abstract geometric design, graph theory, combinatorics, or matrix algebra.

P_{11}: Location Models: These models ask for optimal location of facilities (which affect the supply and demand at origins and destinations in other models) so as to minimize cost or maximize flow.

3. Aggregation Methods

Informally, aggregation means any method that transforms a given model into a model that is smaller in one of its parameters (say, the number of links, nodes, etc.). The second essential ingredient of an aggregation method is that a rule be given by which solutions to the smaller or aggregated model can be "lifted" to meaningful solutions to the original problem. Schematically, this can be represented as follows: Let M denote the original model, $\overline{M}$, the aggregated model. Optimal solutions are denoted by $S(M)$ and $S(\overline{M})$, respectively. Feasible values are denoted by $F(M)$ and $F(\overline{M})$, respectively. The "lifting operation" is a mapping L from $F(\overline{M})$ to $F(M)$. Given $\overline{S} \in S(\overline{M}) \subset F(\overline{M})$, the requirement that the lifted solution be meaningful is expressed as $L(\overline{S}) \in F(M)$. Ideally, $L(\overline{S}) \in S(M)$, that is, the lifted solution of the aggregated problem is a solution to the original problem. In actual fact, most aggregation methods, with one notable exception, only approximate solutions to the original problem. Therefore, the essential questions to be asked center on how well the lifted solutions approximate solutions to the original problem.

For any aggregation method applied to any network model, there are a set of standard questions that can be put which evaluate the efficacy of the method. These are listed below:

(A) What are the savings (say, in computational operations or time) in computing lifted solutions of the aggregated model versus computing exact solutions to the original model? Of course, this question is only well formulated for a combination of a network model, an aggregation method, and a mathematical technique of solution.

(B) Can errors in the lifted solutions be bounded? Of course, this question can be posed at two levels, a macro and a micro level. In most network models, the solution consists of a value for an objective function (such as total system cost) and values for operational parameters (such as flows). In general, we shall consider network values as macro and values associated with network elements as micro.

(C) Can biases in the lifted solutions be determined? (The same distinction between macro and micro made in (B) holds here.)

(D) Can any statistics (e.g., mean or variance) of errors in the lifted solutions for relevant populations of models be estimated? (Again, the distinction between macro and micro holds.)

We have identified two main types of aggregation methods:

A_1: <u>Hierarchical</u>: These methods depend on the recognition of a hierarchical structure in the given network. This structuring, which involves a process of valuation, orders the elements of the networks (e.g., nodes, links, paths, or chains) according to their importance or similarity, possibly merges them, then breaks off each dominant class from the one below. This can be done either arbitrarily, by numerical valuation, or more organically - as the quantitative becomes more qualitative - according to type (e.g., arterials versus feeder roads).

The hierarchical methods further classify as <u>extraction</u> or <u>abstraction</u> of network elements. These will be stated for links below; however, the same methods have been studied for nodes and paths in our research.

Link extraction: This means the judicious removal of links deemed to affect the solution only minimally, by some numerical or qualitative criterion.

Link abstraction: This means the substitution of an "abstract" link for a set of actual links. For example, a set of "parallel" links can be replaced by a single abstract link of equivalent total capacity, or a set of "series" links can be replaced by an abstract link of equivalent total length.

A_2: Intrinsic: This type of aggregation method seeks out an inner structure whose elements are ontologically not on the level of the original network elements - and it is this intrinsic abstract structure that is then represented by the aggregated network.

The unique example of this type of aggregation that we have studied is the Gomory-Hu Flow Analysis algorithm. This aggregation method replaces the original network by a flow equivalent tree, whose succession of links corresponds to a succession of nonintersecting minimal cuts in the original network.

4. Research Results

Our research has centered on an attempt to answer Questions (A) - (D), cited above, posed for the known aggregation methods applied to a subset of the most important network models. In terms of the diagram of the Introduction, this means studying these questions for the following combinations:

		Equi-librium	Shortest Path	Max Flow	Hitchcock Koopmans
Hierarchical	Extraction				
	Abstraction				
Intrinsic					

Although our choice of a model to study in detail was the equilibrium model, the other models have proved to be suggestive of techniques and provide illuminating examples of phenomena that we believe carry over to the equilibrium model.

The results of our research are abstracted below. Papers 1 and 2 are aimed at Question (B) above, using the application of duality theory from mathematical programming. In effect, this idea needs the following ingredients to be successful:

(1) The formulation of the model as a mathematical program.

(2) The formulation of a dual mathematical program.

(3) A lifting rule that converts primal and dual optimal solutions to the aggregated model into primal and dual feasible solutions to the original problem.

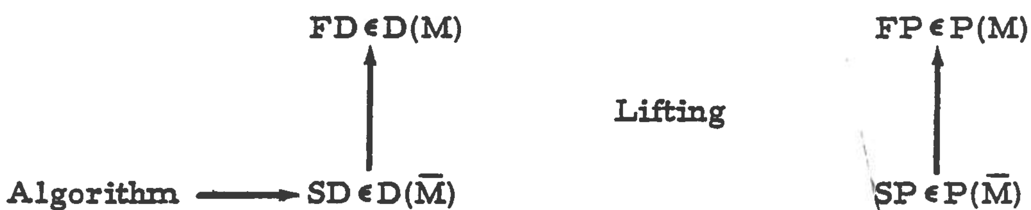
When these three elements are present, a bound can be given on the error in the (macro) objective function for the lifted solution.

A second major accomplishment is the formulation of a new algorithm for the equilibrium model that depends on a path extraction aggregation. Paper 3 provides evidence of the potential performance of this new algorithm and a detailed description of its detailed structure.

Paper 4 is a direct first attack on Question (A) above. Paper 5 presents results on aggregation test networks and attempts to formulate specific aggregation methods and lifting rules for the equilibrium models. Papers 6 and 7 present specific mathematical results on two aggregation related problems. The abstracts of all the papers follow.

Abstract 1. Bounding Aggregation Error in Network Models

This paper describes a new technique for measuring the error introduced in the optimal answer to a network model by aggregation methods. The technique can be explained as follows: Let M denote the original model, $\bar{M}$, the aggregated model. Optimal solutions are denoted by $S(M)$ and $S(\bar{M})$, respectively. It is assumed that the model is formulated as a (primal) mathematical program with a dual program available. Feasible solutions to the primal and dual programs are denoted by $P(M)$, $P(\bar{M})$, $D(M)$, and $D(\bar{M})$, respectively. Optimal solutions to the smaller or aggregated model are assumed "lifted" to feasible solutions to the original problem. This is represented in the following diagram:



Bounds for the aggregation error are provided by the following inequalities:

$$\text{Value of FD} \leq \text{Optimal value for } M \leq \underbrace{\text{Value of FP}}_{\substack{\text{Aggregated value} \\ \text{for } M.}}$$

In this paper, this idea is applied to node abstraction for the Hitchcock-Koopmans transportation model.

Abstract 2. Bounding Aggregation Error in the Equilibrium Model

This paper describes the application of the technique developed in Paper 1 to the equilibrium model (in both descriptive and normal form). By means of several simple test networks and representative cost functions, the various pieces of the theory are illustrated. Thus, the essential features of a theory applicable to more general combinations of aggregation method and mathematical technique are identified. These are primarily the construction of dual solutions and the efficient lifting of both primal and dual solutions.

Abstract 3. A Path Extraction Aggregation Algorithm

This paper presents a new algorithm for the network equilibrium model that works in the space of path flows using a labelling and pivot technique. A detailed set of specifications is given, convergence to an optimal solution is proved, and estimates of computational efficiency are provided.

Abstract 4. Computational Savings from Aggregation

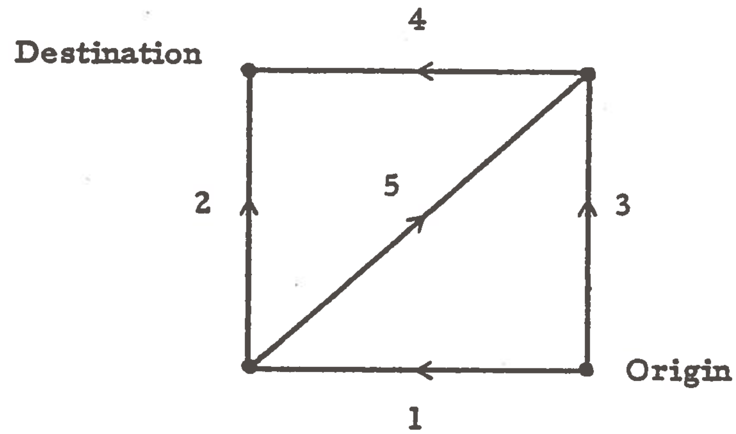
This paper considers the question of measuring the savings to be achieved by applying an aggregation method to a network model solved by a mathematical technique or algorithm. The aggregation method reduces the number of network elements (links, nodes, OD pairs, etc.) and this in turn reduces the number of multiplications, additions, and comparisons performed. Estimates are given for examples of all aggregation methods and all network models.

Abstract 5. Aggregation Test Networks

This paper describes an example constructed at MATHEMATICA for testing aggregation methods of the equilibrium model. It is based on the Massachusetts road distance network with OD demands derived from a primitive gravity model. A variety of mathematical techniques and aggregation methods can be tested on this example.

Abstract 6. Aggregation by the Extraction of a Transversal Link

This paper gives a complete and rigorous treatment of a simple case of aggregation by link extraction. This case is illustrated in the figure below:



This case is the extraction of the single link (5) between two chains (12 and 34) joining the OD pair. It is assumed that a flow of x units on link i costs $A_i x + B_i$, where $A_i > 0$ and $B_i > 0$ for $i = 1, \ldots, 5$.

The main result states that the aggregation cost (for the descriptive, user-optimized, equilibrium model) with an OD demand of K units is

$$\frac{-\Delta(-\Delta K + \Psi)}{\lambda \Sigma}$$

where $\Delta = A_1 A_4 - A_2 A_3$, $\Sigma = A_1 + A_2 + A_3 + A_4$,

$$\lambda = (A_1 + A_2 + A_5)(A_3 + A_4 + A_5) - A_5^2 > 0,$$

$$-\Delta K + \Psi > 0, \text{ and } \Psi \text{ is a polynomial in } \{A_i, B_i\}.$$

Interpreting the above model in terms of link insertion instead of extraction, this result completely characterizes the so-called "Braess Paradox," where the addition of a link increases the travel cost of every individual. The Braess paradox can occur if and only if the aggregation cost is negative (that is, $\Delta > 0$), and the demand K is bounded above and below by certain polynomials in $\{A_i, B_i\}$.

Abstract 7. Bounds and Estimates for Average Speed per OD Path

In a large transportation network, with n_1 origins, n_2 destinations, and an average of p paths joining each OD pair, the computation of the average speed $\bar{v}$ per OD path requires summing $p n_1 n_2$ products $f_\alpha^{OD} v_\alpha^{OD}$, where f_α^{OD} is the quantity shipped from O to D along path α with average speed v_α^{OD}.

By aggregating flows out of every origin and into every destination, this paper gives upper and lower bounds, M and m, and an estimate $\frac{M+m}{2}$ of $\bar{v}$ requiring the summation of only $2(n_1 + n_2)$ products.

If the origins and destinations are disjoint, the path characteristics are completely determined and expressed in terms of link characteristics (usually the only information available). If some nodes are both origins and destinations, then additional information regarding the flow into or out of these nodes is required to determine path flows.

5. Directions for Further Research

We believe that the results contained in this report establish the conclusion that research in aggregation is a promising area that will yield new methodology to extend the range of transportation planning. The subject is still in its infancy; however, we have identified a number of directions that should yield useful results. These fall roughly into three categories, which are described in detail in the last section of this report:

(I) Theory of Network Aggregation

(II) Algorithms Related to Network Aggregation

(III) Network Aggregation and Transportation Planning.

B. TECHNICAL PAPERS

PAPER 1

BOUNDING AGGREGATION ERROR IN NETWORK MODELS

1. Introduction

The basic question to be asked of any aggregation method is: What confidence can we place in the answers obtained through its use? More precisely, what can we say about the answers obtained from the aggregated network compared with the exact (but unavailable) answers for the original network with its full detail? The object of this paper is to describe and illustrate a new technique for answering one aspect of this question, namely: can we calculate upper bounds on the errors inherent in the answers which have been "lifted" from aggregated models?

2. Aggregation Methods and Lifting Rules

We start with a network model M. As understood here, this means a network with numerical (such as capacities) or functional (such as cost) characteristics for its elements and an (optimization) problem posed for this network. For example, we may be given a directed network with one source and one sink, with positive capacities given on all of the links, and ask: what is the maximum feasible flow from source to sink on this network? Optimal solutions for the problem posed on the network will be denoted by S(M). The set of feasible solutions for the problem will be denoted by P(M) and we shall assume, without loss of generality, that the optimization problem is a minimization problem. This means that we are seeking a feasible solution $FP \in P(M)$ such that its value is <u>least</u> among all feasible solutions in P(M).

The aggregation method produces a network model $\overline{M}$. As understood here, this means a network (smaller than M) with numerical or functional characteristics for its elements constructed from those of M and with an (optimization) problem posed for this network. We shall also impose another, very important requirement. Every feasible solution for the aggregated model $\overline{M}$ must be capable of being "lifted" to a feasible solution for the disaggregated model M. Precisely, there is defined a lifting mapping L,

$$L : P(\overline{M}) \xrightarrow[\text{into}]{} P(M).$$

Of course, it is hoped that the aggregated problem can be solved exactly and so the feasible solutions from $P(\overline{M})$ to be lifted are optimal solutions for $\overline{M}$, denoted by $S\overline{P} \in P(\overline{M})$. With this background, the question we pose is: can we calculate an upper bound on the error in $L(S\overline{P}) = FP \in P(M)$?

Before we introduce the technique for calculating such a bound, we shall introduce an example of the concepts defined thus far, to which the technique will be applied later in this paper. The example is the Hitchcock-Koopmans Transportation Problem. The network is a directed graph with $m + n$ nodes, called <u>sources</u> (m) and <u>destinations</u> (n) with a link from each source to each destination. The characteristics are (positive) <u>supplies</u> a_i at each source, (positive) <u>demands</u> b_j at each destination, and (nonnegative) <u>unit costs</u> c_{ij} on each link. The optimization problem is then:

$$\text{Find } x_{ij} \geqq 0 \ (i=1, \ldots, m; j=1, \ldots, n) \quad \text{such that}$$

$$\sum_j x_{ij} = a_i \quad \text{and} \quad \sum_i x_{ij} = b_j \qquad \text{(feasibility)}$$

$$\text{and minimizing } \sum_{i,j} c_{ij} x_{ij} \qquad \text{(optimality)} .$$

(Naturally, to insure feasibility, we have assumed $\sum_i a_i = \sum_j b_j$.)

A "natural" aggregation method is to combine similar destinations, which may be thought of as customers. Of course, the heart of the matter lies in the definition of "similar." However this will be left undefined for the moment. Assume that

$$N = \{1, \ldots, n\} = N_1 \cup \ldots \cup N_p$$

where $N_k \cap N_\ell = \phi$ for $k \neq \ell$. Define $\overline{b}_k = \sum_{j \in N_k} b_j$ and

$\overline{c}_{ik} = \sum_{j \in N_k} b_j \, c_{ij} \, / \, \overline{b}_k$ for $k = 1, \ldots, p$. This specifies the aggregated model $\overline{M}$. The network is a directed graph with $m + p$ nodes, called sources (m) and destinations (p) with a link from each source to each destination. The characteristics are the (positive) supplies a_i, the (positive) demands $\overline{b}_k$, and the (nonnegative) unit costs $\overline{c}_{ik}$ on each link. The optimization problem is then:

Find $\bar{x}_{ik} \geqq 0$ (i=1, ..., m; k=1, ..., p) such that

$$\sum_i \bar{x}_{ik} = a_i \quad \text{and} \quad \sum_i \bar{x}_{ik} = \bar{b}_k \qquad \text{(feasibility)}$$

$$\text{and minimizing} \quad \sum_{i,k} \bar{c}_{ik}\, \bar{x}_{ik} \qquad \text{(optimality).}$$

The lifting operation may be defined in many ways. A direct (and rather bad) operation is

$$x_{ij} = b_j\, \bar{x}_{ik} / \bar{b}_k \quad \text{for} \quad j \in N_k .$$

This insures feasibility of the resulting shipments since:

$$\sum_j x_{ij} = \sum_j b_j\, \bar{x}_{ik} / \bar{b}_k$$

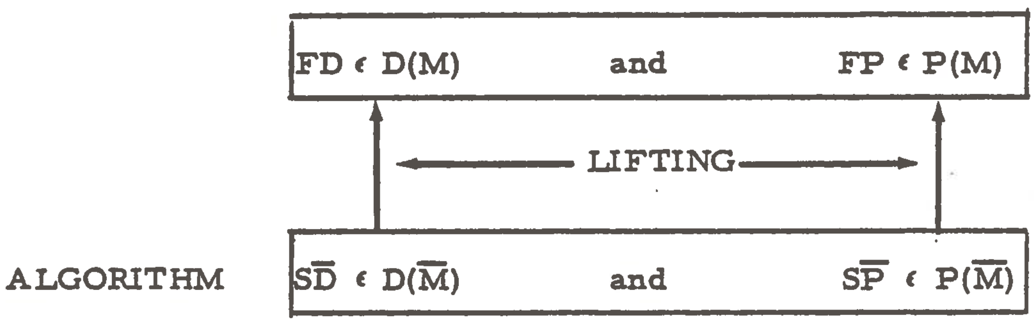
$$= (\sum_j b_j)\, (\sum_k \bar{x}_{ik}) / \bar{b}_k = a_i$$

$$\sum_j x_{ij} = b_j\, (\sum_i \bar{x}_{ik}) / \bar{b}_k = b_j .$$

3. Duality and Error Bounding

For all of the models that have been used prominently in transportation planning, the optimization problem may be formulated as a mathematical program which has a dual mathematical program. Feasible solutions for the primal and dual programs are denoted by P(M) and D(M), respectively. The principal property of these programs that we shall use is the following:

Value of FD $\leqq$ Optimal value for M $\leqq$ Value of FP for any FD $\in$ D(M) and any FP $\in$ P(M). This basic property can be used to provide a bound on the error from an aggregation method, provided that the optimal solutions for both the primal and dual for $\overline{M}$ can be lifted to feasible solutions for both the primal and dual of M. Schematically, this is represented by the following diagram:



A bound for the error in the aggregated solution is provided by:

$$\% \text{ of error in FP} = \frac{\text{value of FP} - \text{value of FD}}{\text{value of FP}} \times 100$$

The difficulty in applying this basically simple idea to actual aggregation methods consists in

(1) formulating the dual program in such a way that the solution is readily available;

(2) defining efficient lifting rules that preserve as much of the information in the optimal solution as possible.

The ideas of this section can be carried out with complete rigor for the Hitchcock-Koopmans Transportation Problem. The dual program is:

Find u_i and v_j $(i - 1, \ldots, m; j = 1, \ldots, n)$ such that

$$u_i + v_j \leqq c_{ij}$$

and maximizing $\sum_i a_i u_i + \sum_j b_j v_j$.

When aggregated as above, the best lifted feasible solution for the dual program is defined to be:

$$u_i = \overline{u}_i \qquad (i = 1, \ldots, m)$$

$$v_j = \min_i (c_{ij} - \overline{u}_i) \qquad (j = 1, \ldots, n) .$$

This definition satisfies all of the requirements of the diagram above and hence the error bound holds.

4. An Example

To demonstrate the effectiveness of the ideas introduced above, a (geometric) Transportation Problem was constructed with 3 sources and 100 destinations. The 3 sources (and supplies) are shown in Figure 1. (Precisely, $a_1 = 87$, $a_2 = 132$, $a_3 = 258$.) The 100 destinations (and demands) are shown in Figure 2. The unit costs are calculated as rectangular distances in the unit grid, that is, if two points have coordinates (x_1, y_1) and (x_2, y_2), then the distance between them is $|x_1 - x_2| + |y_1 - y_2|$. Thus, the distances from the upper left hand square to the three sources are 8, 8, and 15 respectively.

A "natural" aggregation of the 100 destinations into 4 aggregated destinations is shown in Figure 3. This aggregation is not too dissimilar to actual techniques for creating "zones" or "centroids" in transportation

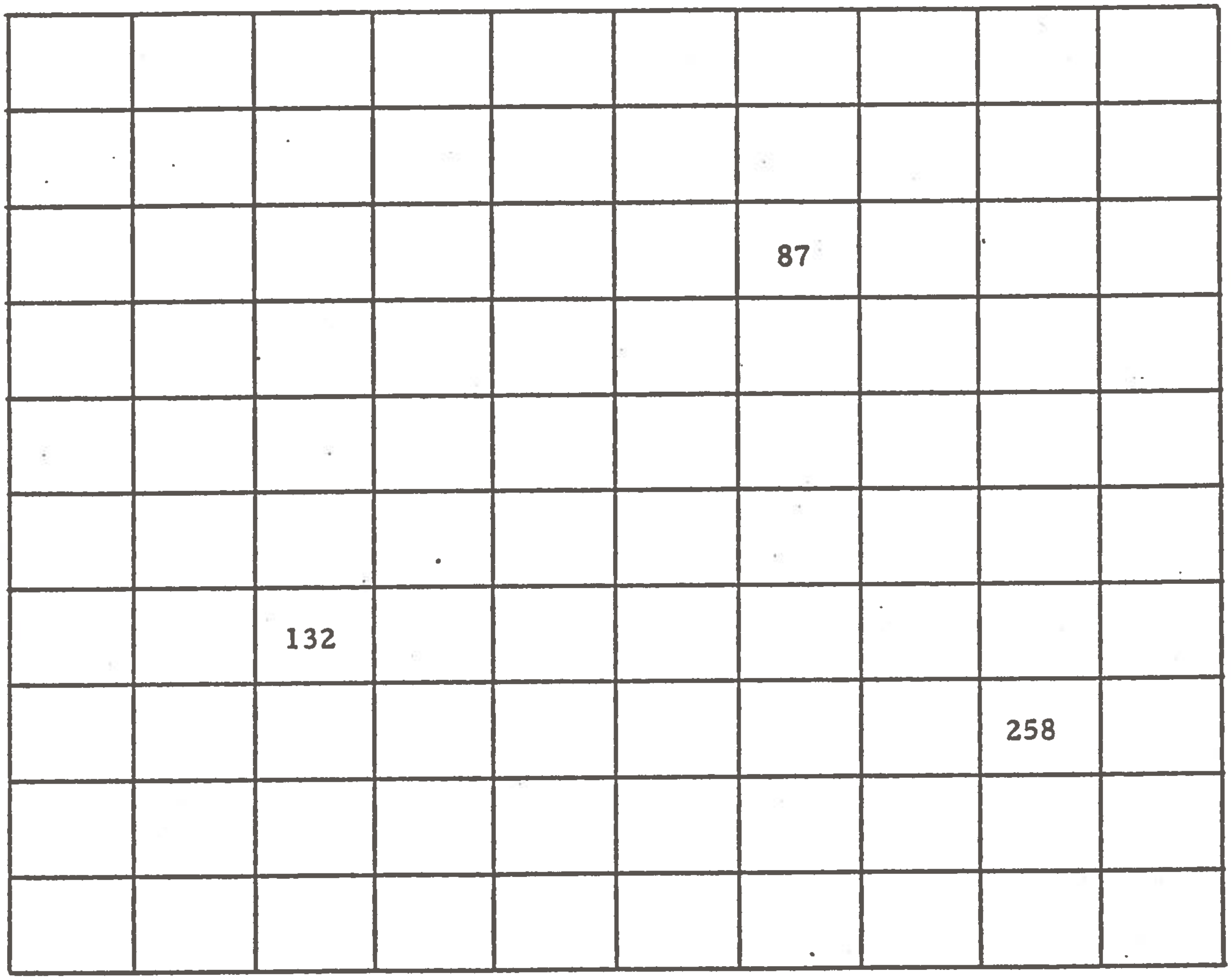


FIGURE 1. LOCATION OF SOURCES FOR ILLUSTRATIVE TRANSPORTATION PROBLEM

3*	1*	4*	1	5	4	6	3	5	2
9*	2*	6*	5	4	9	6	4	3	3
1*	7*	7*	2	4	6	5	3	6	4
5	3	8	5	1	6	6	4	6	2
1	6	4	9	7	6	2	2	8	3
2	5	8	8	1	3	1	8	3	9
9	4	5	1	3	8	8	6	2	1
9	3	7	7	9	3	7	4	8	2
7	6	5	2	5	2	2	7	3	4
3	9	4	7	2	1	8	7	6	9

FIGURE 2. LOCATION OF DESTINATIONS FOR ILLUSTRATIVE TRANSPORTATION PROBLEM

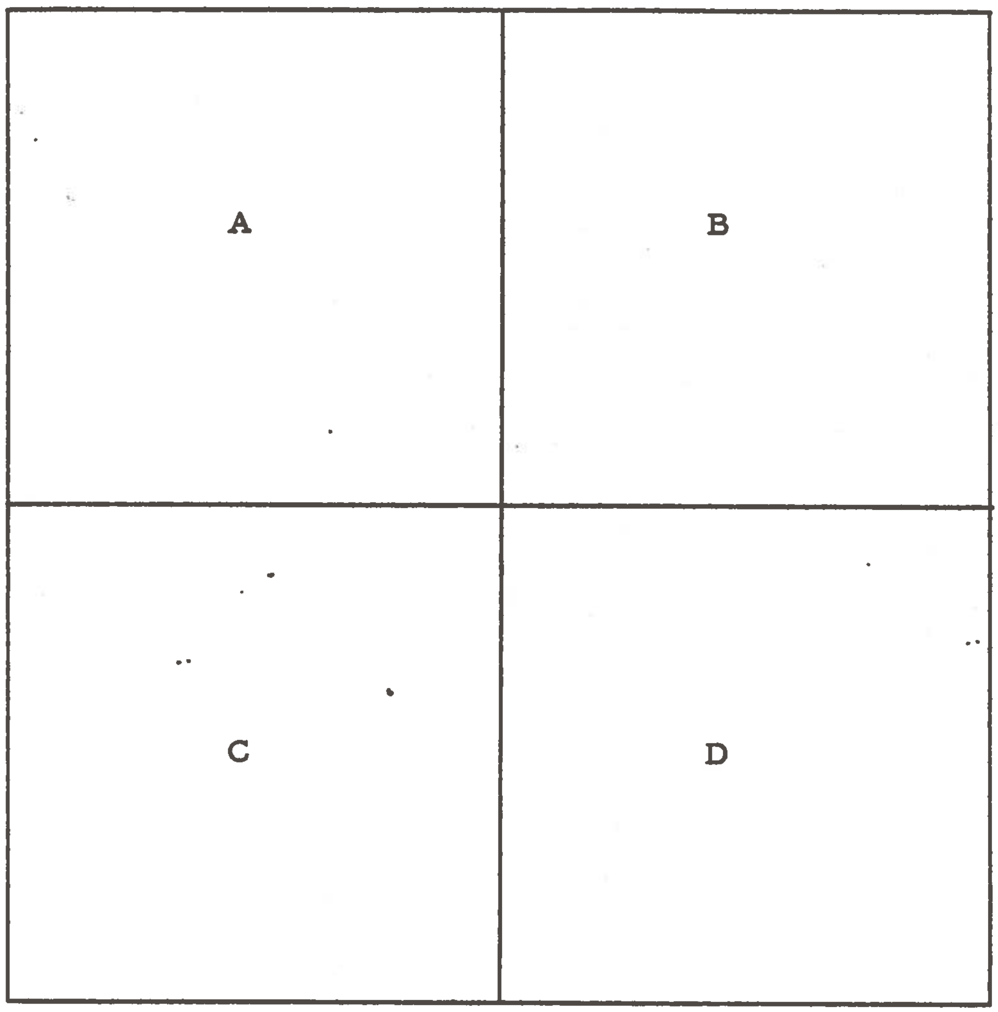


FIGURE 3. "NATURAL" AGGREGATION OF DESTINATION FOR ILLUSTRATIVE TRANSPORTATION PROBLEM

planning. The result of this aggregation is the model $\overline{M}$ presented below, in which the left column lists supplies a_i, the top row lists the demands $\overline{b}_k$, and the entries are the unit costs $\overline{c}_{ik}$ (see p. 26).

	A	B	C	D
	110	114	131	122
87	5.12	2.42	9.21	6.43
132	4.90	8.72	2.56	6.55
258	10.74	6.51	7.27	2.57

This 3 by 4 Transportation Problem is easily solved. The optimal solutions for primal and dual are exhibited below, in which the left column lists u_i, the top row lists v_k, and the entries are the shipments $\overline{x}_{ik}$.

	A	B	C	D
	4.90	1.80	2.56	-2.14
0.22	87			
0	23		109	
4.71		114	22	122

If the primal is lifted by the rule on p. 27, the total cost is 2052.8. (This can be improved by allocating the purchases by customers in A and C to the nearer warehouse as far as possible. The resulting value of the lifted primal is 1925.68.) If the optimal dual solution is lifted by the rule on p. 29, the value is 1762.53. Thus the error bounds are 14% and 8%, respectively. <u>By aggregation, a 3 by 100 Transportation Problem has been solved within 8% by lifting the optimal solution of a 3 by 4 problem.</u>

The dual problem suggests a geometric aggregation that is better than the quadrants of Figure 3. The decision of any customer j will not be changed if all of his costs are changed by the same constant $-c_j$. By the transformation

$$(c_{1j}, c_{2j}, c_{3j}) \longrightarrow (c_{1j} - c_j, c_{2j} - c_j, c_{3j} - c_j)$$

for $3c_j = c_{1j} + c_{2j} + c_{3j}$, we can "normalize" all customer costs to sum to zero. Two customers who were distinct before may now have the same costs. As an example, the 9 starred customers in Figure 2 all have the normalized costs $\left(-\frac{7}{3} - \frac{7}{3}, \frac{14}{3}\right)$. The geometry of this transformation is shown in Figure 4, where the original 100 customers have only 40 distinct normalized cost vectors. The numbers give total demands at these points in the plane $c_1 + c_2 + c_3 = 0$. For example the 9 starred customers in Figure 2 are shown as one customer with demand 40 (shown with a star).

To summarize, the numbers in Figure 4 give the total demand for each normalized cost vector graphed. This is an exact aggregation of the original 100 destinations into 40 destinations.

Figure 4, combined with the dual variables from the previous aggregation, suggests a second level aggregation by the following reasoning. We have already noted that the optimal shipments are not changed by transformations of the form

$$(c_{1j}, c_{2j}, c_{3j}) \longrightarrow (c_{1j} - c_j, c_{2j} - c_j, c_{3j} - c_j).$$

The same statement holds for transformations of the form

$$(c_{1j}, c_{2j}, c_{3j}) \longrightarrow (c_{1j} - u_1 - c_j, c_{2j} - u_2 - c_j, c_3 - u_3 - c_j).$$

Moreover, if the resulting points are projected in the plane $c_1 + c_2 + c_3 = 0$ by the correct choice of c_j, optimality is assured if and only if we choose (u_1, u_2, u_3) so that the demands b_j can be allocated to the three closed sets $\overline{A}_1$, $\overline{A}_2$, $\overline{A}_3$ that are the projections of the three quadrants:

$$A_1 = \left\{(c_1, c_2, c_3) \Big/ c_1 = 0, \; c_2 \geqq 0, \; c_3 \geqq 0\right\}$$

$$A_2 = \left\{(c_1, c_2, c_3) \Big/ c_1 \geqq 0, \; c_2 = 0, \; c_3 \geqq 0\right\}$$

$$A_3 = \left\{(c_1, c_2, c_3) \Big/ c_1 \geqq 0, \; c_2 \geqq 0, \; c_3 = 0\right\},$$

so that a total of a_i falls in $\overline{A}_i$ for $i = 1, 2, 3$. This is illustrated in Figure 5 where the optimal values of $(u_1, u_2, u_3) = (0, 0, 5)$ are used and demands which fall on the boundary of the resulting frame are split as shown. This yields:

Optimal value of $P(M) \big(\text{or } D(M)\big) = 1780,$

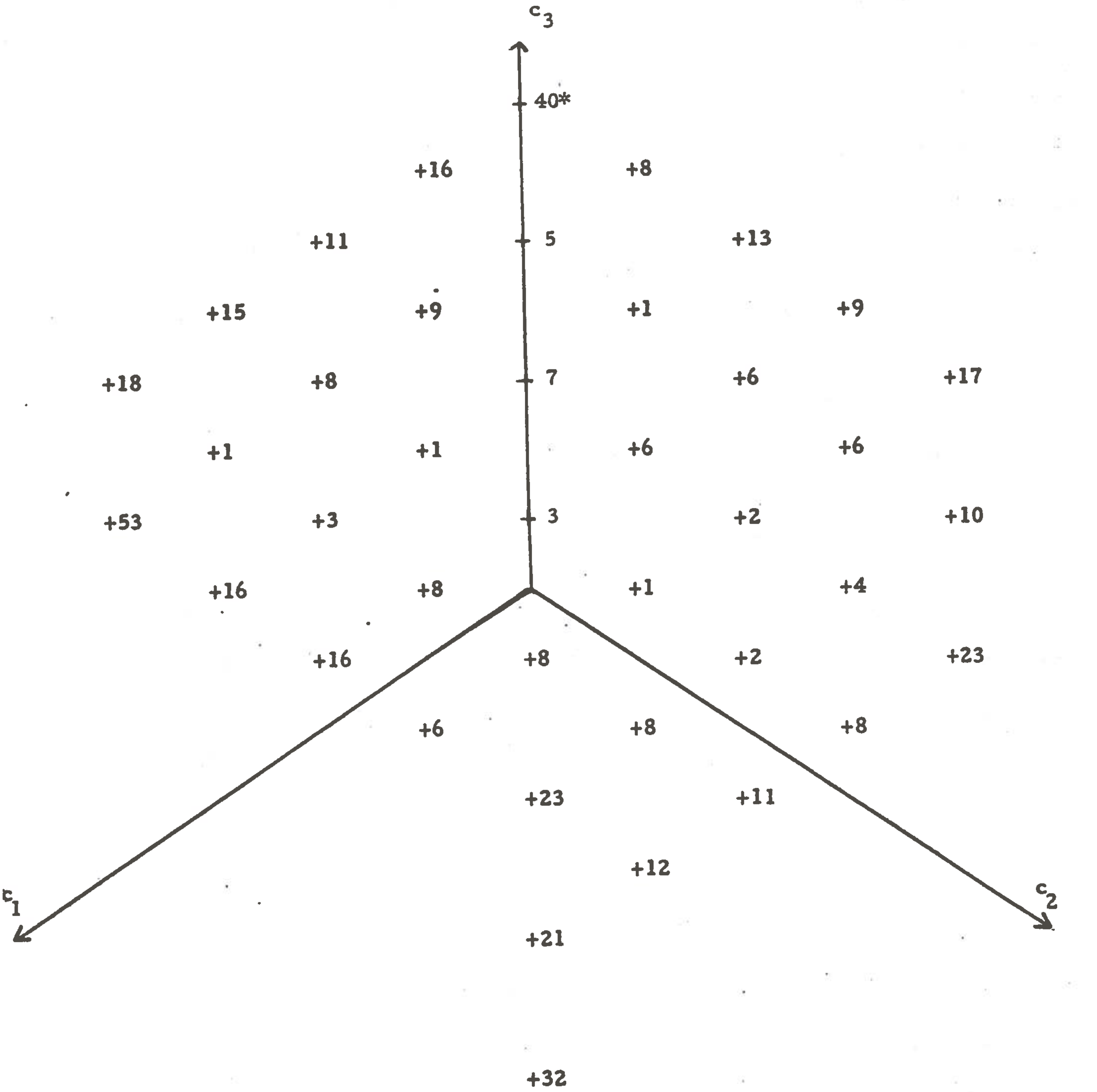


FIGURE 4. AGGREGATION OF DESTINATIONS SUGGESTED BY THE DUAL PROGRAM

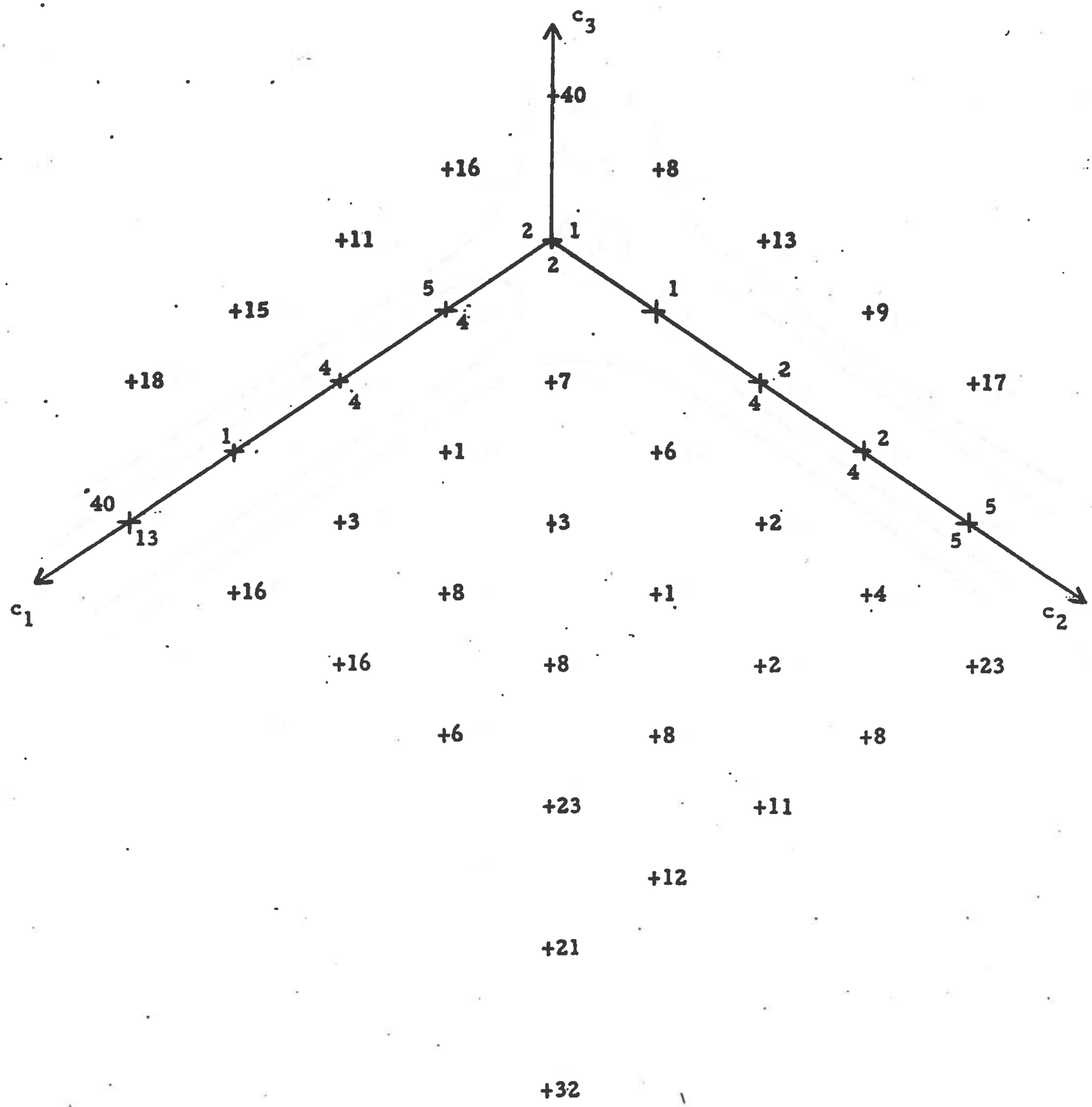


FIGURE 5. SECOND LEVEL AGGREGATION

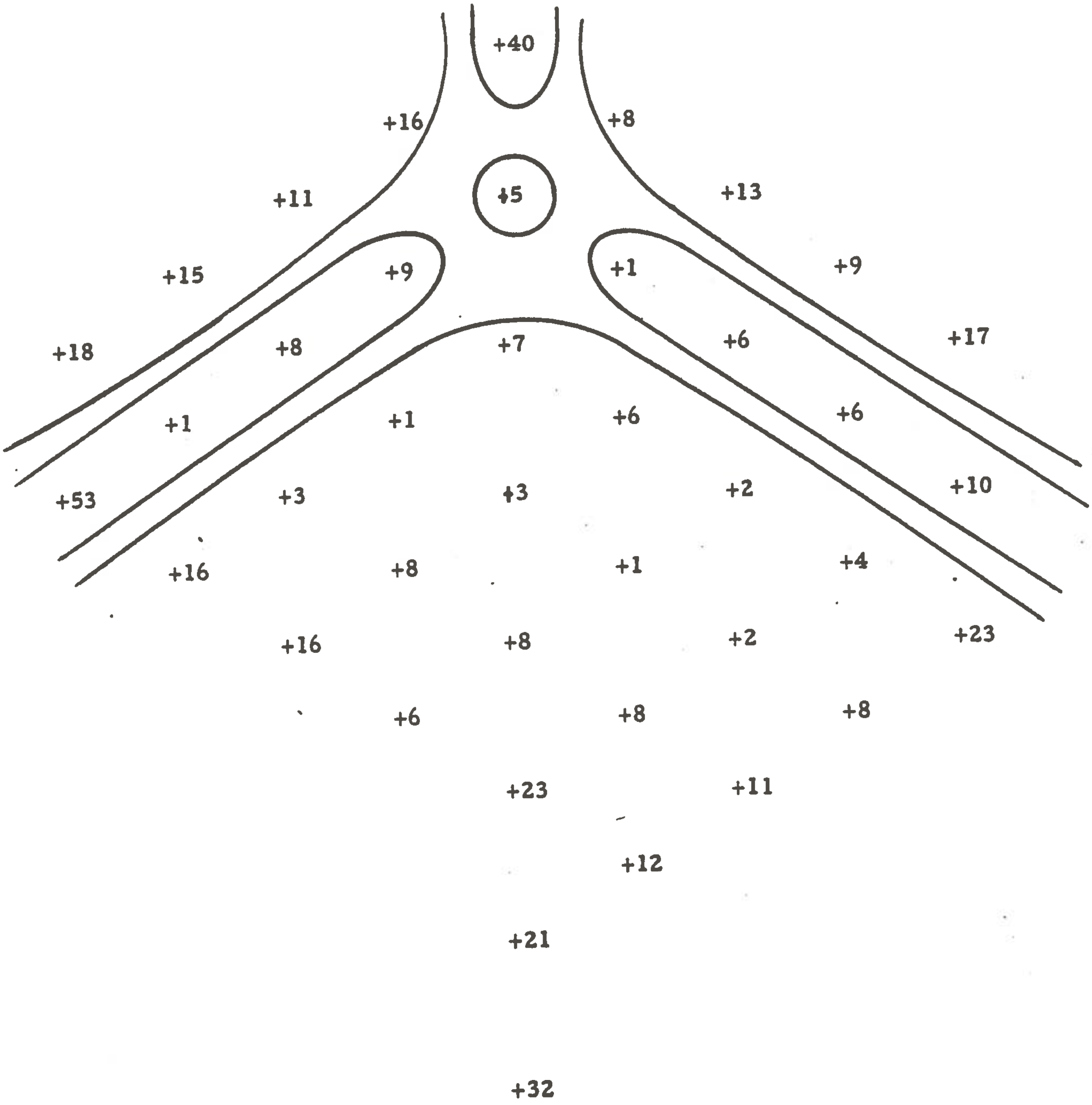


FIGURE 6. AGGREGATION INDUCED BY OPTIMAL DUAL SOLUTION

and reveals that the lifted value of the dual computed previously (1762.52) was very good indeed.

The last calculation depended on having available an optimal solution for $D(M)$, and hence is not allowed with the information available. However, we are entitled to use the lifted optimal solution for $D(\overline{M})$! Of course, we will not in general be able to achieve optimality by distributing the proper amounts to $\overline{A}_1$, $\overline{A}_2$, $\overline{A}_3$. However, we can enlarge these sets by a given tolerance (the amount 0.5 is appropriate in the present case) and create aggregated destinations corresponding to the following sets:

Destination	Set
1/2/3	$\overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3$.
1/2	$\overline{A}_1 \cap \overline{A}_2 - \overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3$
1/3	$\overline{A}_1 \cap \overline{A}_3 - \overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3$
2/3	$\overline{A}_2 \cap \overline{A}_3 - \overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3$
1	$\overline{A}_1 - (\overline{A}_1 \cap \overline{A}_1 + \overline{A}_1 \cap \overline{A}_3) + \overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3)$
2	$\overline{A}_2 - (\overline{A}_1 \cap \overline{A}_2 + \overline{A}_2 \cap \overline{A}_3) + \overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3)$
3	$\overline{A}_3 - (\overline{A}_1 \cap \overline{A}_3 + \overline{A}_2 \cap \overline{A}_3) + \overline{A}_1 \cap \overline{A}_2 \cap \overline{A}_3)$

This duality induced aggregation is shown in Figure 6. Returning to our original geometry, the assignment to aggregated customers is shown in Figure 7. The aggregated model $\overline{M}$ is displayed below:

1/2	1/2	1/2	1	1	1	1	1/3	3	3
1/2	1/2	1/2	1	1	1	1	1/3	3	3
1/2	1/2	1/2	1	1	1	1	1/3	3	3
2	2	2	1/2/3	1/3	1/3	1/3	3	3	3
2	2	2	2/3	3	3	3	3	3	3
2	2	2	2/3	3	3	3	3	3	3
2	2	2	2/3	3	3	3	3	3	3
2/3	2/3	2/3	3	3	3	3	3	3	3
2/3	2/3	2/3	3	3	3	3	3	3	3
2/3	2/3	2/3	3	3	3	3	3	3	3

FIGURE 7. ASSIGNMENT TO AGGREGATED CUSTOMERS, IN ORIGINAL GEOMETRY

	1	2	3	1/2	1/3	2/3	1/2/3
	57	60	221	40	23	71	5
87	2.60	7.45	6.07	5.73	1.78	9.62	4
132	7.77	2.28	6.57	5.73	7.96	2.86	4
258	9.19	9.28	3.77	12.73	6.78	7.86	9

This 3 by 7 Transportation Problem has the optimal solution exhibited below:

	2.60	2.28	-1.23	5.73	1.78	2.86	4
0	57			30			
0		60		10		62	
5			221		23	9	5

The value of the lifted primal (using the unsophisticated method of Section 2) is 1796.4. The value of the lifted dual is 1780. Hence, the error is less than 1%. Finally, if the sophisticated lifting of the primal is used, then the lifted primal feasible solution is optimal. That is, by solving a 3 by 4 and a 3 by 7 problem, we have solved a 3 by 100 problem exactly.

5. Automatic Multilevel Aggregation

The example above was started with a "natural" geometric aggregation into four zones. However, the subsequent aggregation by dual variables suggests that this is not necessary and that we may start with the most gross initial aggregation of all destinations into <u>one</u> destination. Subsequent aggregations are carried out by guides provided by the dual variables. The course of the calculations is illustrated below (using only unsophisticated lifting):

LEVEL 1:

	477
87	5.93
132	5.59
258	6.69

→ Optimal

	5.59
0.34	87
0	132
1.10	258

LIFTED PRIMAL: 2979.4

LIFTED DUAL: 1486.5

ERROR BOUND: 50%

LEVEL 2:

	181	151	145
87	3.80	8.59	6.47
132	7.90	2.63	6.14
258	8.59	8.15	2.78

→ Optimal

	3.07	2.63	-2.74
0.73	87		
0		132	
5.52	94	19	145

LIFTED PRIMAL: 2043.2

LIFTED DUAL: 1726.7

ERROR BOUND: 15%

LEVEL 3: SAME AS 3 BY 7 PROBLEM ABOVE; ERROR BOUND: 1%.

PAPER 2

BOUNDING AGGREGATION ERROR IN THE EQUILIBRIUM MODEL

1. Introduction

In Paper 1 we have seen the application of a new technique for bounding aggregation error and for using dual variables to guide efficient and accurate aggregation. The purpose of this paper is the generalization of this technique to the traffic assignment model.

The fundamental ingredients for this technique are described in detail in Section 3 of Paper 1. We shall not repeat this here but note that the following elements are necessary for its successful application:

(1) The original model must be formulated as a (primal) mathematical program;

(2) This primal program must possess a dual program that can be calculated in some form from the data;

(3) Primal and dual programs should enjoy the properties (using the notation of Paper 1):

$$\text{Value of FD} \leqq \text{Value of FP}$$

and equality should hold for optimal solutions;

(4) The algorithm that solves the model should provide optimal primal and dual solutions;

(5) An aggregation method should be available with a lifting rule that takes optimal solutions for the primal and dual aggregated model versions into (good) feasible solutions for the disaggregated model.

When these five ingredients are present, a rigorous bound can be found for the lifted feasible solutions for the disaggregated primal model.

Paper 1 showed that all five features are readily available for the Hitchcock-Koopmans Transportation Problem. In the remaining sections of this paper, we shall show by theory and example that the equilibrium problem has features (1) - (3) and shall indicate some suggestions on how to complete these by (4) and (5) to a complete theory.

2. A Sketch of Fenchel Duality Theory

The purpose of this section is to describe the necessary elements of Fenchel duality theory for reference later in the paper. (These ideas were introduced by Fenchel[1]. An excellent presentation can be found in Karlin[2] or Rockafellar[3].)

Suppose we seek to minimize a convex function over a convex set. For the purposes of the theory, this problem is set in a special form:

Suppose f is convex on the convex set C and g is concave on the convex set D. We seek

$$\inf_{x \in C \cap D} \left[f(x) - g(x) \right].$$

---

[1] W. Fenchel, Convex Cones, Sets, and Functions, Dept. of Mathematics, Princeton University, 1953.

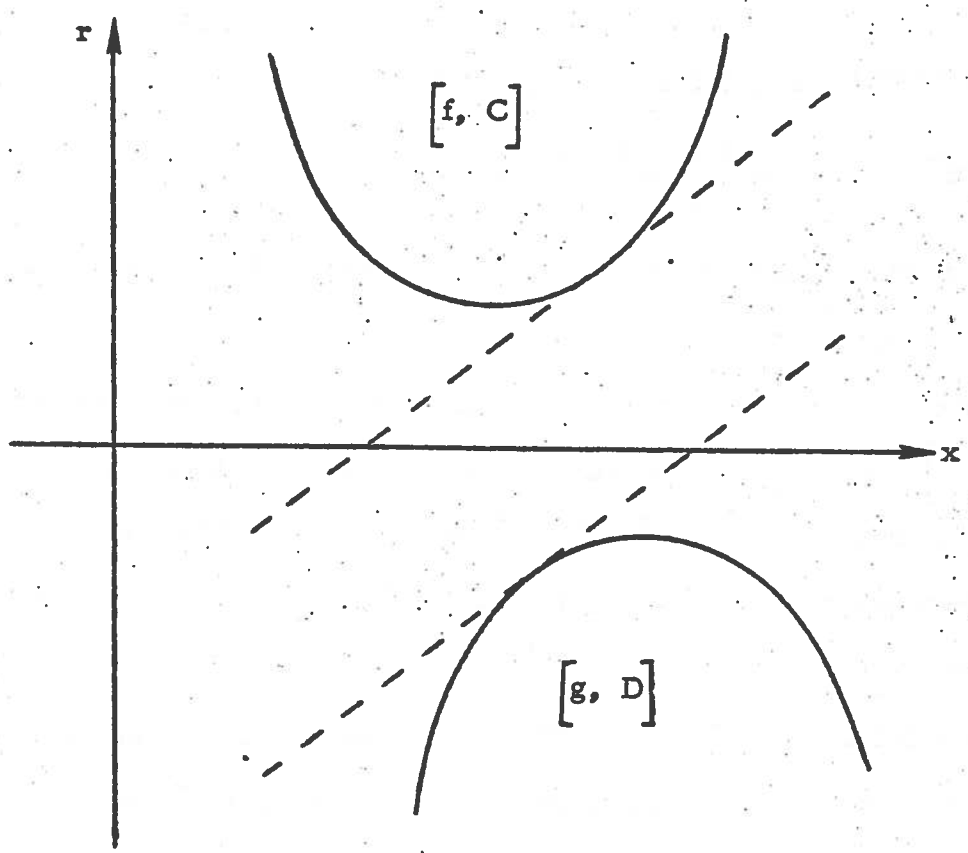
[2] S. Karlin, Mathematical Methods and Theory in Games, Programming, and Economics, Vol. I, Addison-Wesley, 1959.

[3] T. Rockafellar, Convex Analysis, Princeton University Press, 1970.

(In many applications, $g(x)$ is taken to be zero and the advantage of this generalization comes from the choice of D.) If we define the set set $[f, C]$ in $R \times X$ as

$$[f, C] = \left\{(r, x) \,\middle|\, x \in C, \; f(x) \geqq r\right\}.$$

then the problem can be interpreted as finding the minimum vertical separation between the sets $[f, C]$ and $[g, D]$ as shown in the figure below:



It is obvious that this is equal to the maximal vertical separation of two parallel hyperplanes separating $[f, C]$ and $[g, D]$. This relation, between a given minimization problem and an equivalent maximization problem is the duality we shall use.

The analytic tool for expressing this precisely is the concept of conjugate function (also due to Fenchel).

DEFINITION. Let f be a convex function defined on a convex set C. The <u>conjugate set</u>

$$C^* = \left\{ x^* \epsilon X^* \;\middle|\; \sup_{x \epsilon C} \left[\langle x, x^* \rangle - f(x) \right] < \infty \right\}$$

and the function f^* <u>conjugate</u> to f is defined as

$$f^*(x^*) = \sup_{x \epsilon C} \left[\langle x, x^* \rangle - f(x) \right] .$$

(Throughout this definition, x^* denotes a linear function and $\langle x, x^* \rangle$ denotes inner product.)

Since $-f^*(x^*)$ is the vertical distance to a support hyperplane below $[f, C]$ and $-g^*(x^*)$ is the vertical distance to the parallel support hyperplane above $[g, D]$, $g^*(x^*) - f^*(x^*)$ is the vertical separation of the two hyperplanes. The duality principle stated above leads to the following theorem.

FENCHEL DUALITY THEOREM. Assume that f and g are, respectively, convex and concave functions on the convex sets C and D in a space X. Assume that $C \cap D$ contains points in the relative interior of C and D and that either $[f, C]$ or $[g, D]$ has nonempty interior. Suppose further that

$$\inf_{x \in C \cap D} \left[f(x) - g(x) \right] < \infty$$

then $$g^*(x^*) - f^*(x^*) \leqq f(x) - g(x)$$

for all "feasible" $x \in C \cap D$ and $x^* \in C^* \cap D^*$ and

$$\max_{x^* \in C^* \cap D^*} \left[g^*(x^*) - f^*(x^*) \right] = \inf_{x \in C \cap D} \left[f(x) - g(x) \right] .$$

This theorem provides exactly what is needed for requirement (3) stated in the Introduction. Of course, to carry out the construction explicitly we must compute conjugate functionals having chosen C and D appropriately. This is exactly what we shall do in the next section.

3. Duality for Equilibrium Models

For the discussion of equilibrium models we shall use the following notation:

OD pair: $k \in \{1, \ldots, p\}$.

Path joining OD pair k: $kj \in \{k1, \ldots, kn_k\}$.

Link: $\ell \in \{1, \ldots, q\}$.

Link-path incidence matrices: $\left(A^{(1)}, \ldots, A^{(p)}\right)$

$A^{(k)} = \left(a^{(k)}_{\ell j}\right)$ where $a^{(k)}_{\ell j} = 1$ if link ℓ is on path kj joining OD pair k and $a^{(k)}_{\ell j} = 0$ otherwise.

Demand on OD pair k: d_k .

Flow on path kj : x_{kj} .

Induced flow on link ℓ: $x_\ell = \sum_k \left(\sum_j a_{\ell j}^{(k)} x_{kj} \right)$

User cost for flow x_ℓ : $C_\ell(x_\ell) \geqq 0$ increasing.

Cost of path kj joining OD pair k induced by flows

$$(x_{kj}) : C_{kj}(x_{kj}) = \sum_\ell a_{\ell j}^{(k)} C_\ell (x_\ell) .$$

EXAMPLE: $p = 1, \quad n_1 = 3, \quad q = 5, \quad d_1 = 6$

$$A^{(1)} = \begin{bmatrix} 1 & 0 & 0 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

$$C_1(x) = C_4(x) = 50 + x, \quad C_2(x) = C_3(x) = 10x, \quad C_5(x) = 10 + x.$$

(This is the description of the example used by Steenbrink to illustrate the paradox of Braess.) It is well-known that the descriptive assignment (user optimized) and normative assignment (system optimized) can be formulated as mathematical programs. These are stated below:

DESCRIPTIVE ASSIGNMENT: Choose (x_{kj}) so as to minimize

$$\sum_\ell \int_0^{x_\ell} C_\ell(x)\, dx \quad \text{subject to} \quad x_\ell = \sum_k \left(\sum_j a_{\ell j}^{(k)} x_{kj} \right)$$

$$\sum_j x_{kj} = d_k \quad \text{and} \quad x_{kj} \geqq 0.$$

NORMATIVE ASSIGNMENT: Choose (x_{kj}) so as to minimize

$$\sum_\ell C_\ell(x) x_\ell \quad \text{subject to} \quad x_\ell = \sum_k \left(\sum_j a_{\ell j}^{(k)} x_{kj} \right)$$

$$\sum_j x_{kj} = d_k \quad \text{and} \quad x_{kj} \geqq 0.$$

EXAMPLE (continued): The two problems (stated in path variables) are:

DESCRIPTIVE ASSIGNMENT: Choose (x_{11}, x_{12}, x_{13}) so as to minimize

$$1/2\,(11x_{11}^2 + 21x_{12}^2 + 11x_{13}^2 + 20x_{11}x_{12} + 20x_{12}x_{13} + 100x_{11} + 20x_{12} + 100x_{13})$$

subject to:

$$x_{11} + x_{12} + x_{13} = 6.$$
$$x_{11} \geqq 0, \quad x_{12} \geqq 0, \quad x_{13} \geqq 0.$$

NORMATIVE ASSIGNMENT: Choose (x_{11}, x_{12}, x_{13}) so as to minimize:

$$11x_{11}^2 + 21x_{12}^2 + 11x_{13}^2 + 20x_{11}x_{12} + 20x_{12}x_{13} + 50x_{11} + 10x_{12} + 50x_{13}$$

subject to:

$$x_{11} + x_{12} + x_{13} = 6$$
$$x_{11} \geqq 0, \quad x_{12} \geqq 0, \quad x_{13} \geqq 0.$$

To illustrate the construction of a dual for the equilibrium model, we shall first transform the DESCRIPTIVE ASSIGNMENT from path variables to link variables. Note that:

$$x_1 = x_{11}, \quad x_2 = x_{11} + x_{12}, \quad x_3 = x_{12} + x_{13}, \quad x_4 = x_{13}, \quad x_5 = x_{12}$$

and hence the constraints on the path flows

$$x_{11} + x_{12} + x_{13} = 6$$
$$x_{11} \geqq 0, \quad x_{12} \geqq 0, \quad x_{13} \geqq 0$$

are equivalent to the constraints on the link flows:

$$
\begin{array}{lllllll}
x_1 & & & + x_4 & + x_5 & = & 6 \\
x_1 & - x_2 & & & + x_5 & = & 0 \\
 & & - x_3 & + x_4 & + x_5 & = & 0
\end{array}
$$

$$x_1 \geqq 0,\ x_2 \geqq 0,\ x_3 \geqq 0,\ x_4 \geqq 0,\ x_5 \geqq 0.$$

DESCRIPTIVE ASSIGNMENT (Link variables) Find (x_ℓ) so as to minimize

$$(50x_1 + 1/2x_1^2) + 5x_2^2 + 5x_3^2 + (50x_4 + 1/2x_4^2) + (10x_5 + 1/2x_5^2)$$

subject to the constraints above.

It is clear that all three of the problems formulated above are quadratic programs with strictly convex objective functions to be minimized and constraints that are linear. This will be true in general for models with linear user costs on the links. This also means that the dual programs are readily available in various forms from the special theory of quadratic programming. However, we wish to use this example as a paradigm for the construction of a dual by the Fenchel duality theory. We shall do this for the DESCRIPTIVE ASSIGNMENT with link variables.

EXAMPLE (continued):

Let $f(x) = (50x_1 + 1/2x_1^2) + 5x_2^2 + 5x_3^2 + (50x_4 + 1/2x_4^2)$
$\qquad + (10x_5 + 1/2x_5^2)$

$g(x) = 0$

$C = \{x \mid x \geqq 0\}$

$D = \{x \mid x_1 + x_4 + x_5 = 6,\ \ x_1 - x_2 + x_5 = 0,\ -x_3 + x_4 + x_5 = 0\}.$

The function conjugate to g is

$$g^*(x^*) = \sup_{x \epsilon D} \langle x^*, x \rangle$$

This is only finite when

$$x^* = y_1 (1,0,0,1,1) + y_2(1,-1,0,0,1) + y_3(0,0,-1,1,1)$$

and $g^*(x^*) = 6y_1$. Each of the summands of $f(x)$ have the form $ax + bx^2$. An elementary calculation shows that the conjugate function has the form $(x^* - a)^2/ 4b$. Hence the dual program is:

$$\text{Maximize } 6y_1 - \frac{(y_1 + y_2 - 50)^2}{2} - \frac{y_2^2}{20} - \frac{y_3^2}{20}$$

$$- \frac{(y_1 + y_3 - 50)^2}{2} - \frac{(y_1 + y_2 + y_3 - 10)^2}{2}$$

where y_1, y_2, and y_3 are unrestricted.

To complete this example, the unrestricted maximum problem is easily solved yielding

$$y_1 = 92, \quad y_2 = y_3 = -40.$$

These values determine values of x_ℓ (via the conjugate functions) as:

$$x_1 = 2, \; x_2 = 4, \; x_3 = 4, \; x_4 = 2, \; x_5 = 2.$$

The common value of the objective function is 386. (Of course, the link flows above correspond to path flows $x_{11} = x_{12} = x_{13} = 2$ with total user cost of 552.)

4. Conclusions

This example is typical of the problem of formulating the program dual to an equilibrium model and proves that it poses no intrinsic theoretical problem. Of course, the computation of the conjugate functions was specially simple for this example. However, the computation appears to be just as simple for any of the cost functions that have been used in the traffic assignment literature.

Two fascinating properties have appeared in the formulation of the dual. The dual set D* was spanned by two types of vectors. The first is (1, 0, 0, 1, 1) and is a cut set separating the OD pair. The second types are (1, -1, 0, 0, 1) and (0, 0, -1, 1, 1) and correspond to the conservation requirements at the nodes. It is conjectured that these properties hold for more general networks and that this fact will also yield useful information for the lifting of aggregated optimal solutions.

Secondly, the only dual variable that contributes to the linear part of the objective function is the cut set variable y_1. The value of the linear part at the optimum is 552, the user optimized cost. It is also conjectured that this property generalizes.

Finally, in order to carry out our complete program, we still need features (4) and (5). Paper 5 makes a start at formalized aggregation rules on test networks. As for algorithms which compute simultaneously primal and dual solutions, they are not available in the form in which they can be used at present. Therefore, further research is called for to carry out this program completely.

PAPER 3

A PATH EXTRACTION AGGREGATION ALGORITHM

1. Introduction

This paper presents a new algorithm for the network equilibrium model that works in the space of the path flows using a labelling and pivot technique. A detailed set of specifications is given, convergence to an optimal solution is proved, and estimates of computational efficiency are provided. Since the description that follows is quite concise, some account of the relation of this algorithm to other similar methods and of the relevance of this development to aggregation must be given.

The terms "labelling and pivoting" have been used in computing economic equilibrium for about eight years since the initial work of Scarf[1] in this area. Since then the methods have been developed and extended by a number of other mathematicians and economists. (A comprehensive survey by Kuhn[2] has a bibliography of 42 items.)

The idea behind the pivotal methods that is extended to the traffic assignment problem in this paper can be motivated by a classical model of equilibrium in an exchange economy. This economy trades n goods indexed by $k = 1, \ldots, n$ with prices $p_1, \ldots, p_n$. A price vector $p = (p_1, \ldots, p_n)$ generates supply and demand for each of these goods

---

[1] H. E. Scarf, "The Approximation of Fixed Points of a Continuous Mapping," SIAM J. Appl. Math. 15 (1967) 1328-43.

[2] H. W. Kuhn, "How to Compute Economic Equilibria by Pivotal Methods," Mathematics Department, Princeton University, January, 1975.

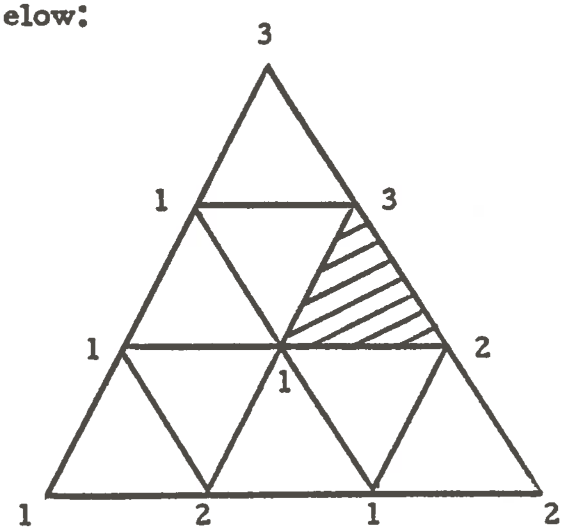
and their difference is called the "excess demand function" for each good k and is denoted by $g_k(p)$. These n functions are assumed to be continuous and homogeneous of degree zero for nonnegative, nonzero prices. Also, the prices and excess demands are assumed to satisfy Walras' Law which says that expenditures equal revenues when summed over all goods, that is,

$$p_1 g_1(p) + \dots + p_n g_n(p) = 0 \quad \text{for all} \quad p \geqq 0.$$

Equilibrium for such a model is a set of prices $\bar{p}$ that generate a supply that is not less than the demand for each good, that is, such that $g_k(\bar{p}) \leqq 0$ for $k = 1, \dots, n$. (Of course, Walras' Law implies that the price of a good with excess supply is zero.)

The algorithm works on the price simplex where prices are normalized to sum to one and starts with the standard subdivision of this simplex. The vertices of this subdivision are given labels by the following rule:

A price vector p is given the label $k \epsilon \{1, 2., \dots, n\}$ if the excess demand for good k is smallest among those goods with positive price. (In case of ties, choose the first.) A possible labelling for a case of 3 goods is shown below:



Of course, Walras' Law implies that:

(*) If the label of p is k then $g_k(p) \leqq 0$.

Furthermore, the rule for labelling implies that:

(**) If the label of p is k then $p_k > 0$.

Properties (**) is the hypothesis of Sperner's Lemma[1], which asserts the existence of at least one triangle with all three labels. (In the figure above, such a triangle is shaded.)

This is enough to approximate an equilibrium arbitrarily closely. For a fine enough subdivision, the points in the completely labelled piece satisfy

$$g_k(p) \leqq \epsilon$$

for given $\epsilon > 0$. Hence they are nearly equilibria and approach equilibria as ϵ tends to zero.

In the algorithm that follows, the ideas of this simple illustration are generalized and extended in the following ways:

(1) We work not on a price simplex but in the space of proportionate flows joining OD pairs. Geometrically, this is the Cartesian products of simplices.

(2) This product set is subdivided in an efficient way for computer calculation.

(3) The appropriate labelling for traffic assignment equilibrium is constructed.

---

[1] C. B. Tompkins, "Sperner's Lemma and Some Extensions" in Applied Combinatorial Mathematics, E. F. Beckenbach, ed., John Wiley & Sons, 1964.

(4) The technique of "restarting"[1] is generalized to cover this case.

The algorithm is intended to apply to networks for which an extremely accurate answer for equilibrium is desired. Furthermore, the algorithm presumes the aggregation by extraction of paths. In common sense terms, it is presumed that there are only a small number of candidate paths joining each OD pair. In the notation of the next section, current computational experience suggests that inexpensive and very accurate answers can be given to networks with $(n_1 + \ldots + n_p) - p \leq 100$. This will be extremely helpful in studying other aggregation methods.

It should be remarked that, although the intended application of this algorithm is to an extracted subset of the paths joining OD pairs, if all paths are used, then the algorithm constructs (in the limit) an exact equilibrium. Also, the extracted subset of the paths can easily be modified on the basis of intermediate calculations. For example, we may add a path if it is definitely cheaper than those in use or delete a path if it has a nearly zero flow at the current approximation.

---

[1] H. W. Kuhn and J. G. MacKinnon, "Sandwich Method for Finding Fixed Points," J. Optimization Theory and Appl. 17 (1975) 189-204.

2. Notation

OD pairs: $k = 1, \ldots, p$.

Path joining OD pair k: $j = 1, \ldots, n_k$.

Link: $\ell = 1, \ldots, q$.

Link-path incidence matrices: $(A^{(1)}, \ldots, A^{(p)})$

$A^{(k)} = a_{\ell j}^{(k)}$ where $a_{\ell j}^{(k)} = 1$ if link ℓ is on path j joining OD pair k and $a_{\ell j}^{(k)} = 0$ otherwise.

Demand on OD pair k: d_k

Proportionate flows on paths joining OD pair k:

$X_k = (x_{k1}, \ldots, x_{kn_k}) \in S_{n_k}$, that is, all $x_{kj} \geqq 0$ and $\sum_j x_{kj} = 1$.

Flow on path j joining OD pair k: $x_{kj} d_k$.

Induced flow on link ℓ:

$$f_\ell = \sum_k \left(\sum_j a_{\ell j}^{(k)} x_{kj} \right) d_k .$$

User cost for flow f_ℓ: $c_\ell(f) \geqq 0$ increasing.

Cost of path j joining OD pair k induced by proportionate flows $(X_1, \ldots, X_p)$:

$$c_{kj}(X_1, \ldots, X_p) = \sum_\ell a_{\ell j}^{(k)} c_\ell(f_\ell).$$

Label of proportionate flows $(X_1,\ldots,X_p)$:

$L(X_1,\ldots,X_p) = (j_1,\ldots,j_p)$ where

$c_{kj_k}(X_1,\ldots,X_p)$ is maximum of $c_{kj}(X_1,\ldots,X_p)$ over j

with $x_{kj} > 0$ (if ties, take j_k largest such index).

Integer label of proportionate flows $(X_1,\ldots,X_p)$:

If $L(X_1,\ldots,X_p) = (n_1,\ldots,n_p)$ then

$IL(X_1,\ldots,X_p) = n_1+\ldots+n_p - p + 1.$

If $L(X_1,\ldots,X_p) = (j_1,\ldots,j_p)$ and

$j_1 = n_1,\ldots,j_k = n_k,\ j_{k+1} < n_{k+1}$ for $0 \leqq k \leqq p-1$ then

$IL(X_1,\ldots,X_p) = n_1+\ldots+n_k+j_{k+1}-k.$

3. Preliminary Results

Theorem 1. If $n_1+\ldots+n_p-p+1$ proportionate flows have the integer labels $1,2,\ldots,n_1+\ldots+n_p-p+1$ then given j, $0 \leqq j \leqq n_k$, there exists a proportionate flow in the set with $L = (j_1,\ldots,j_p)$ and $j_k = j$. That is, completeness in integer labels implies completeness of the (vector) labels for each OD pair.

Theorem 2. If $\overline{X} = (\overline{X}_1,\ldots,\overline{X}_p)$ is a proportionate flow such that in every neighborhood there are a complete set of integer labels then $\overline{X}$ is a user-optimized equilibrium.

Proof. Suppose $\overline{x}_{kj} > 0$ and $\overline{x}_{kj'} > 0$ and $c_{kj}(\overline{X}) < c_{kj'}(\overline{X})$. Then, if $L(X_1,\ldots,X_p) = (j_1,\ldots,j_p)$, $j_k \neq j$ in a neighborhood of $\overline{X}$ since path j is never the most expensive path joining OD pair k in this neighborhood and $x_{kj} > 0$ there.

Hence $c_{kj}(\overline{X})$ is constant, say $c_k(\overline{X})$, for all j with $x_{kj} > 0$. Suppose $\overline{x}_{kj} = 0$ and $c_{kj}(\overline{X}) < \overline{c}$. Then, if $L(X_1,\ldots,X_p) = (j_1,\ldots,j_p)$, $j_k \neq j$ in a neighborhood of $\overline{X}$ since $j \neq j$ when $x_{kj} = 0$ and, when $x_{kj} > 0$, j cannot be the most expensive path. Thus we have verified the conditions of an equilibrium.

Theorem 3. Consider any subdivision of $S_{n_1} \times \ldots \times S_{n_p}$ into simplices of dimension $n_1+\ldots+n_p-p+1$ and give their vertices (vector) labels such that $L(X_1,\ldots,X_p) = (j_1,\ldots,j_p)$ implies $x_{kj_k} > 0$ for $k = 1,\ldots,p$. Then there exists a simplex of the subdivision with a complete set of induced integer labels.

4. Subdivision of $S_{n_1} \times \ldots \times S_{n_p}$

The following subdivision has been designed for efficient computer implementation. It is called a subdivision of degree $(D_1,\ldots,D_p)$ where the the D_k are positive integers. A vertex of the subdivision is specified by a vector of non-negative integers $(Z_1,\ldots,Z_p)$ where $Z_k = (z_{kj})$ and $\sum_j z_{nj} = D_k$ for $k = 1,\ldots,p$. (Note that $x_{kj} = D_k$ defines the corresponding proportionate flows; for computer work it is easier to work with integers.) A simplex of the subdivision is a set of N+1 vertices, where $N = n_1+\ldots+n_p-p$, that can be arranged as:

$$Z_0 = (Z_1,\ldots,Z_p)_0$$

$$Z_n = (Z_1,\ldots,Z_p)_N$$

and

$$Z_{n+1} = Z_n + (\ldots, E_{k,j+1} - E_{k,j}, \ldots) \text{ for } n=1,\ldots,N-1.$$

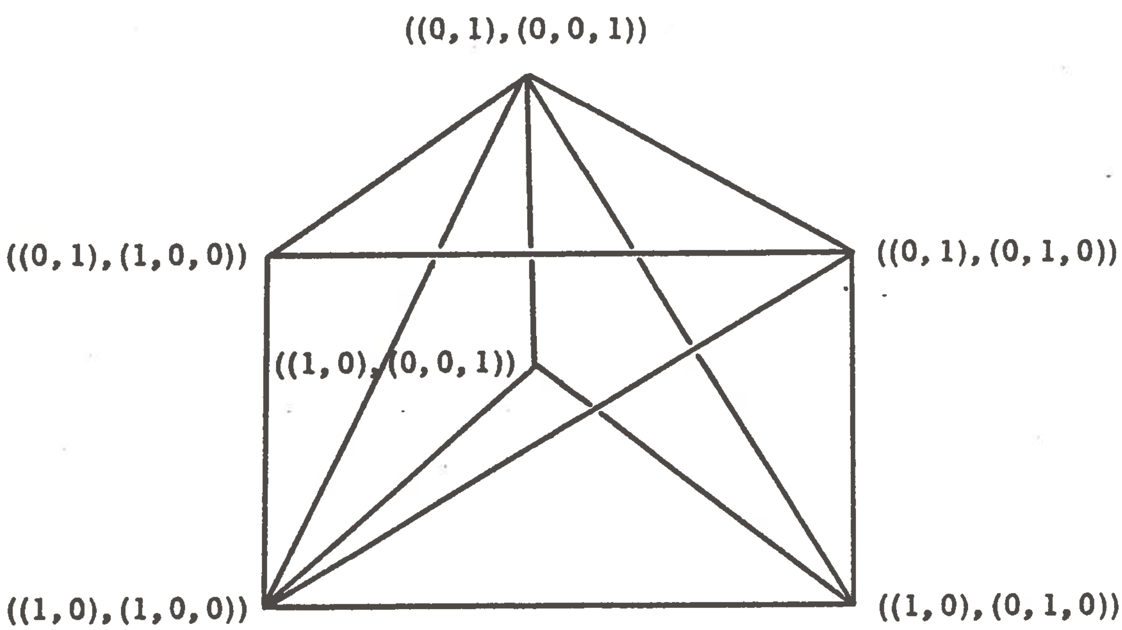
In this definition, $E_{k,j}$ is the j^{th} unit vector of S_{n_k}, and every kj is to appear in the sequence exactly once for $j=1,\ldots,n_k-1$.

This definition is illustrated by the following example: $S_2 \times S_3$ of degree (1, 1).

The three simplices of the subdivision are:

Z_0:	((1,0),(1,0,0))	((1,0),(1,0,0))	((1,0),(1,0,0))
Z_1:	((0,1),(1,0,0)) →	((1,0),(0,1,0))	((1,0),(0,1,0))
Z_2	((0,1),(0,1,0))	((0,1),(0,1,0)) →	((1,0),(0,0,1))
Z_3:	((0,1),(0,0,1))	((0,1),(0,0,1))	((0,1),(0,0,1))

This subdivision is shown in the following figure:



The most important feature of this subdivision is that it permits a trivially easy computation of the "pivot" step of dropping one vertex to pass to the adjacent simplex through the remaining face. If the vertices are ordered as above:

New vertex = Vertex before + Vertex after - Old vertex.

This is shown in the listing of the three simplices above by arrows pointing from the old vertex (out) to the new vertex (in).

It is easy to give a count of the number of pieces in a subdivision of degree $(D_1, \ldots, D_p)$. The product subdivision gives $D_1^{n_1-1} D_2^{n_2-1} \ldots D_p^{n_p-1}$ pieces and each is cut into $\frac{(n_1+\ldots+n_p-p)!}{(n_1-1)!\ldots(n_p-1)!}$ pieces. For example, a (10, 10, 10) subdivision of $S_2 \times S_2 \times S_2$ has 6,000 pieces. Fortunately, the algorithm examines only a small fraction of them.

5. The Algorithm

The algorithm works on a subdivision of degree $(D_1, \ldots, D_p, 1)$ of $S_{n_1} x \ldots x S_{n_p} \times S_2$. We use integer labelling as before on vertices of the form $(Z_1, \ldots, Z_p, (0,1))$. To define an integer labelling for vertices of the form $(Z_1, \ldots, Z_p, (1,0))$, let $(\overline{Z}_1, \ldots, \overline{Z}_p)$ be such that $(\overline{Z}_1+E_{n_1}, \ldots \overline{Z}_p+E_{n_p})$ is the best available non-negative integer approximation to the equilibrium of degree $(D_1, \ldots, D_p)$. We then define

$$L(Z_1, \ldots, Z_p, (1,0)) = (j_1, \ldots, j_p)$$

if j_k is first maximum of $z_{kj}-\overline{z}_{kj}$. This induces an integer label as before and we are able to pick a starting simplex of dimension $n_1+\ldots+n_p-p+1$ to begin the algorithm. It is listed below (with the integer labels IL at left):

IL	
1	$((\bar{z}_{11}+1,\bar{z}_{12},\ldots,\bar{z}_{1n_1}),(\bar{z}_{21}+1,\bar{z}_{22},\ldots,\bar{z}_{2n_2}),\ldots,(\bar{z}_{p1}+1,\bar{z}_{p2},\ldots\bar{z}_{pn_p}),(1,0))$
2	$((\bar{z}_{11},\bar{z}_{12}+1,\ldots,\bar{z}_{1n_1}),(\bar{z}_{21}+1,\bar{z}_{22},\ldots,\bar{z}_{2n_2}),\ldots,(\bar{z}_{p1}+1,\bar{z}_{p2},\ldots,\bar{z}_{pn_p}),(1,0))$
	$\vdots$
n_1	$((\bar{z}_{11},\bar{z}_{12},\ldots,\bar{z}_{1n_1}+1),(\bar{z}_{21}+1,\bar{z}_{22},\ldots,\bar{z}_{2n_2}),\ldots,(\bar{z}_{p1}+1,\bar{z}_{p2},\ldots,\bar{z}_{pn_p}),(1,0))$
n_1+1	$((\bar{z}_{11},\bar{z}_{12},\ldots,\bar{z}_{1n_1}+1),(\bar{z}_{21},\bar{z}_{22}+1,\ldots,\bar{z}_{2n_2}),\ldots,(\bar{z}_{p1}+1,\bar{z}_{p2},\ldots,\bar{z}_{pn_p}),(1,0))$
	$\vdots$
$+\ldots+n_p-p+1$	$((\bar{z}_{11},\bar{z}_{12},\ldots,\bar{z}_{1n_1}+1),(\bar{z}_{21},\bar{z}_{22},\ldots,\bar{z}_{2n_2}+1),\ldots,(\bar{z}_{p1},\bar{z}_{p2},\ldots,\bar{z}_{pn_p}+1),(1,0))$
?	$((\bar{x}_{11},\bar{z}_{12},\ldots,\bar{z}_{1n_1}+1),(\bar{z}_{21},\bar{z}_{22},\ldots,\bar{z}_{2n_2}+1),\ldots,(\bar{z}_{p1},\bar{z}_{p2},\ldots,\bar{z}_{pn_p}+1),(0,1))$

As is customary in pivotal algorithms, when the new label (marked "?") is computed, the vertex with a repeated label is dropped. A new vertex enters by the rules given before and the algorithm continues. The success of the algorithm depends on the following theorem.

<u>Theorem 4</u>. The algorithm can only pass through the boundaries of $S_{n_1}\times\ldots\times S_{n_p}\times S_2$ through the face $S_{n_1}\times\ldots\times S_{n_p}\times(0,1)$. This is signalled by the appearance of a vertex $(Z_1,\ldots,Z_p,(-1,2))$. When this occurs, the remaining vertices in the face carry a complete set of integer labels,

$\{1,\ldots,N+1\}$. By Theorem 2, for fine enough subdivisions, these vertices approximate a user-optimized equilibrium.

Of course, this theorem gives a constructive proof of Theorem 3. The algorithm is intended to be used as the Sandwich method, restarting from each approximation by increasing the degrees by an integer factor. Experience with the Sandwich Method indicates that the algorithm should be practical for problems for which $n_1+\ldots+n_p-p$ can be as large as 100. For example, 100 OD pairs with two paths

EXAMPLE: (Potts-Oliver, pp. 96-100)

$k = 1,2,3,4$ (that is, four OD pairs).

$n_k = 2,2,1,2$ (that is, two paths per OD pair, except for OD pair 3).

$\ell = 1,2,3,4,5$ (that is, five links).

The link-path incidence matrices are:

$$
\begin{array}{c}
 \\ 1 \\ 2 \\ 3 \\ 4 \\ 5
\end{array}
\begin{array}{c}
\begin{array}{ccccccc} A^{(1)} & & A^{(2)} & & A^{(3)} & & A^{(4)} \end{array} \\
\left[\begin{array}{cc:cc:c:cc}
1 & 0 & 1 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 1 & 1 & 0 \\
0 & 1 & 0 & 0 & 0 & 0 & 1
\end{array}\right]
\end{array}
$$

$d_k = 3,6,2,5$

Proportionate flows are given by:

$$((x_{11},x_{12}),(x_{21},x_{22}),1,(x_{41},x_{42})) \geqq 0$$

$$x_{11}+x_{12} = 1,\ x_{21}+x_{22} = 1,\ x_{41} + x_{42} = 1.$$

Induced flow on links is given by:

$$f_1 = 3x_{11} + 6x_{21} + 5x_{41}$$

$$f_2 = 6x_{21}$$

$$f_3 = 3x_{12} + 6x_{22}$$

$$f_4 = 2 + 5x_{41}$$

$$f_5 = 3x_{12} + 5x_{42} .$$

User costs on links are:

$$c_\ell(f_\ell) = \begin{cases} \dfrac{1}{5-f_\ell} & 0 \leqq f_\ell \leqq 4.99 \\[2ex] 100 + 10^4(f_\ell - 4.99) & f_\ell \geqq 4.99 . \end{cases}$$

The following is an example of a path cost computation:

$$c_{12}(X_1,\ldots,X_4) = c_3(f_3) + c_5(f_5)$$

$$= c_3(3x_{12}+6x_{22}) = c_5(3x_{12}+5x_{42}) .$$

The following is an example of a label calculation:

To calculate

$$L((1,0),(0.2,0.8),1,(0.1,0.9))$$

we first calculate flows:

$$f_1 = 3 + 1.2 + 0.5 = 4.7$$

$$f_2 = 1.2$$

$$f_3 = 0 + 4.8 = 4.8$$

$$f_4 = 2 + 0.5 = 2.5$$

$$f_5 = 0 + 4.5 = 4.5 .$$

We then calculate link costs:

$$c_1 = 10/3$$

$$c_2 = 5/19$$

$$c_3 = 5$$

$$c_4 = 2/5$$

$$c_5 = 2$$

These induce the path costs:

$$c_{11} = 3.33*$$

$$c_{12} = 7.00$$

$$c_{21} = 3.59$$

$$c_{22} = 5*$$

$$c_{31} = 0.40*$$

$$c_{41} = 3.73*$$

$$c_{42} = 2$$

The most expensive path with a positive flow is starred for each OD pair. Hence

$$L(X_1, X_2, X_3, X_4) = (1, 2, 1, 1)$$

The induced integer label is

$$IL(X_1, X_2, X_3, X_4) = 1 .$$

PAPER 4

COMPUTATIONAL SAVINGS FROM AGGREGATION

1. Introduction

The object of this paper is to estimate the computational savings that can be achieved by applying an aggregation method to a network model. Since these savings are measured in terms of the number of arithmetical and logical operations to be performed, the question only makes sense when posed for the usual three way combination: network model, aggregation method, and mathematical technique or algorithm. The aggregation method reduces the number of network elements (links, nodes, OD pairs, etc.) and this in turn reduces the number of multiplications, additions, and comparisons to be performed. The remaining sections of this paper are organized according to the classification of network models given in the summary guide, namely, Hitchcock-Koopmans Transportation Problem, Max Flow, Shortest Path, and Equilibrium Models.

Before entering on the specific details of the estimates, several general issues must be discussed. First, as will be seen below, most of the specific results available on operation counts are upper bounds rather than average performance. If these upper bounds are tight or if they are of the right order of magnitude this does not matter. However, a good example is provided by the Simplex Algorithm. The only valid upper bounds on the number of pivot steps grow much more rapidly in the parameters of the problem than does actual computation experience. Therefore, if we are interested in estimating the expected savings from aggregation we should use the latter, not the former.

Secondly, the payment for computation is made for machine time, not for operation counts, although there is obviously a connection between the two. Therefore, ideally we should have estimates of the machine time for a particular programmed version of an algorithm applied to a network model of various sizes. This information is seldom available. However, we shall be able to give examples in the later sections of this paper.

2. Hitchcock-Koopmans Transportation Problem

For this problem, there are m source nodes, n destination nodes, and mn directed links joining. There are two primary algorithms which have been used to find optimal solutions, the special version of the Simplex Method and labelling algorithms (such as that of Ford and Fulkerson).

For the Simplex Method, no upper bound for the number of operations seems to be available. However, there is a considerable body of experience to draw upon. Since there are $m + n - 1$ linearly independent equation constraints, we may expect a (small) multiple of $m + n - 1$ basis changes. Each basis change requires the determination of $m + n - 1$ dual variables, possibly $mn - m - n + 1 = (m - 1)(n - 1)$ comparisons, and the alteration of the basis, which may alter $2 \max(m, n)$ elements. If we assume that m is much smaller than n, an average performance of $0(mn^2)$ is not a bad estimate of the behavior of the algorithm.

The major candidate for aggregation for this problem is (desination) node abstraction (as illustrated in Paper 1). If the problem is compressed from m by n to m by p, the operation count should be reduced by a factor of about p^2/n^2. Thus, in the last example of Paper 1, the operation count should be about $(1 + 9 + 49)/10^4 = 0.0059$ of the number of operations to solve the original problem since three problems of size 3 by 1, 3 by 3, and 3 by 7, were solved. However, this estimate is overly optimistic since it omits the effect of lifting. On the other hand, the estimate of $0(n^2)$, holding m constant, is backed by other evidence for the labelling algorithms.

First, there is the classic result of Munkres (for the assignment problem where $m = n$ and all $a_i = b_j = 1$) that bounds the number of operations by

$$\frac{11n^3 + 12n^2 + 31n}{6}$$

According to Tomizawa, $n^3/3$ additions and $2n^3/3$ comparisons are sufficient to solve assignment problems by an adaptation of the Dijkstra shortest route algorithm.

This concurs with the estimate of Dwyer who, using a technique of reduced matrices on the Transportation Problem, finds a bound for the number of steps to be $m + 2n - 1$. At each step there are mn comparisons and hence we obtain an upper bound of $mn(m + 2n - 1)$. For m much smaller than n, the estimated performance of $0(n^2)$ results as m is held constant. Alternatively for (destination) node abstraction, we expect an improvement of p^2/n^2.

3. Maximum Flow Problem

Let us assume that the connected network N has n nodes, and n_L directed links.

Assume for convenience's sake that n_L is maximal, that is $n_L = n(n - 1)$ although some of the link capacities may be zero. Each link has a nonnegative capacity. The problem consists in finding the maximum flow from one node to another.

The solution algorithm consists of two steps: 1) a labelling process, and 2) a flow change. Our aim is to determine a bound on the (finite) number of flow augmenting paths.

For each flow F, let N(F) denote the network N with a flow F, but with the following capacity changes: if link (i, j) is saturated by F with the quantity x_{ij}, then link (i, j) is given capacity zero, but conversely line (j, i) acquires capacity x_{ij}.

The labelling method generates a sequence of network $N(F_i)$, for $i = 1, \ldots, k$, F_k being the maximal flow. Flow F_{i+1} is obtained from F_i by superimposing a flow augmenting path.

Now it can be shown that a link can be saturated in one network but not in the next one for at most $\frac{n-1}{2}$ times. Since each flow augmenting path saturates at least one link in N(F), and there are n(n-1) links altogether, the total number of low augmenting paths is bounded above by

$$n(n - 1) \left(\frac{n - 1}{2}\right) + 1 \leqq n^3.$$

This bound can be somewhat improved but not reduced to $0(n^2)$.

Since the most natural aggregation method for this problem is not a reduction of the number nodes by extraction or abstraction, but rather a reduction of the number of links by extraction (say, by deleting all links below a given capacity), the above argument (which is adapted from Hu, Edmonds, and Karp) must be modified. This is easily done and yields an upper bound of $0(n_L n)$. Thus, cutting the number of links by half merely reduces the amount of labor by half.

4. Shortest Path Problem

For this problem we shall assume that the network has n nodes and n_L links. Associate to each link (i, j) a distance d_{ij}, which is not necessarily equal to d_{ji}.

The problem consists of finding the shortest:

(a) Single path from one node to one other node.

(b) Paths from one node to all nodes.

(c) Paths between all nodes.

Two main types of algorithms exist:

(A) Tree building algorithms for problems (a) and (b), characterized by the fact that every node has a predecessor (1-5 below).

(B) Matrix algorithm for problem type (c), where network, shortest path distances, the shortest paths themselves, are in matrix form (6-9 below).

The following list gives an account of the best known algorithms.

1. Dijkstra (refined by Whitney and Hillier). This is generally recognized as the most efficient algorithm, for $d_{ij} \geq 0$.

A shortest distance tree is built up from origin to all other nodes by a system of tentative labels which, through a series of iterative steps for each node, eventually assigns a permanent label to each node.

It requires:

$\frac{n(n - 1)}{2}$ additions and comparisons to compute tentative node labels

$\frac{n(n - 1)}{2}$ comparisons to find minimal label at each step

$(n - 1)^2$ comparisons to consult the indices,

or, altogether, less than a total of

$\frac{n^2}{2}$ additions and $2n^2$ comparisons.

(Note this is a "once through" algorithm and solves (a) and (b).)

2. Minty (improved by Whitney and Hillier). Again $d_{ij} \geqq 0$. Here the original algorithm required $n^3/6$ additions and comparisons. After the improvement, the bound is reduced to $n^2/2$.

This appears as a misleading amelioration of the Dijkstra method, since it requires:

$n^2 \log_2 n$ additional comparisons for data ordering, and

$3n^2$ additional comparisons for data modification required for certain link deletions.

Thus it is only efficient for sparse networks with far fewer than n^2 links.

(Note this is a "once through" algorithm and solves (a) and (b).)

3. Ford, Moore, Bellman, d'Esopo. Here d_{ij} are arbitrary. A shortest distance tree is built from origin to all other nodes by a system of labelling, or indexing, or flagging, or shift constructing, where a node becomes temporarily active. Each node is active at least once,

but can be many times. For each active node, there are as many operations a links connected to it. The repeated updating of lables requires at most n^3 additions and comparisons. It is most efficient if there are fewer than 4 arcs per node.

(Note this is an "updating" algorithm and solves (a) and (b) with d_{ij} arbitrary.)

4. Yen. The Yen improvement of the above reduces the operations to $n^3/4$.

(This is an "updating" algorithm and solves (a) and (b) with d_{ij} arbitrary.)

5. Dantzig, Blattner, and Rao. This algorithm seems difficult to program and requires at most $n^3/3$ additions and $2n/3$ comparisons.

(This is an "updating" algorithm and solves (a) and (b) with d_{ij} arbitrary.)

6. Floyd. The original matrix is changed in n steps into final matrix of shortest distances. For $r = 1, \ldots, n$, each of these matrices M_r must be inspected for all i, except $i = r$, connected with all j, except $j = 1$ or r. Thus the total number of additions and comparisons is $n(n-1)(n-2)$.

(Note that this solves (c) and, with modification, works for d_{ij} arbitrary.)

7. Farbey, Land, Murchland (Cascade algorithm). This algorithm requires $n(n-2)$ additions and comparisons for the forward process and $n(n-1)(n-2)$ additions and comparisons for the backward process. Thus, the total is $n^2(n-2)$ operations.

(Note that this solves (c).)

8. Dantzig. This algorithm generates successive matrices M_r of increasing size and requires $n(n-1)(n-2)$ calculations.

(Note that this solves (c).)

9. Floyd, Hu. For a sparsely connected network with n_L about $4n$, say, the following partition improves the Floyd algorithm. Let $N = A \cup B \cup Z$, where A and B are not connected, $|A| = n_A$, $|B| = n_B$, and $|Z| = n_Z$. The total number of additions and comparisons is

$$2(n_A + n_Z)^3 + (n_Z + n_B)^3 + 2n_A n_B n_Z .$$

(Note that this solves (c).)

10. Dantzig. This method also generates n successive $r \times r$ matrices M_r of increasing size for $r = 1, \ldots, n$. These are shortest distance matrices using nodes $\{1, \ldots, r\}$ as intermediate nodes. For each r, three types of evaluation are performed requiring

$$(r-1)^2,\ (r-1)^2, \text{ and } (r-1)(r-2) \text{ operations},$$

respectively. Thus we obtain the bound

$$\sum_{r=1}^{n} \left\{2(r-1)^2 + (r-1)(r-2)\right\} = n(n-1)^2.$$

(Note that this solves (c).)

While we have been unable to discover any direct aggregation method to apply to this problem, the most important conclusion to draw is the fact that when it appears as part of an equilibrium algorithm we must solve (c). Thus the overwhelming evidence is that an operation count of $0(n^3)$ is appropriate here. Estimates of link aggregation would require refinements of the estimates given above.

Another important consideration which has determined the choice of algorithm in many applications is the storage requirements. Briefly, tree algorithms require less than $4n + 2n_L$ storage positions, while matrix algorithms require more than $2n^2$. In a transportation network where n_L is about $4n$, it is clear that algorithms (A) are clearly preferable.

5. Equilibrium Models

We shall restrict our discussion to one aspect of the algorithm proposed in Paper 3. This algorithm has the characteristics of pivoting algorithms where it has been shown that the computation time depends on two factors: (1) the number of labellings and (2) the computation involved in a labelling. Without entering into the detail of the mass of computational experience that is available, it may be summarized as follows, at least for equilibrium models. Both factors appear to be proportional to $(d - 1)^2$, where d is the dimension of the problem, and with possible savings in the functional evaluations. Thus if we extract $n_1, \ldots, n_p$ paths for the p OD pairs, we expect the computation time to grow like

$$(n_1 + \ldots + n_p - p - 1)^4.$$

Thus, a reduction from 3 paths per OD pair to 2 paths per OD pair should cut the computation time by a factor of 16.

PAPER 5

AGGREGATION TEST NETWORKS

1. Introduction

Numerical experimentation with aggregation rules has been hampered by the lack of programs and networks that are suited for such a purpose. This paper reports on a set of programs and models that has been assembled to provide support for research on aggregation. The equilibrium model, TRAFFIC, developed at the University of Montreal provides the foundation for the program battery. This has been converted to run on IBM computers and numerous small improvements have been made. Several cost functions are currently available. These have been supplemented with a simple gravity model for generating demands based on populations.

A forty node problem has been developed to serve as a test network. The model is abstracted from the Massachusetts highway systems. The network design is oriented towards providing a basis on which aggregation experiments can be conducted rather than towards obtaining a factual representation of a particular situation. We have thus taken some liberties in the treatment of nodes, arcs, and their characteristics.

An aggregated network of twenty nodes has been abstracted from the larger network using a specific set of rules for selecting nodes and combining arcs. The following sections describe in detail the programs, the model, and the aggregation.

2. The Network Programs

The TRAFFIC program is a general equilibrium model. It assigns traffic flow to links in a transportation network such that no individual can unilaterally improve his routing. The solution depends on the origin to destination demands between centroids and on links specific cost functions provided by the user.

As described in the program documentation,[1]

> ...the program reads in the characteristics of the links of the network and the demands between origin/destination pairs of nodes (centroids). The algorithm used to perform the assignment is an iterative method developed by S. Nguyen[2] (algorithm A1 in reference). The user supplies parameters which determine the number of iterations which will be performed before the program stops and a report on the resulting solution is put out. The program also allows the user to restart computation from an existing solution for the same network, and carry out further iterations as desired.

The inputs to the system include program parameters, network descriptions, and the demand matrix. The program parameters include two report title cards, a comment card; a parameter card containing tolerances, maximum cycles, print control, restart, and number of delay functions, delay (or cost) function coefficient cards, and a network size card containing the numbers of nodes, links, and centroids (nodes generating and/or absorbing flow). The network description is link oriented. Each card in this group provides the start and end nodes of a link, the length, the number of lanes, the number of the applicable delay function plus two link dependent coefficients as required, and observed characteristics

---

[1]Nguyen, Sang, and Linda James. 1975 (March). TRAFFIC: An Equilibrium Tráffic Assignment Program, Publication #17. Centre de Recherche sur les Transports; Univerite' de Montréal: Montreal, Canada. Much of the discussion in this section is based on this report.

[2]Nguyen, Sang. 1973 (April). A Mathematical Programming Approach to Equilibrium Methods of Traffic Assignment with Fixed Demands, Publication #138. Départment d'Informatique, Universite' de Montréal: Montreal, Canada.

including speed, time, and volume. The demand matrix consists of a set of cards for each origin containing the nonzero demands to other centroids with the centroid numbers. A restart capability using a file of intermediate results is available but has not been used in our experiments to date.

The outputs from the system include a log of each cycle, a terminal solution report and a file containing the final solution in a form suitable for restarting. Each log line reports the current value of the objective function and the rate of change. The terminal solution report contains the reason for terminations, execution times (currently inoperative), the total vehicle hours, miles, and average speed, and a line for each link which reports the input parameters plus the observed time and volume. The only difference in the output reports is the addition of the delay function coefficients on the link section. The reader is referred to Nguyen and James for a further discussion of the program. The outputs will be illustrated below in presenting the network model.

Cost functions are provided to TRAFFIC by the user. Four such functions have been programmed. These include the following cost (time) functions per traveler on a link and the associated objective functions.

Exponential:

User cost

$$= \quad (\alpha e^{\beta \cdot \text{Flow}}) \cdot \text{Distance}$$

where α and β are two parameters. α is the reciprocal of the speed under traffic free conditions. Several pairs of α and β can be provided to the program (on the delay function coefficient cards); a selection of one pair is provided for each link on the network description (through the delay function number parameter). The flow and distance depend on the individual link.

Objective function cost

$$= \frac{\alpha}{\beta} (e^{\beta \cdot \text{Flow}} - 1) \cdot \text{Distance}$$

In the objective function this is summed over all links.

Linear:

User cost

$$= (\alpha + \beta \cdot \text{Flow}) \cdot \text{Distance}$$

where α and β are two parameters. α is the reciprocal of the speed under traffic free conditions.

Objective function cost

$$= (\alpha \cdot \text{Flow} + \frac{1}{2} \beta \cdot \text{Flow}^2) \cdot \text{Distance}$$

Fourth Power Rule:

User cost

$$= T0 \, (1 + .15 \, (\frac{\text{Flow}}{C})^4)$$

where T_o is the time to transit the link when there is no traffic and C is the effective capacity as defined by transportation planners who use this function.

Objective function cost

$$= \quad T_0 \cdot \text{Flow} \cdot (1 + .03 (\frac{\text{Flow}}{C})4)$$

Inverse (Modified from Potts and Oliver)

User cost

$$= \begin{cases} \dfrac{1}{5 - \text{Flow}} & \text{if Flow is less than } 4.99 \\[2ex] \dfrac{1}{5 - 4.99} + N(\text{Flow} - 4.99) & \text{if Flow is greater than } 4.99 \end{cases}$$

The original function was capacitated. It was modified here to provide linear growth at a very high rate (N) and effectively capacitate the problem. In the final solution, no flow should exceed 4.99.

Objective function cost

$$\ln 5 - \ln (5 - \text{Flow}) \quad \text{if Flow} \leq 4.99$$

$$\ln 5 - \ln (.01) + (100 + \tfrac{1}{2} N (\text{Flow} - 4.99)) \; (\text{Flow} - 4.99) \quad \text{if Flow} \geq 4.99$$

A gravity demand model has been developed to generate demand data for the TRAFFIC model. It uses the basic formula

$$\text{DEMAND (I to J)} = \frac{\text{SCALE} \cdot \text{POPULATION (I)} \cdot \text{POPULATION(J)}}{\text{DISTANCE (I to J)}^2}$$

The scale factor ('SCALE') is provided for the user to fit the resulting demands to any observed demands.

The program accepts as input the number of centroids, a cutoff below which demand is treated as zero, the scale factor, the populations and the distance matrix (in upper triangular form). If demands are not to be included for a pair of nodes, this can be indicated by setting the distance to zero.

An additional capability is provided for automatic centroid aggregation. The user provides as input the original nodes to be aggregated into each new node. The demand between two aggregated nodes is the sum of all the demands between the two sets of original nodes which are combined into the aggregated nodes. Intra-set demands are disregarded.

The printed output of the gravity model includes a listing of the input and the calculated demands (after any aggregation). A file is created for input to the traffic program containing the demands. The job control language provides for this file to be automatically concatenated to the other TRAFFIC input data.

3. The Models

A small experimentation model has been developed for testing alternative aggregation rules and parameters. This model is abstracted from the Massachusetts highway system and is shown in Figure 1. It includes 28 intra-state nodes and 13 extra-state nodes which feed demands into the system as identified in Figure 2. All nodes are centroids and the demands have been developed from 1970 population data. Forty-seven bidirectional links within the state are used representing major turnpike, four lane and two

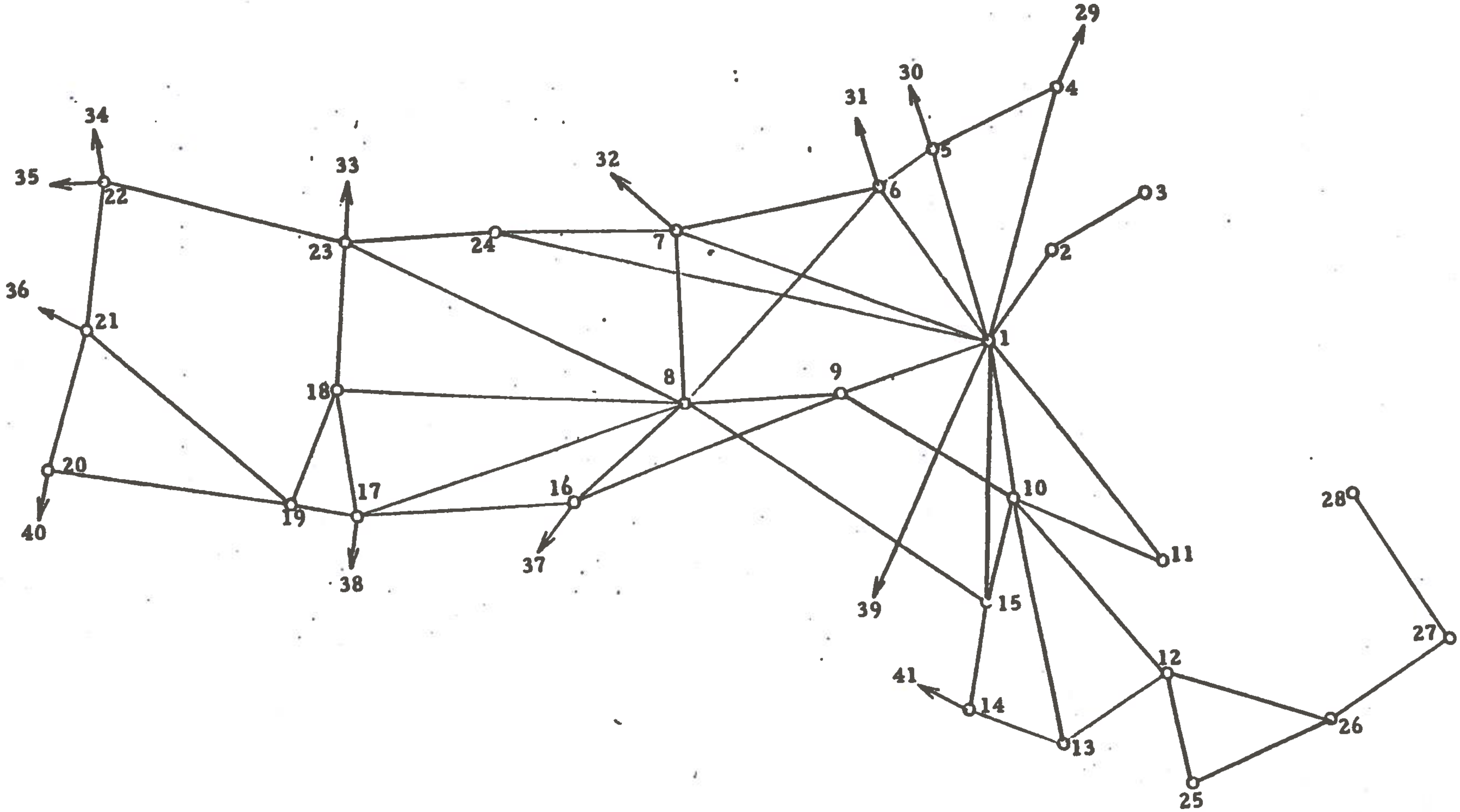


FIGURE 1. TEST NETWORK (BEFORE AGGREGATION)

1.	Boston	22.	Williamstown
2.	Salem	23.	Greenfield
3.	Gloucester	24.	New Salem
4.	Newburyport	25.	Falmouth
5.	Lawrence	26.	Hyannis-
6.	Lowell	27.	Orleans
7.	Fitchburg	28.	Provincetown
8.	Worcester	29.	Portland (Me)
9.	Framingham	30.	Concord (NH)
10.	Brockton	31.	Manchester (NH)
11.	Plymouth	32.	Keene (NH)
12.	Wareham	33.	Brattleboro (VT)
13.	New Bedford	34.	Burlington (VT)
14.	Fall River	35.	Troy (NY)
15.	Taunton	36.	Albany (NY)
16.	Sturbridge	37.	New York (NY)
17.	Springfield	38.	Hartford (CN)
18.	Northampton	39.	Providence (RI) via Boston
19.	Westfield	40.	New York (NY) via Gt. Barrington
20.	Gt. Barrington	41.	Providence (RI) via Fall River
21.	Pittsfield		

FIGURE 2. TEST NETWORK NODE IDENTIFICATIONS

lane roads. Some liberties were taken to develop a useful test network in this process. A linear cost (time) model is used with the parameters set so that average speeds under uncongested conditions are 60 mph, 50 mph, and 40 mph. The increase in time with congestion is scaled so that most roads show a moderate decrease in average speed. Distances are taken from a state highway map. The solution showing the network loading by links is displayed in Figure 3 .

4. Aggregation

A nodel aggregation rule has been developed and applied to the Massachusetts test model. The following algorithm is used for node (and centroid) aggregation:

Step 1. Define neighborhood of node $i = n_i = \{j \mid \|(i,j)\| \leq 24\}$

Let $S = \{i \mid |n_i| \geq 2\}$.

Step 2. Order S according to population size.

Step 3. Aggregate $\{\{i\} \cup n_i \mid i \in S\}$ in above order with stipulations:

a) If $K \cap n_{i_\mu}$, $i_1 < i_2 < \ldots$, include K in first n_i with $|n_i| = 2$, if one exists. Otherwise, let $K \in n_{i_1}$.

b) If because of a), $n_i \rightarrow n_i'$ with $|n_i'| < 2$, exclude i from S.

Step 4. Aggregate remaining single end nodes K to adjacent set $\{i\} \cup n_i$ if some link joining K to i or n_i has length ≤ 26.

Step 5. Consider remaining single nodes as aggregated nodes.

These aggregation rules are heuristic; the parameters have been set so as to produce a network of reasonable size. However, they illustrate an informed attempt to formulate a precise rule.

MASSACHUSETTS STATEWIDE TRAFFIC MODEL

NC AGGREGATION

LINK IDEN	NODE FROM	NODE TO	LENGTH MILES	NUMB LANES	DELAY FUNCT	OBSERVED TRAVEL SPEED (MPH)	OBSERVED TRAVEL TIME (MINS)	OBSERVED VOLUME	LINK TYPE	PREDICTED TRAVEL TIME (MINS)	PREDICTED VOLUME	PARAMETERS A	B
1	1	2	14.0C	0.1	2	0.0	0.0	NA	0	28.77	475	0	0
2	1	4	35.CC	0.1	2	C.0	0.0	NA	0	44.14	34	0	0
3	1	5	26.00	C.1	1	C.C	0.C	NA	0	34.59	220	0	C
4	1	6	25.00	0.1	1	0.0	0.0	NA	0	38.49	360	0	0
5	1	7	47.CC	C.1	1	0.0	0.0	NA	0	66.81	281	0	0
6	1	9	20.0C	C.1	1	C.0	0.0	NA	0	5C.01	1000	0	0
7	1	10	23.C0	0.1	2	0.0	0.0	NA	0	44.60	411	0	0
8	1	11	35.00	0.1	1	C.0	C.C	NA	0	36.00	19	0	0
9	1	15	35.00	0.1	1	C.0	0.0	NA	0	55.67	394	0	0
1C	1	24	68.CC	0.1	3	0.C	C.C	NA	0	102.34	2	0	0
11	1	39	44.00	0.1	0	0.0	0.0	NA	0	C.0	297	0	0
12	2	1	14.CC	1.0	2	0.0	C.C	NA	0	28.77	475	0	0
13	2	3	16.00	0.1	1	C.0	C.0	NA	0	17.63	68	0	0
14	3	2	16.00	1.C	1	0.0	0.0	NA	0	17.63	68	0	0
15	4	1	35.00	1.0	2	C.0	0.0	NA	0	44.14	34	0	0
16	4	5	20.00	0.1	1	C.0	0.C	NA	0	20.15	5	0	0
17	4	29	7C.00	0.1	0	0.0	0.0	NA	0	C.0	10	0	0
18	5	1	26.CC	1.0	1	C.0	0.C	NA	0	34.59	220	0	0
19	5	4	20.C0	1.0	1	0.C	C.0	NA	0	20.15	5	0	0
2C	5	6	11.CC	0.1	1	C.0	0.0	NA	0	12.62	98	0	0
21	5	30	41.C0	C.1	0	C.0	0.0	NA	0	0.0	14	0	0
22	6	1	25.00	1.C	1	C.0	0.0	NA	0	38.49	360	0	0
23	6	5	11.CC	1.0	1	C.0	0.0	NA	0	12.62	98	0	0
24	6	7	27.C0	0.1	1	C.0	C.0	NA	0	29.C6	51	0	0
25	6	8	41.00	C.1	1	C.C	C.C	NA	0	44.57	58	0	0
26	6	31	31.0C	0.1	0	0.0	C.0	NA	0	0.0	67	0	0
27	7	1	47.CC	1.0	1	C.0	0.0	NA	0	66.81	281	0	0
28	7	6	27.CC	1.0	1	C.0	C.0	NA	0	29.06	51	0	0
29	7	8	25.CC	0.1	3	G.0	0.G	NA	0	38.88	23	0	0
3C	7	24	24.00	C.1	1	C.0	0.0	NA	0	34.54	293	0	0
31	7	32	39.C0	C.1	0	C.0	0.0	NA	0	0.0	5	0	0
32	8	6	41.00	1.0	1	0.0	C.C	NA	0	44.57	58	0	0
33	8	7	25.C0	1.0	3	0.0	0.0	NA	0	38.88	23	0	0
34	8	9	20.00	C.1	1	C.0	0.C	NA	0	30.50	350	0	0
35	8	15	55.C0	0.1	3	C.C	C.0	NA	0	89.41	52	0	0
36	8	16	20.CC	0.1	3	0.0	C.0	NA	0	36.C1	125	0	0
37	8	17	52.0C	C.1	3	C.0	C.C	NA	0	78.64	5	0	0
38	8	18	54.0C	C.1	3	C.0	0.0	NA	0	101.25	156	0	0
39	8	23	62.CC	0.1	3	C.0	0.0	NA	0	93.28	2	0	0
4C	9	1	20.0C	1.0	1	0.0	C.C	NA	0	5C.01	1000	0	0
41	9	8	20.00	1.G	1	C.C	0.0	NA	0	30.50	350	0	0
42	9	10	32.CC	0.1	3	C.0	C.0	NA	0	51.87	50	0	0
43	9	16	38.00	0.1	1	0.0	0.0	NA	0	63.57	449	0	0
44	10	1	23.00	1.C	2	C.C	G.C	NA	0	44.60	411	0	0
45	10	9	32.00	1.0	3	C.0	C.0	NA	0	51.87	50	0	0

FIGURE 3. TRAFFIC LOADINGS ON TEST NETWORK

MASSACHUSETTS STATEWIDE TRAFFIC MODEL

NO AGGREGATION

LINK IDEN	NODE FROM	NODE TO	LENGTH MILES	NUMB LANES	DELAY FUNCT	OBSERVED TRAVEL SPEED (MPH)	OBSERVED TRAVEL TIME (MINS)	OBSERVED VOLUME	LINK TYPE	PREDICTED TRAVEL TIME (MINS)	PREDICTED VOLUME	PARAMETERS A	PARAMETERS B
46	10	11	25.00	0.1	3	0.0	0.0	NA	0	37.56	1	0	0
47	10	12	31.00	0.1	2	0.0	0.0	NA	0	37.48	5	0	0
48	10	13	33.00	0.1	1	0.0	0.0	NA	0	38.84	118	0	0
49	10	15	15.00	0.1	2	0.0	0.0	NA	0	18.86	32	0	0
50	11	1	35.00	1.0	1	0.0	0.0	NA	0	36.00	19	0	0
51	11	10	25.00	1.0	3	0.0	0.0	NA	0	37.56	1	0	0
52	12	10	31.00	1.0	2	0.0	0.0	NA	0	37.48	5	0	0
53	12	13	17.00	0.1	2	0.0	0.0	NA	0	20.40	0	0	0
54	12	25	22.00	0.1	2	0.0	0.0	NA	0	26.44	1	0	0
55	12	26	26.00	0.1	2	0.0	0.0	NA	0	31.34	3	0	0
56	13	10	33.00	1.0	1	0.0	0.0	NA	0	38.84	118	0	0
57	13	12	17.00	1.0	2	0.0	0.0	NA	0	20.40	0	0	0
58	13	14	11.00	0.1	1	0.0	0.0	NA	0	12.80	109	0	0
59	14	13	11.00	1.0	1	0.0	0.0	NA	0	12.80	109	0	0
60	14	15	17.00	0.1	1	0.0	0.0	NA	0	27.23	401	0	0
61	14	41	17.00	0.1	0	0.0	0.0	NA	0	0.0	301	0	0
62	15	1	35.00	1.0	1	0.0	0.0	NA	0	55.67	394	0	0
63	15	8	55.00	1.0	3	0.0	0.0	NA	0	89.41	52	0	0
64	15	10	15.00	1.0	2	0.0	0.0	NA	0	18.86	32	0	0
65	15	14	17.00	1.0	1	0.0	0.0	NA	0	27.23	401	0	0
66	16	8	20.00	1.0	3	0.0	0.0	NA	0	36.01	125	0	0
67	16	9	38.00	1.0	1	0.0	0.0	NA	0	63.57	449	0	0
68	16	17	32.00	0.1	1	0.0	0.0	NA	0	41.73	203	0	0
69	16	37	56.00	0.1	0	0.0	0.0	NA	0	0.0	517	0	0
70	17	8	52.00	1.0	3	0.0	0.0	NA	0	78.64	5	0	0
71	17	16	32.00	1.0	1	0.0	0.0	NA	0	41.73	203	0	0
72	17	18	19.00	0.1	1	0.0	0.0	NA	0	24.95	209	0	0
73	17	38	26.00	0.1	0	0.0	0.0	NA	0	0.0	70	0	0
74	18	8	54.00	1.0	3	0.0	0.0	NA	0	101.25	156	0	0
75	18	17	19.00	1.0	1	0.0	0.0	NA	0	24.95	209	0	0
76	18	19	17.00	0.1	3	0.0	0.0	NA	0	46.09	505	0	0
77	18	23	23.00	0.1	1	0.0	0.0	NA	0	28.88	170	0	0
78	19	18	17.00	1.0	3	0.0	0.0	NA	0	46.09	505	0	0
79	19	20	37.00	0.1	1	0.0	0.0	NA	0	59.63	408	0	0
80	19	21	47.00	0.1	3	0.0	0.0	NA	0	73.08	23	0	0
81	20	19	37.00	1.0	1	0.0	0.0	NA	0	59.63	408	0	0
82	20	21	20.00	0.1	2	0.0	0.0	NA	0	28.64	129	0	0
83	20	40	30.00	0.1	0	0.0	0.0	NA	0	0.0	535	0	0
84	21	19	47.00	1.0	3	0.0	0.0	NA	0	73.08	23	0	0
85	21	20	20.00	1.0	2	0.0	0.0	NA	0	28.64	129	0	0
86	21	22	21.00	0.1	2	0.0	0.0	NA	0	29.90	124	0	0
87	21	36	37.00	0.1	0	0.0	0.0	NA	0	0.0	14	0	0
88	22	21	21.00	1.0	2	0.0	0.0	NA	0	29.90	124	0	0
89	22	23	42.00	0.1	3	0.0	0.0	NA	0	75.93	128	0	0
90	22	34	35.00	0.1	0	0.0	0.0	NA	0	0.0	1	0	0

FIGURE 3. TRAFFIC LOADINGS ON TEST NETWORK (CONTINUED)

MASSACHUSETTS STATEWIDE TRAFFIC MODEL

NO AGGREGATION

LINK IDEN	NODE FROM	NODE TO	LENGTH MILES	NUMB LANES	DELAY FUNCT	OBSERVED TRAVEL SPEED (MPH)	OBSERVED TRAVEL TIME (MINS)	OBSERVED VOLUME	LINK TYPE	PREDICTED TRAVEL TIME (MINS)	PREDICTED VOLUME	PARAMETERS A	PARAMETERS B
91	22	35	33.00	0.1	0	0.0	0.0	NA	0	0.0	5	0	0
92	23	8	62.00	1.0	3	0.0	0.0	NA	0	93.28	2	0	0
93	23	18	23.00	1.0	1	0.0	0.0	NA	0	28.88	170	0	0
94	23	22	42.00	1.0	3	0.0	0.0	NA	0	75.93	128	0	0
95	23	24	23.00	0.1	3	0.0	0.0	NA	0	50.56	291	0	0
96	23	33	20.00	0.1	0	0.0	0.0	NA	0	0.0	1	0	0
97	24	1	68.00	1.0	3	0.0	0.0	NA	0	102.34	2	0	0
98	24	7	24.00	1.0	1	0.0	0.0	NA	0	34.54	293	0	0
99	24	23	23.00	1.0	3	0.0	0.0	NA	0	50.56	291	0	0
100	25	12	22.00	1.0	2	0.0	0.0	NA	0	26.44	1	0	0
101	25	26	21.00	0.1	3	0.0	0.0	NA	0	31.50	0	0	0
102	26	12	26.00	1.0	2	0.0	0.0	NA	0	31.34	3	0	0
103	26	25	21.00	1.0	3	0.0	0.0	NA	0	31.50	0	0	0
104	26	27	19.00	0.1	2	0.0	0.0	NA	0	22.80	0	0	0
105	27	26	19.00	1.0	2	0.0	0.0	NA	0	22.80	0	0	0
106	27	28	26.00	0.1	2	0.0	0.0	NA	0	31.20	0	0	0
107	28	27	26.00	1.0	2	0.0	0.0	NA	0	31.20	0	0	0
108	29	4	70.00	1.0	0	0.0	0.0	NA	0	0.0	10	0	0
109	30	5	41.00	1.0	0	0.0	0.0	NA	0	0.0	14	0	0
110	31	6	31.00	1.0	0	0.0	0.0	NA	0	0.0	67	0	0
111	32	7	39.00	1.0	0	0.0	0.0	NA	0	0.0	5	0	0
112	33	23	20.00	1.0	0	0.0	0.0	NA	0	0.0	1	0	0
113	34	22	35.00	1.0	0	0.0	0.0	NA	0	0.0	1	0	0
114	35	22	33.00	1.0	0	0.0	0.0	NA	0	0.0	5	0	0
115	36	21	37.00	1.0	0	0.0	0.0	NA	0	0.0	14	0	0
116	37	16	56.00	1.0	0	0.0	0.0	NA	0	0.0	517	0	0
117	38	17	26.00	1.0	0	0.0	0.0	NA	0	0.0	70	0	0
118	39	1	44.00	1.0	0	0.0	0.0	NA	0	0.0	297	0	0
119	40	20	30.00	1.0	0	0.0	0.0	NA	0	0.0	535	0	0
120	41	14	17.00	1.0	0	0.0	0.0	NA	0	0.0	301	0	0

FIGURE 3. TRAFFIC LOADINGS ON TEST NETWORK (CONCLUDED)

	New	Old
Results:	1	1, 2, 3, 10, 11
	2	8, 9, 16
	3	17, 18, 19
	4	12, 13, 14
	5	4, 5, 6
	6	20, 21, 22
	7	7, 23, 24
	8	25, 26, 27, 28
	9	15

Link aggregation is accomplished by combining into one aggregated link all links from nodes in one aggregated node to nodes in another aggregated node. The distance is averaged based on guestimated usage. The constant part of the cost function is similarly averaged. The slope part is averaged using the usage squared.

The network which results from the application of these rules is shown in Figure 4. An illustration from a part of the network is provided in Figure 5 which shows how groups of nodes are combined into single nodes and how clusters of links are combined into single links.

The solution showing the network loading by links of the aggregated network is displayed in Figure 6. The difference in results in the South-Eastern area (shown above in Figure 5) has been analyzed to determine some of the effects of the aggregation. The major difference is that 267 units of flow were transferred from the path 10-15-14 to the paths 10-12 and 10-13 (where the numbers refer to disaggregated nodes). There is also an increase in apparent times of zero to ten minutes on various paths.

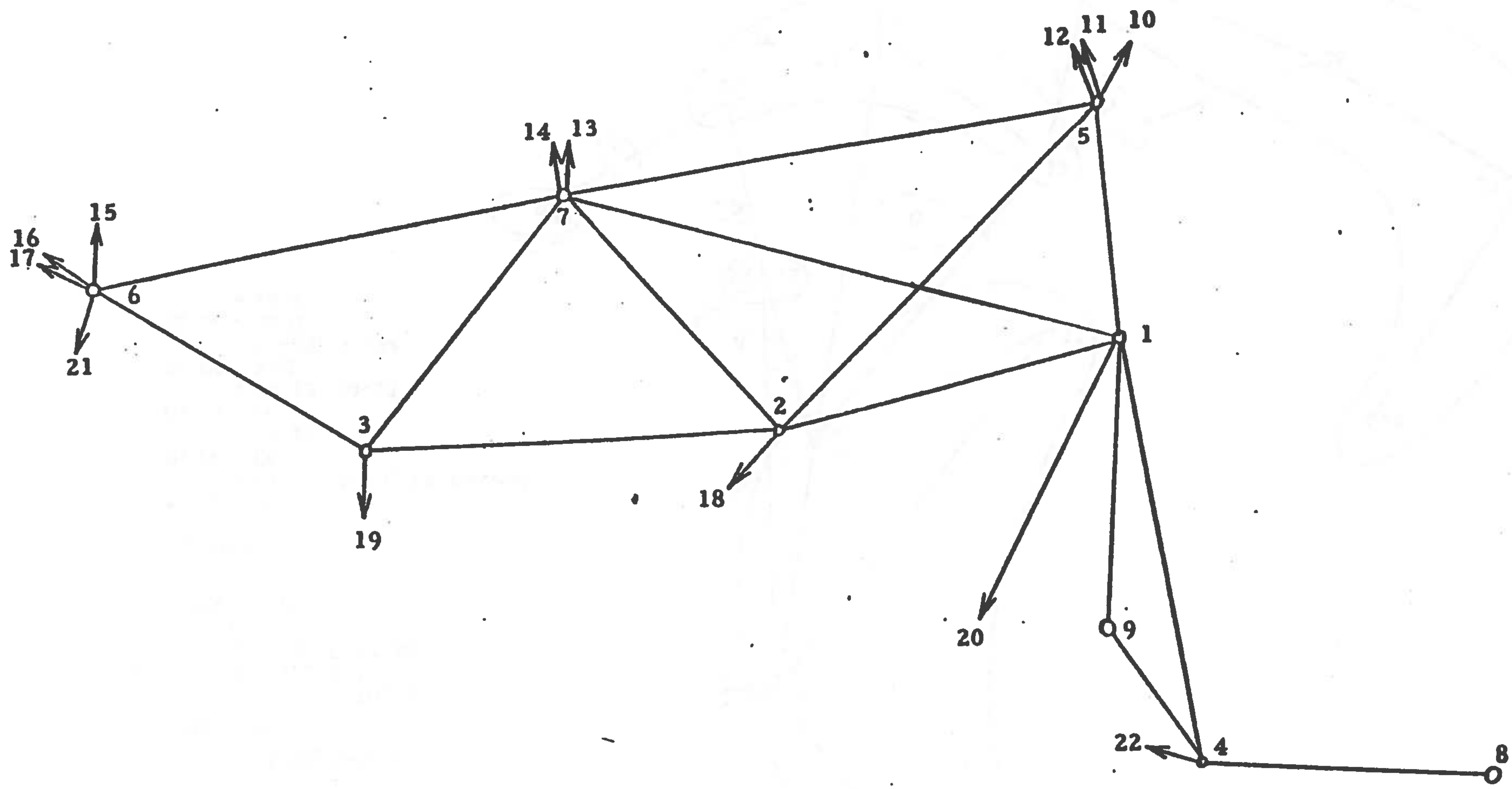


FIGURE 4. AGGREGATED TEST NETWORK

AGGREGATION

NODES:

I = 1, 2, 3, 10, 11
IV = 12, 13, 14.
VIII = 25, 26, 27, 28
XI = 15
XXII = 41

LINKS:

A: I - IX
= 15-1, 15-10, 15-Beyond
B: IV - IX
= 14-15
C: I - IV
= 10-12, 10-13
D: IV - VIII
= 12-25, 12-26
E: IV - XXII
= 14-41

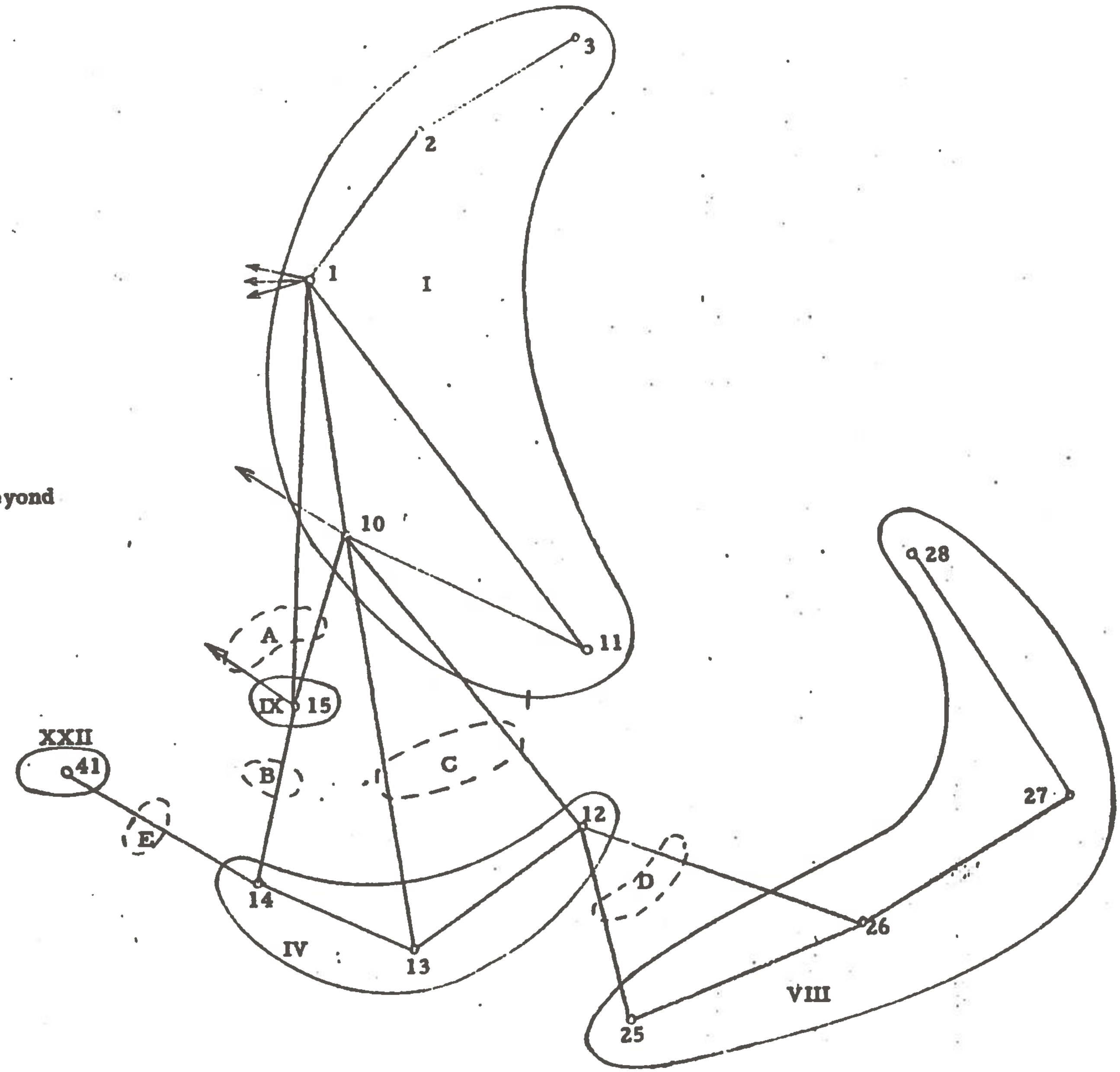


FIGURE 5. DETAILED AGGREGATION DIAGRAM (SOUTHEASTERN SECTION)

MASSACHUSETTS STATEWIDE TRAFFIC MODEL
AGGREGATION NUMBER 1

LINK IDEN	NODE FROM	NODE TO	LENGTH MILES	NUMB LANES	DELAY FUNCT	OBSERVED TRAVEL SPEED (MPH)	OBSERVED TRAVEL TIME (MINS)	OBSERVED VOLUME	LINK TYPE	PREDICTED TRAVEL TIME (MINS)	PREDICTED VOLUME	PARAMETERS A	B
1	1	2	13.CO	0.1	2	C.0	0.0	NA	0	46.01	1300	0	0
2	1	4	56.CC	C.1	1	0.0	0.C	NA	0	89.25	396	0	0
3	1	5	26.GC	C.1	1	C.0	C.0	NA	0	49.25	596	0	0
4	1	7	68.CC	0.1	2	0.0	0.0	NA	0	98.08	135	0	0
5	1	9	36.CC	C.1	2	C.0	0.C	NA	0	56.82	210	0	0
6	1	20	17.CC	0.1	C	C.C	C.0	NA	0	0.0	297	0	0
7	2	1	13.CC	1.0	2	C.0	0.0	NA	0	46.01	1300	0	0
8	2	3	56.00	C.1	1	C.0	0.0	NA	0	9C.11	406	0	0
9	2	5	43.C0	0.1	2	C.0	0.0	NA	0	54.94	43	0	0
1C	2	7	28.CC	0.1	3	C.0	0.0	NA	0	52.80	161	G	0
11	2	18	56.CG	0.1	0	C.0	0.0	NA	0	G.0	524	0	0
12	3	2	56.00	1.0	1	C.0	0.0	NA	0	90.11	406	0	0
13	3	6	53.CC	0.1	2	0.0	C.C	NA	0	94.25	321	0	0
14	3	7	40.C0	0.1	2	0.0	G.C	NA	0	49.09	15	0	0
15	3	19	26.CC	C.1	0	C.0	0.0	NA	0	0.0	71	0	0
16	4	1	56.GO	1.0	1	C.0	0.0	NA	0	89.25	396	0	0
17	4	8	33.00	0.1	2	0.0	0.C	NA	0	39.84	4	0	0
18	4	9	23.00	C.1	2	G.0	0.C	NA	0	33.15	134	0	0
19	4	22	44.0C	C.1	0	C.0	C.C	NA	0	0.0	301	0	0
2C	5	1	26.00	1.0	1	0.0	0.0	NA	0	49.25	596	0	0
21	5	2	43.C0	1.0	2	C.0	C.C	NA	0	54.94	43	0	0
22	5	7	56.CC	0.1	2	C.0	0.0	NA	0	70.58	34	0	0
23	5	10	70.CG	0.1	0	0.0	C.0	NA	0	0.0	10	0	0
24	5	11	41.CC	0.1	0	G.0	0.0	NA	0	C.0	14	0	0
25	5	12	31.0C	C.1	C	C.0	0.0	NA	0	0.0	68	0	0
26	6	3	53.C0	1.0	2	C.0	0.0	NA	0	94.25	321	0	0
27	6	7	64.CC	0.1	3	C.0	0.0	NA	0	133.89	247	0	0
28	6	15	35.CO	0.1	0	C.C	0.0	NA	0	0.0	1	0	0
29	6	16	33.00	0.1	0	C.0	0.C	NA	0	0.0	5	0	0
30	6	17	37.CC	0.1	G	C.0	0.0	NA	0	0.0	14	0	0
31	6	21	3C.CC	C.1	0	C.0	0.0	NA	0	0.0	544	0	0
32	7	1	68.00	1.0	2	0.0	C.0	NA	0	98.08	135	0	0
33	7	2	28.CC	1.C	3	0.0	0.0	NA	0	52.80	161	0	0
34	7	3	40.CG	1.C	2	C.0	0.C	NA	0	49.09	15	0	0
35	7	5	56.CC	1.0	2	C.0	C.C	NA	0	70.58	34	0	0
36	7	6	64.0C	1.0	3	C.0	0.C	NA	0	133.89	247	0	0
37	7	13	39.CC	C.1	0	C.0	0.C	NA	0	0.0	5	0	0
38	7	14	20.CC	0.1	0	C.0	C.0	NA	0	0.0	1	0	0
39	8	4	33.CC	1.0	2	0.0	0.C	NA	0	39.84	4	0	0
4C	9	1	36.CC	1.C	2	C.0	0.C	NA	0	56.82	210	0	0
41	9	4	23.0C	1.0	2	G.0	C.0	NA	0	33.15	134	0	0
42	10	5	7C.CC	1.0	0	C.0	0.0	NA	0	0.0	10	0	0
43	11	5	41.CO	1.0	G	C.0	0.C	NA	0	C.0	14	0	0
44	12	5	31.C0	1.G	0	C.C	0.0	NA	0	0.0	68	0	0
45	13	7	39.00	1.0	0	C.G	0.C	NA	0	G.0	5	0	0

FIGURE 6. TRAFFIC LOADINGS ON AGGREGATED NETWORK

MASSACHUSETTS STATEWIDE TRAFFIC MCDEL
AGGREGATICN NUMBER 1

LINK ICEN	NCCE FRCM	NCDE TC	LENGTH MILES	NUMB LANES	DELAY FUNCT	OBSERVED TRAVEL SPEED (MPH)	CBSERVED TRAVEL TIME (MINS)	CBSERVED VCLUME	LINK TYPE	PREDICTED TRAVEL TIME (MINS)	PREDICTED VCLUME	PARAMETERS A	B
46	14	7	2C.CC	1.C	0	C.0	C.0	NA	0	0.0	1	0	0
47	15	6	35.00	1.0	C	C.0	0.0	NA	0	C.0	1	0	0
48	16	6	33.CC	1.0	0	C.0	0.0	NA	0	C.0	5	0	0
49	17	6	37.0C	1.0	0	C.0	C.C	NA	0	C.0	14	0	0
50	18	2	56.00	1.C	0	C.0	0.0	NA	0	0.0	524	0	0
51	19	3	26.00	1.0	C	0.C	0.C	NA	0	0.0	71	0	0
52	20	1	17.CC	1.0	C	0.0	C.0	NA	0	0.0	297	0	0
53	21	6	30.CC	1.0	0	C.0	0.C	NA	0	0.0	544	0	0
54	22	4	44.00	1.0	0	C.0	0.0	NA	0	0.0	301	0	0

FIGURE 6. TRAFFIC LOADINGS ON AGGREGATED NETWORK (CONTINUED)

Note that a bias has been introduced through the parameters of the aggregation rule. A table of differences in results is shown in Figure 7 .

Identifier	Origin	Destination	Volume	Minutes	Differences (Aggregate over Disaggregate)	
					Volume	Minutes
A	1	15	394	55.67		
	10	15	32	18.86		
	"Beyond"	15	52	89.41		
	Total		478			
	Average			56.88		
	Aggregate		211	56.82	-267	-0.06
B	14	15	401	27.23		
	Aggregate		134	33.15	-267	+5.92
C	10	13	118	38.84		
	12	13	5	37.48		
	Total		123			
	Average			38.78		
	Aggregate		390	89.25	+267	+50.47* (+5.87)
D	25	12	1	26.44		
	26	12	3	31.34		
	Total		4			
	Average			30.12		
	Aggregate		4	39.84	0	+9.72
E	14	22	301	0		
	Aggregate		301	0	0	0

*Due to intranodal travel - most traffic continues on to node 1 for an additional period of 44.60 minutes, a total of 83.38 minutes and a difference of -5.87

FIGURE 7. AGGREGATE VS. DISAGGREGATE RESULTS (SOUTHEASTERN SECTION)

PAPER 6

AGGREGATION BY THE EXTRACTION OF A TRANSVERSAL LINK

1. Introduction

The aim of this paper is to study the effect on the network descriptive assignment problem of extracting (or equivalently inserting) a transversal link between two chains joining one fixed OD pair.

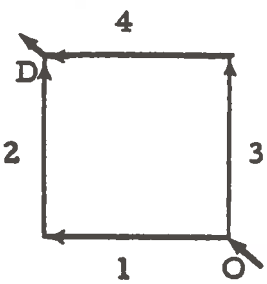
The main result is the precise evaluation of the cost change per individual traveler brought about by the simplest non-trivial aggregation procedure: the extraction of a single link. This cost can be given explicitly in terms of the coefficients of the (linear) link costs and the demand. (If this result is interpreted in terms of link insertion instead of extraction, this result characterizes completely the so-called "Braess Paradox", where the addition of a link increases the travel cost of every individual.)

The link costs (disutility functions) are initially first degree polynomials (with non-negative coefficients) of the flows. In the last sections, these costs will be generalized to polynomials of arbitrary degree. (Nowhere will integer solution requirements be considered).

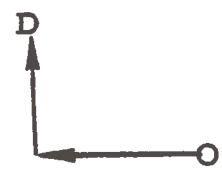
2. Problem P_I. Linear costs without transversal link.

Consider the network N below, whose directed links are numbered from 1 to 4.

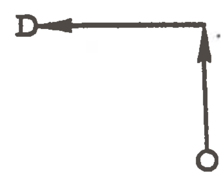
N



We assume a flow of K units from O to D, which can be distributed along the two paths:



denoted by L



denoted by R.

To each link, we associate a cost function/individual

$$\left.\begin{array}{l} f_j(x) = A_j x + B_j \\ A_j,\ B_j \in \mathbb{R}^+,\ \text{unless zero} \\ A_1+A_2 > 0, A_3+A_4 > 0 \end{array}\right\} \qquad (j = 1,\ldots,4), \tag{2.1}$$

for travel along link j with flow of x units. Let P_I denote the assignment problem characterized by $\{N, K, f_j\}$.

<u>Descriptive solution of P_I</u>. Let x_0, $K-x_0$ be optimal descriptive flows on L, R respectively. Then our requirement that both L <u>and</u> R be utilized,

namely, $0 < x_0 < K$, also takes the form

(2.2) $$\mu = \mathrm{Max}\left(\frac{\beta}{A_3+A_4}, \frac{-\beta}{A_1+A_2}\right) < K,$$

where

2.3) $$\beta = B_1 + B_2 - B_3 - B_4 .$$

Let $$\Sigma = A_1 + A_2 + A_3 + A_4 > 0 .$$

It is easy to show that if (2.2) holds,

2.4) $$x_0 = \frac{K(A_3+A_4) - \beta}{\Sigma}$$

and the minimal cost/individual on L or R is

(2.5) $$C_I = \frac{(A_1+A_2)(A_3+A_4)K + (A_1+A_2)(B_3+B_4) + (A_3+A_4)(B_1+B_2)}{\Sigma} .$$

<u>Normative solution of P_1</u>. Here in order to utilize L and R, we must have an OD flow K with

(2.6) $$\tfrac{1}{2}\mu < K,$$

in which case, the optimal normative flow along L is

2.7) $$x_0 = x_0 + \frac{\beta}{2\Sigma} ,$$

and the normative costs/individual on L and R are given by

2.8) $$C_I + \frac{(A_1+A_2)\beta}{2\Sigma} \quad \text{and} \quad C_1 - \frac{(A_3+A_4)\beta}{2\Sigma}$$

respectively; and the normative cost for the total OD flow of K units is

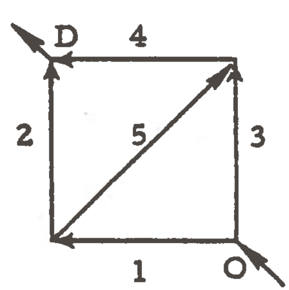
2.9) $$KC_I - \frac{\beta^2}{4\Sigma} .$$

<u>Theorem 2.1</u>. Optimal descriptive and normative solutions to P_I are given by (2.4), (2.5) and (2.7), (2.8), respectively. The difference in total OD flow costs of both solutions is

$$\frac{1}{4\Sigma} (B_1+B_2-B_3-B_4)^2 .$$

3. <u>Problem P_{II}: Linear costs with a transversal link.</u> Denote by P_{II} the problem characterized by $\{M, f_j, K\}$ where the network M with a transversal fifth link is as follows:

M



Let the cost function

$$f_5(x) = mx + n \qquad (m, n \in \mathbb{R}^+ \text{ if not zero}). \tag{3.1}$$

Denote by Z the third, now available, OD path $\{1,5,4\}$. Let the descriptive optimal flows of P_{II} along paths L, R, and Z be denoted by x_1, x_2 and x_3 respectively, $x_1+x_2+x_3 = K$.

To simplify the notation, let

$$\begin{cases} \psi_1 = (A_1+A_3)(B_2-B_4-n) , \\ \psi_2 = (A_2+A_4)(B_3-B_1-n) , \end{cases} \tag{3.2}$$

and denote the following determinants as follows:

$$\Delta = A_1A_4 - A_2A_3 , \tag{3.3}$$

$$\lambda = \begin{vmatrix} A_2+A_4+m & m \\ m & A_1+A_3+m \end{vmatrix} = (A_1+A_3)(A_2+A_4)+m\Sigma > 0 . \tag{3.4}$$

Then

$$\begin{cases} x_1 = -\frac{1}{\lambda}\left[\psi_1 + \beta m - K[A_4(A_1+A_3) + m(A_3+A_4)]\right] \\ x_2 = -\frac{1}{\lambda}\left[\psi_2 - \beta m - K[A_1(A_2+A_4) + m(A_1+A_2)]\right] \\ x_3 = \frac{1}{\lambda}\left[-\Delta K + \psi_1 + \psi_2\right] \end{cases} \tag{3.5}$$

if the following conditions in K, which ensure positive x_i, hold:

(3.6) $$\gamma = \mathrm{Max}\left\{\frac{\psi_1 + m\beta}{(A_1+A_3)A_4+(A_3+A_4)m}, \frac{\psi_2 - m\beta}{(A_2+A_4)A_1+(A_1+A_2)m}\right\} < K ,$$

(3.7) $$0 < -\Delta K + \psi_1 + \psi_2 .$$

The expression on the right of (3.7) is fundamental in what follows and will be denoted by

(3.8) $$\theta \equiv -\Delta K + \psi_1 + \psi_2 .$$

It is easily verified by (2.1), (2.4) that

(3.9) $$\frac{\theta}{\Sigma} = f_2(x_0)-f_4(K-x_0)-f_5(0) = f_3(K-x_0)-f_1(x_0)-f_5(0) .$$

<u>Theorem 3.1.</u> Given the descriptive assignment problems P_I, P_{II} with an OD flow K greater than the lower bound imposed by (2.2), (3.6). Then the Z path of P_{II} will be utilized with optimal flow $\lambda^{-1}\theta$ only if $\theta > 0$.

4. <u>Linear link aggregation cost and Braess paradox.</u> Assuming positive descriptive optimal flows along all 3 paths of P_{II}, it turns out that the elimination of link 5 and path Z induces the following defections from Z to L and R:

4.1) $$\begin{cases} \delta_1 = x_0 - x_1 = \dfrac{A_1+A_3}{\Sigma}\dfrac{\theta}{\lambda} = \dfrac{A_1+A_3}{\Sigma} x_3 \\ \delta_2 = K - x_0 - x_2 = \dfrac{A_2+A_4}{\Sigma}\dfrac{\theta}{\lambda} = \dfrac{A_2+A_4}{\Sigma} x_3 \end{cases} .$$

We next compute the difference in descriptively optimal travel costs for an individual on path L, R or Z according to the presence or absence of link 5. It turns out that

4.2) $$C_{II}(L) - C_I(L) = C_{II}(R) - C_I(R) = C_{II}(Z) - C_I(L) = \frac{\Delta}{\lambda\Sigma}\theta .$$

<u>Theorem 4.1</u>. On each of the three paths of P_{II}, let the descriptive optimal cost be denoted by C_{II}. Then the <u>aggregation</u> cost of <u>extracting link 5</u> is

4.3) $$Ⓐ = C_I - C_{II} = -\frac{\Delta\theta}{\lambda\Sigma} = -\frac{\Delta}{\lambda}[f_2(x_0) - f_4(K-x_0) - f_5(0)] .$$

<u>Corollary 1</u>. If $\Delta = 0$, Ⓐ $= 0$.

<u>Corollary 2</u>. If $\Delta \neq 0$,

4.4) $$\frac{Ⓐ}{-\Delta} = \frac{x_3}{\Sigma} = \frac{\delta_1}{A_1+A_3} = \frac{\delta_2}{A_2+A_4} = \frac{\theta}{\lambda\Sigma}$$

<u>Theorem 4.2</u>. Necessary and sufficient conditions for the Braess conditions to hold, that is for Ⓐ<0 (with positive flows on all available paths) are given by

(i) $\Delta > 0$,

(ii) OD Flow K bounded by $\text{Max}(\mu, \nu) < K < \frac{1}{\Delta}(\psi_1 + \psi_2)$,

where μ, ν are given by (2.2), (3.6) respectively, and the ψ_i by (3.2).

Corollary. P_{II} has a built-in "normative" valve, in the sense that if, descriptively, $C_{II} - C_I > 0$, the total cost link 5 imposes on "society" is bounded above:

(4.5) $$K(C_{II} - C_I) = \frac{K\Delta\theta}{\lambda\Sigma} < \frac{(\psi_1 + \psi_2)^2}{\lambda\Sigma} ,$$

for all feasible Braess flows K.

5. Polynomial Costs. We generalize problems P_I and P_{II} by associating link costs $f_i(x)$, with the stipulation that these be polynomials with coefficients in $\mathbb{R}^+$ (if nonzero), hence convex along with their derivatives and integrals over $\mathbb{R}^+$.

Again we express the requirement that the descriptive optional flow to P_I be interior to the constraint set by:

(5.1) $$\begin{cases} f_1(K) + f_2(K) > f_3(0) + f_4(0) \\ f_3(K) + f_4(K) > f_1(0) + f_2(0) . \end{cases}$$

For such an optimal flow x_0 on L, we must also have

(5.2) $$f_1(x_0) + f_2(x_0) = f_3(K - x_0) + f_4(K - x_0)$$

Again let x_1, x_2, x_3 denote the optimal descriptive flows for P_{II}. Sufficient conditions for the interiority of x_1 x_2 are

(5.3) $$\begin{cases} f_4(K) + f_5(K) > f_2(0) \quad , \\ f_1(K) + f_5(K) > f_3(0) \quad . \end{cases}$$

Corresponding to our former positive flow on link 5, $\theta > 0$, we must have

(5.4) $$f_2(x_0) - f_4(K-x_0) = f_3(K-x_0) - f_1(x_0) > f_5(0) \; .$$

While the equilibrium conditions are given by

(5.5) $$f_2(x_1) - f_4(K-x_1) = f_3(x_2) - f_1(K-x_2) = f_5(K-x_1-x_2) \; ,$$

which can be stated as:

(5.6) $$f_2(x_0-\delta_1) - f_4(K-x_0+\delta_1) = f_3(K-x_0-\delta_2) - f_1(x_0+\delta_2) = f_5(\delta_1+\delta_2) \; .$$

Our present aim is to estimate the aggregation cost Ⓐ $= C_I - C_{II}$. By using the convexity properties of the f_i, we can prove:

<u>Lemma 5.1.</u> If K is bounded by (5.1), (5.3), (5.4),

(5.7) $$\begin{cases} -\overline{A}_4\delta_1{}^2 - \overline{A}_3\delta_2{}^2 \leq Ⓐ + A_4\delta_1 - A_3\delta_2 \leq 0 \; , \\ -\overline{A}_2\delta_1{}^2 - \overline{A}_1\delta_2{}^2 \leq Ⓐ - A_2\delta_1 + A_1\delta_2 \leq 0 \; . \end{cases}$$

where

$$
(5.8)\quad \begin{cases}
f_5'(0) = m \\
f_i'(x_0) = A_i \qquad (i = 1,2)\,, \\
f_i'(K-x_0) = A_i \qquad (i = 3,4)\,, \\
f_1''(x_0+\delta_2) = \overline{A}_1,\ f_2''(x_0) = \overline{A}_2,\ f_3''(K-x_0) = \overline{A}_3,\ f_4''(K-x_0+\delta_1) = \overline{A}_4,\ f_5''(\delta_1+\delta_2) = \overline{A}_5\,.
\end{cases}
$$

Let

$$\theta^* \equiv f_2(x_0) - f_4(K-x_0) - f_5(0) = f_3(K-x_0) - f_1(x_0) - f_5(0) > 0\,,$$

by (5.4).

Lemma 5.2. If $\lambda > 0$ is given by (3.4), with the A_i, m as in (5.8),

$$-\gamma_{11}\delta_1^2 - \gamma_{12}\delta_2^2 < (A_1+A_3)\theta^* - \lambda\delta_1 < \alpha_{11}\delta_1^2 + \alpha_{12}\delta_2^2$$

$$-\gamma_{21}\delta_1^2 - \gamma_{22}\delta_2^2 < -(A_2+A_4)\theta^* + \lambda\delta_2 < \alpha_{21}\delta_1^2 + \alpha_{22}\delta_2^2\,,$$

where the α_{ij}, γ_{ij} are polynomials in A_i and m.

Theorem 5.1. The expression $-\frac{\Delta}{\lambda}\theta^*$ is an estimate of Ⓐ, in the sense that the difference between Ⓐ and it is bounded below and above by 2nd degree polynomials in the δ_i, as in lemma 5.2.

6. Example of quadratic Braess model.

On network M, let

$$\text{Flow } K = 6$$

$$f_1 = f_4 = x^2 + 10x$$

$$f_2 = f_3 = x^2 + x + 66$$

$$f_5 = x^2 + x + 10$$

Solution to P_I: $x_0 = 3$, $C_I = 117$

Solution to P_{II}: $x_1 = x_2 = x_3 = 2$, $C_{II} = 128$.

Hence

$$\text{Ⓐ} = C_I - C_{II} = -11.$$

Computing the estimate of Theorem 5.1, gives the value

$$-\frac{\Delta\theta^*}{\lambda} = -10.44 ,$$

the error being thus ≈ 5% Ⓐ.

7. Generalization to two OD chains. All above results can be trivially generalized to a series aggregation model, with one transversal link between two chains sharing the same origin and destination.

Graphs illustrating theorems 4.2, 4.3

$$y = \text{Ⓐ} = C_I - C_{II} = -\frac{\Delta\theta}{\lambda\Sigma} = rk + s \ ,$$

where

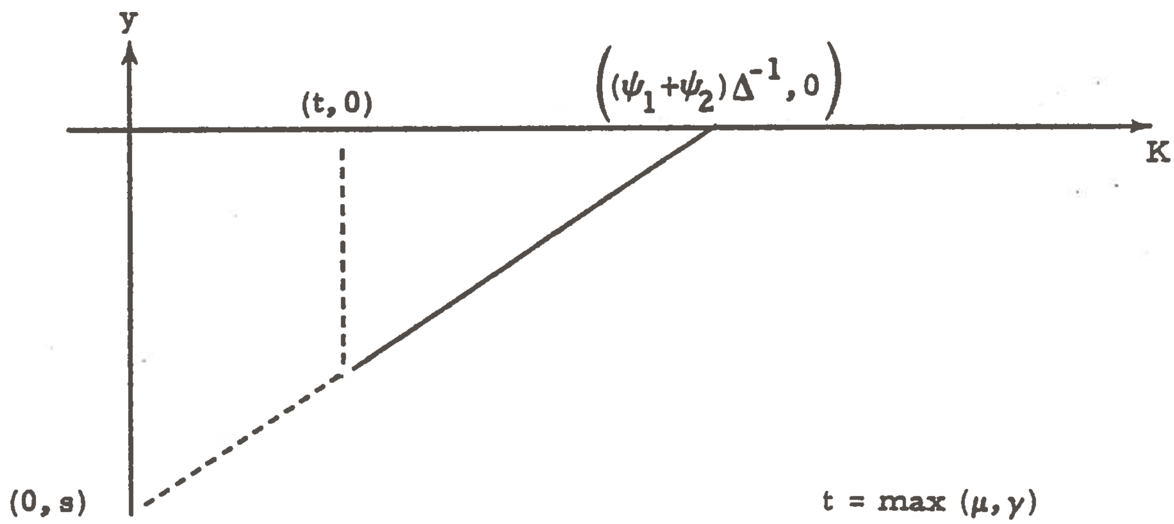
flow K $\geq$ Max (μ,γ) by (2.2), (3.6)

slope $r = \frac{\Delta}{\lambda\Sigma} \geq 0$

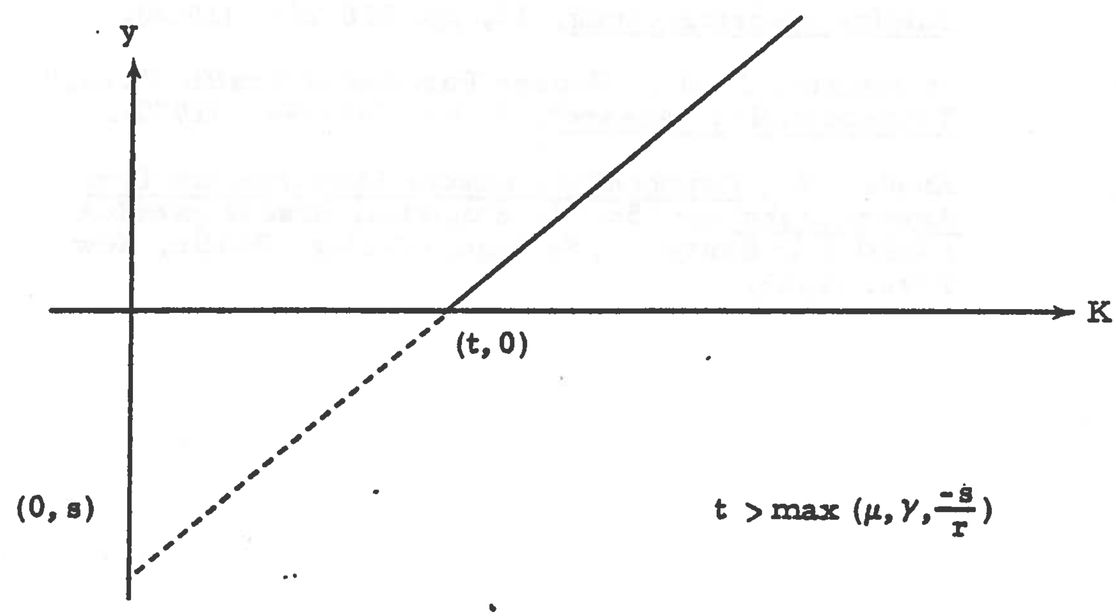
$$s = \frac{-\Delta(\psi_1+\psi_2)}{\lambda\Sigma}$$

Assume $\neq 0$, otherwise $y \equiv 0$.

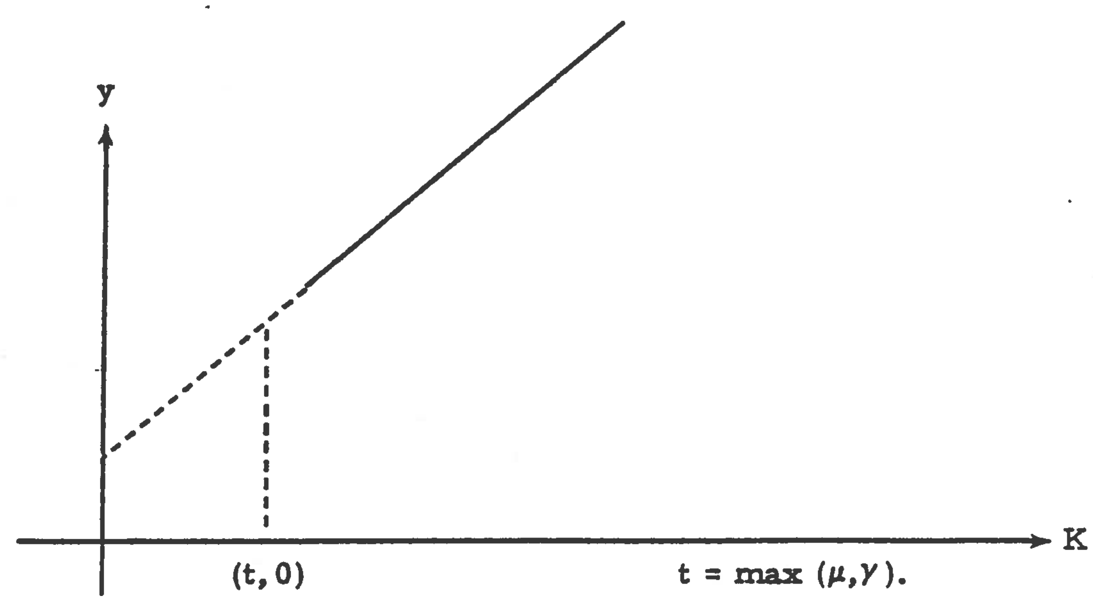
Case a) Braess situation: $y < 0 \Rightarrow \Delta > 0 \Rightarrow \psi_1+\psi_2 > 0 \Rightarrow s < 0$.



Case b) Ⓐ > 0, $s < 0$. Then $rk + s \geq 0 \Rightarrow K \geq \frac{-s}{r}$.



Case c) $y > 0$, $s > 0$.



<u>BIBLIOGRAPHY</u>

1 Braess, D., "Über ein Paradoxen der Verkhersplanung," <u>Unteinehmensforschung</u>, 12, pp. 258-268. (1968).

2 Murchland, J. D., "Braess Paradox of Traffic Flow," <u>Transportation Research</u>, 4, pp. 391-394. (1970).

3 Knödel, W., <u>Graphentheoretische Methoden und ihre Anwendungen</u> (pp. 56-59: empirical Braess paradox situation in Stuttgart), Springer-Verlag, Berlin, New York. (1969)

PAPER 7

BOUNDS AND ESTIMATES FOR AVERAGE SPEED PER OD PATH

1. Introduction

In a large transportation network, with n_1 origins, n_2 destinations, and an average of p paths joining each OD pair, the computation of the average speed $\bar{v}$ per OD path requires summing $p n_1 n_2$ products $f_\alpha^{OD} v_\alpha^{OD}$, where f_α^{OD} is the quantity shipped from O to D along path α with average speed v_α^{OD}.

By aggregating flows out of every origin and into every destination, this paper gives upper and lower bounds, M and m, and an estimate $\frac{M+m}{2}$ of $\bar{v}$ requiring the summation of only $2(n_1+n_2)$ products.

If the origins and destinations are disjoint, the path characteristics are completely determined and expressed in terms of link characteristics (usually the only information available). If some nodes are both origins and destinations, then additional information regarding the flow into or out of these nodes is required to determine path flows.

2. Definition of Problem

Given a network N with nodes $\{i\}$, directed links $\{(i, j)\}$ and link characteristics for $(i, j) \in N$ as follows:

$$\begin{cases} \text{flow } f_{ij} \text{ in units of homogeneous product} \\ \text{speed } v_{ij} \text{ in miles/hr} \\ \text{distance } d_{ij} \text{ in miles.} \end{cases} \qquad (1.1)$$

The last two define

$$\text{link time } t_{ij} = d_{ij}/v_{ij} \text{ in hours.} \qquad (1.2)$$

The product is assumed to be shipped from various origins to various destinations along various paths α of N. We are interested in expressing the average speed $\bar{v}$ per OD path α in terms of link characteristics. When an explicit formula is not available owing to the complexity of the network, we will define upper and lower bounds, and hence an estimate of $\bar{v}$, by aggregating the flows out of every origin and into every destination.

If

$$\begin{cases} O = \{\text{origins} \in N\} \\ D = \{\text{destinations} \in N\}, \end{cases} \qquad (1.3)$$

and $k \in O$, $p \in D$, let $\alpha = [k, p]$ denote a path from k to p in N. Let

$$Q = \{\alpha | \text{all } k \in O, \text{ all } p \in D\}. \qquad (1.4)$$

Let

$$v_{\alpha}^{kp} = \text{average velocity from k to p along } \alpha. \qquad (1.5)$$

$$f_{\alpha}^{kp} = \text{OD flow from k to p along } \alpha. \qquad (1.6)$$

Then

$$\bar{v} = \sum_{\alpha \epsilon Q} f_{\alpha}^{kp} v_{\alpha}^{kp} \Big/ \sum_{\alpha \epsilon Q} f_{\alpha}^{kp} = \frac{Z}{F} . \qquad (1.7)$$

3. <u>General situation</u>

If $k \epsilon O$, let

$$\alpha(k) = \{\text{paths } \alpha \epsilon Q \text{ originating in } k\}. \qquad (2.1)$$

If $p \epsilon D$, let

$$\beta(p) = \{\text{paths } \alpha \epsilon Q \text{ terminating in } p\}. \qquad (2.2)$$

Then Z can be expressed either as

$$Z = \sum_{k \epsilon O} \left[\sum_{\alpha \epsilon \alpha(k)} f_{\alpha}^{kp} v_{\alpha}^{kp} \right] , \qquad (2.3)$$

or as

$$Z = \sum_{p \epsilon D} \left[\sum_{\alpha \epsilon \beta(p)} f_{\alpha}^{kp} v_{\alpha}^{kp} \right] . \qquad (2.4)$$

If $p \epsilon D$, let the total flow into p be denoted by

$$R^{p} = \sum_{\alpha \epsilon \beta(p)} f_{\alpha}^{kp} . \qquad (2.5)$$

If $k \epsilon O$, let the total flow out of k be denoted by

$$S^{k} = \sum_{\alpha \epsilon \alpha(k)} f_{\alpha}^{kp} . \qquad (2.6)$$

Then the total flow through N,

$$F = \sum_{p \epsilon D} R^p = \sum_{k \epsilon O} S^k . \tag{2.7}$$

We relate S^k and R^p to link characteristics as follows:

If j is an arbitrary node of N, let

$$\begin{cases} A(j) = \{ i \; \epsilon \; N \mid (j, i) \; \epsilon \; N \} \\ B(j) = \{ i \; \epsilon \; N \mid (i, j) \; \epsilon \; N \} \end{cases} \tag{2.8}$$

Then the flows entering and leaving j are respectively

$$E_j = \sum_{i \epsilon B(j)} f_{ij} \tag{2.9}$$

and

$$F_j = \sum_{i \epsilon A(j)} f_{ji} . \tag{2.10}$$

Then

$$f_j = E_j - F_j \qquad (j \epsilon N) , \tag{2.11}$$

denotes the demand (if positive) or supply (if negative) at j and is expressed in terms of link characteristics. Note that we also have:

$$f_j = R^j - S^j \qquad (j \epsilon O \cap D) , \tag{2.12}$$

$$\begin{cases} f_j = -S^j < 0 & (j \epsilon O \cap \overline{D}) , \\ f_j = R^j > 0 & (j \epsilon \overline{O} \cap D), \\ f_j = 0 & (j \epsilon \overline{O \cap D}) . \end{cases} \tag{2.13}$$

4. **Bounds for $\bar{v}$**

In order to express the path velocities in terms of link characteristics for each OD path α, we aggregate in series times and distances. Thus if

$$\alpha = k(k, r_1)\, r_1\, (r_1, r_2)\, r_2 \ldots r_m (r_m, p) p, \tag{3.1}$$

let

$$D^{\alpha}_{kp} = d_{kr_1} + d_{r_1 r_2} + \ldots + t_{r_m p} \;. \tag{3.2}$$

$$T^{\alpha}_{kp} = t_{kr_1} + t_{r_1 r_2} + \ldots + t_{r_m p} \;. \tag{3.3}$$

Then

$$v^{kp}_{\alpha} = D^{\alpha}_{kp} \,/\, T^{\alpha}_{kp} \tag{3.4}$$

The following maxima and minima are thus computed by (3.2), (3.3), (3.4) in terms of the d_{ij} and t_{ij}:

$$V_k = \max_{\alpha \in \alpha(k)} v^{kj}_{\alpha}, \quad v_k = \sum_{\alpha \in \alpha(k)} v^{kj}_{\alpha} \qquad (k \in O) \tag{3.5}$$

$$U_p = \max_{\alpha \in \beta(p)} v^{jp}_{\alpha}, \quad u_p = \sum_{\alpha \in \beta(p)} v^{jp}_{\alpha} \qquad (p \in D) \tag{3.6}$$

Let

$$\begin{cases} M = \min\left\{ \sum_O S^k V_k, \sum_D R^p U_p \right\} \\ m = \max\left\{ \sum_O S^k v_k, \sum_D R^p u_p \right\} \end{cases} \tag{3.7}$$

Then by (2.3), (2.4), we have

$$m \leq Z \leq M \tag{3.8}$$

for all sets of path flows compatible with (1.1) and one of the sets

$$\{R^j\} \quad \underline{\text{or}} \quad \{S^j\} \qquad (j \in O \cap D) \tag{3.9}$$

__Theorem.__ The average speed $\overline{v}$ per OD path is constrained by

$$\frac{m}{F} \leq \overline{v} \leq \frac{M}{F} \tag{3.10}$$

where the two bounds can be computed from the available link characteristics (1.1) and additional information (3.9), by the formulas given by (3.2) to (3.7). (This involves the computation of $2(n_1+n_2)$ instead of $\lambda n_1 n_2$ products, where $n_1 = |O|$, $n_2 = |D|$, λ = average number of paths per OD pair.)

__Proof.__ Formulas (3.2) to (3.7) determine m and M in terms of (1.1) and $\{S^k | k \in O\}$, $\{R^p | p \in D\}$. By (2.9) to (2.13), S^k for $k \in O \cap \overline{D}$, R^p for $p \in \overline{O} \cap D$ are known in terms of (1.1), while (3.9) and (2.12) determine the remaining S^k and R^p. Finally, by (2.7), F is determined by $\{R^p\}$.

__Corollary.__ A "reasonable" estimate for $\overline{v}$ is given by

$$\frac{M + m}{2F} \tag{3.11}$$

5. __Special cases__

(i) __Unique path α for each OD pair.__ We can let

We can then let $p = \tau(k) = \hat{k}$. Then

$$f^{k\hat{k}} = S^k = F_k - E_k \qquad (k \epsilon O) \qquad (4.6)$$

and explicitly

$$\bar{v} = \sum_{k \epsilon O} \frac{D_{k\hat{k}}}{T_{k\hat{k}}} (F_k - E_k) \Big/ \sum_{k \epsilon O} (F_k - E_k) . \qquad (4.7)$$

(iv) <u>$O \cap D = \phi$ and mapping</u> $\sigma : p \epsilon D \rightarrow k \epsilon O$ if $[k,p] \epsilon Q$, is <u>single valued.</u> <u>Moreover, path α is unique per OD pair.</u>

Let $k = \sigma(p) = \bar{p}$. Then

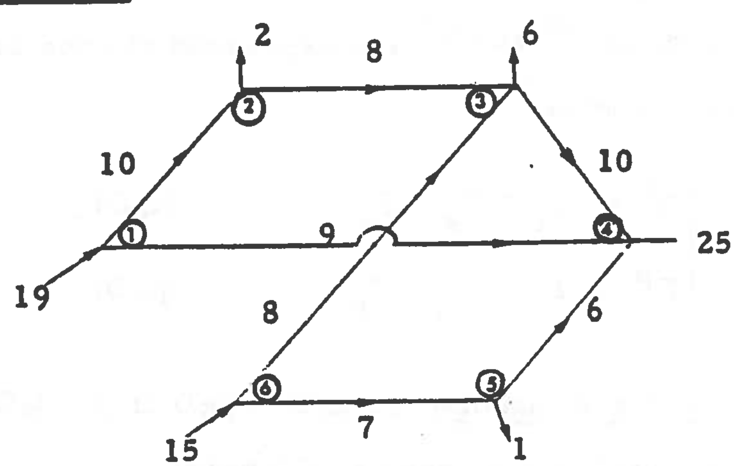
$$f^{\bar{p}p} = R^p = E_p - F_p \qquad (p \epsilon D) \qquad (4.8)$$

and explicitly

$$\bar{v} = \sum_{p \epsilon D} \frac{D_{\bar{p}p}}{T_{\bar{p}p}} (E_p - F_p) \Big/ \sum_{p \epsilon D} (E_p - F_p). \qquad (4.9)$$

6. <u>Illustrations</u>

Special Case (ii)



$O = \{1, 6\} \qquad D = \{2, 3, 4, 5\}$.

$$f^{kp}_{[kp]} = f^{kp}, \quad v^{kp}_{[kp]} = v^{kp}, \quad \text{so that}$$

$$Z = \sum_{\substack{k\epsilon O \\ p\epsilon D}} f^{kp} v^{kp} . \tag{4.1}$$

while

$$S^k = \sum_{p\epsilon D} f^{kp}, \quad R^p = \sum_{k\epsilon O} f^{kp} . \tag{4.2}$$

Again let D_{kp}, T_{kp} denote the (unique) aggregated distance and time ((3.2), (3.3)) between k and p; then

$$v^{kp} = D_{kp}/T_{kp} , \tag{4.3}$$

$$\begin{cases} V_k = \max\limits_{j\epsilon D} v^{kj}, & v_k = \min\limits_{j\epsilon D} v^{kj} \quad (k\epsilon O), \\ U_p = \max\limits_{j\epsilon O} v^{jp}, & u_p = \min\limits_{j\epsilon O} v^{jp} \quad (p\epsilon D) . \end{cases} \tag{4.4}$$

(ii) $O \cap D = \phi$. Since it is impossible for a node to be both an origin and destination, all S^k and R^p are expressed at once by (2.13) in terms of link characteristics:

$$\begin{cases} S^k = -f_k = F_k - E_k & (k\epsilon O) , \\ R^p = f_p = E_p - F_p & (p\epsilon D) . \end{cases} \tag{4.4}$$

(iii) $O \cap D = \phi$ and mapping $\tau : k\epsilon O \longrightarrow p\epsilon D$ if $[k, p]\epsilon Q$, is single valued. Moreover, path α is unique per OD pair.

$link_{ij}$	d_{ij}	f_{ij}	v_{ij}	t_{ij}
12	100	10	25	4
23	200	8	40	5
34	150	10	50	3
14	300	9	30	10
63	400	8	50	8
65	60	7	60	1
54	140	6	70	2

Link table

node i	E_i	F_i	R^i	S^i
1 } ϵO	0	19		17
6 } ϵO	0	15		15
2 } ϵD	10	8	2	
3 } ϵD	16	10	6	
4 } ϵD	25	0	25	
5 } ϵD	7	6	1	

Flow table

OD table

ij	path α	D_{ij}^{α}	T_{ij}^{α}	v_{α}^{ij}	For direct computation: Two possible sets of f_{α}^{ij}	
12	(1, 2)	100	4	25	2	2
13	[1, 3]	300	9	33	6	3
14	(1, 4)	300	10	30	2	5
	through 2	450	12	37	0	3
63	(6, 3)	400	8	50	8	5
64	through 3	550	11	50	1	1
	through 5	200	3	66	6	6
65	(6, 5)	60	1	60	9	9

$$\{v^{j2}\} = \{25\} \Rightarrow U_2 = u_2 = 25$$

$$\{v^{j3}\} = \{50, 33\} \Rightarrow U_3 = 50, \quad u_3 = 33$$

$$\{v^{j4}\} = \{50, 66, 37, 30\} \Rightarrow U_4 = 66, \quad u_4 = 37$$

$$\{v^{j5}\} = \{60\} \Rightarrow U_5 = u_5 = 60$$

$$\{v^{1j}\} = \{25, 33, 37, 30\} \Rightarrow V_1 = 37, \quad v_1 = 25$$

$$\{v^{6j}\} = \{50, 60, 50, 66\} \Rightarrow V_6 = 66, \quad v_6 = 50$$

$$\sum_{D} R^p U_p = 2(25) + 6(50) + 25(66) + 60 = 2060$$

$$\sum_{D} T^p u_p = 2(25) + 6(33) + 25(37) + 60 = 1233$$

$$\sum_{\sigma} S^k V_k = 17(37) + 15(66) = 1619$$

$$\sum_{\sigma} S^k v_k = 17(25) + 15(50) = 1175$$

$$F = S^1 + S^6 = 34 \; .$$

Thus:

$$1233 \leq F\bar{v} \leq 1619$$

$$37 \leq \bar{v} \leq 47$$

<u>Estimate of $\bar{v}$</u> = 42, by (3.11)

<u>Actual value of $\bar{v}$:</u>

Two columns of possible f_{α}^{ij} compatible with other data are given on the right. Formula (1.7) then yields the values of 42.6 and 42.9 for $\bar{v}$.

Special case (i)

N is line network:

ij	d_{ij}	v_{ij}	t_{ij}	f_{ij}
12	200	30	6.6	4
23	300	20	15	2
34	300	50	6	3
45	400	40	10	3
56	100	60	1.7	2

Let $O = \{1,2,3\}$, $D = \{2,3,5,6\}$

$S^1 = 4$, $S^2 = 1$, $S^3 = 2$

$R^2 = 3$, $R^3 = 1$, $R^5 = 1$, $R^6 = 2$.

Here $F = 7$.

<u>OD velocity table</u>

$v^{12} = 30$

$v^{13} = 23.1$ $\quad v^{23} = 20$

$v^{15} = 31.5$ $\quad v^{25} = 32.3$ $\quad v^{35} = 43.7$

$v^{16} = 33.1$ $\quad v^{26} = 33.7$ $\quad v^{36} = 45.3$

$V_1 = 33.1$, $v_1 = 23.1$

$V_2 = 33.7$, $v_2 = 20$

$V_3 = 45.3$, $v_3 = 43.7$

$U_2 = 30$ $\quad u_2 = 30$

$U_3 = 23.1$ $\quad u_3 = 20$

$U_5 = 43.7$ $\quad u_5 = 31.8$

$U_6 = 54.3$ $\quad u_6 = 33.1$

Formula (3.10) gives

$$29.7 \le \bar{v} \le 35.3$$

and estimate

$$\bar{v} = 32.5$$

for all compatible f^{ij} .

a) Let $f^{12} = 3$, $f^{16} = 1$, $f^{23} = 1$, $f^{35} = 1$, $f^{36} = 1$

$\bar{v} = 33.2$

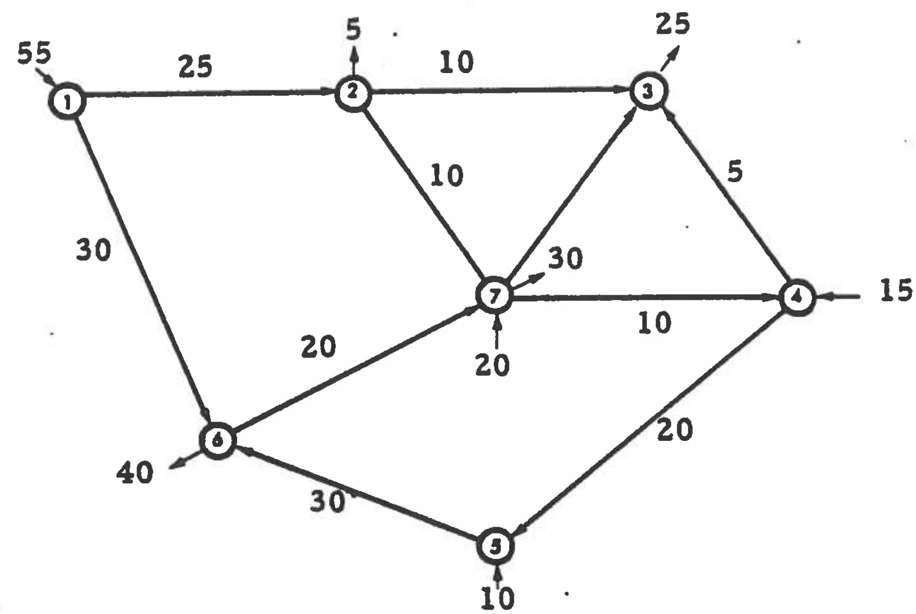
b) Let $f^{12} = e$, $f^{13} = 1$, $f^{25} = 1$, $f^{36} = 2$

$\bar{v} = 33.7$

c) Let $f^{12} = 3$, $f^{15} = 1$, $f^{23} = 1$, $f^{36} = 2$

$\bar{v} = 33.2$

General case



$O = \{1,4,5,7\}$ $D = \{2,3,6,7\}$ $O \cap D = \{7\}$.

Link table

ij	d_{ij}	v_{ij}	t_{ij}	f_{ij}
12	100	50	2	25
16	400	40	10	30
23	250	50	5	10
27	100	25	4	10
43	50	20	2.5	5
45	160	40	4	20
56	210	35	6	30
67	360	30	12	20
74	280	70	4	10
73	120	60	2	10

Flow table

K	S^k	R^k	E_k	F_k
1	55		0	55
4	15		10	25
5	10		20	30
7	20	30	30	20
2		5	25	20
3		25	25	0
6		40	60	20

$F = 100$

OD table

ij	α	D_{ij}^{α}	T_{ij}^{α}	v_{α}^{ij}	For direct computation: A possible set of f_{α}^{ij}	$v_{\alpha}^{ij} f_{\alpha}^{ij}$
12	(1, 2)	100	2	50	5	250
13	(1, 2, 3)	350	7	50	0	
	(1, 2, 7, 3)	320	8	40	0	
16	(1. 6)	400	10	40	30	1200
17	(1, 2, 7)	200	6	33.3	20	666
	(1, 6, 7)	760	22	34.5	0	
43	(4, 3)	50	2.5	20	5	100
46	(4, 5, 6)	370	10	37	10	370
53	(5, 6, 7, 3)	690	20	34.5	0	
56	(5, 6)	210	6	35	0	
57	(5, 6, 7)	570	18	31.5	10	317
73	(7, 3)	120	2	60	20	1200
76	(7, 4, 5, 6)	650	14	46.4	0	
						4103 => $\bar{v}$ = 41.

$V_1 = \max\{50, 40, 33.3, 34.5\} = 50 \qquad v_1 = 33.3$

$V_4 = \max\{20, 37\} = 37 \qquad v_4 = 20$

$V_7 = \max\{60, 46.4\} = 60 \qquad v_7 = 46.4$

$V_5 = \max\{34.5, 35, 31.7\} = 35 \qquad v_5 = 31.7$

$\Sigma S^k V_k = (55)(50)+(15)(37)+(10)(35)+(20)(60) = 4855$

$\Sigma S^k v_k = (55)(33.3)+(15)(20)+(10)(31.7)+(20)(46.4) = 3378$

$\Rightarrow 33.8 \leq \bar{v} \leq 46.4$

Estimated

$\bar{v} \approx 40.1$

$U_2 = 50 \qquad u_2 = 50$

$U_3 = \max\{50, 40, 20, 34.5, 60\} = 60 \qquad u_3 = 20$

$U_6 = \max\{40, 37, 35, 46.4\} = 46.4 \qquad u_6 = 35$

$U_7 = \max\{33.3, 34.5, 31.7\} = 34.5 \qquad u_7 = 31.7$

$\Sigma R^p U_p = 5(50)+25(60)+40(46.4)+30(34.5) = 4641$

$\Sigma R^p u_p = 5(50)+25(20)+40(35)+30(31.7) = 3101$

C. DIRECTIONS FOR FURTHER RESEARCH

MATHEMATICA has been engaged in research on aggregation in network models in transportation planning over the past 14 months. The principal results of this research are reported in the technical papers of this report. As a natural by-product of this research, we have identified a number of directions for further research which promise useful results. These fall roughly into three categories, which will be described separately below:

I. THEORY OF NETWORK AGGREGATION

II. ALGORITHMS RELATED TO NETWORK AGGREGATION

III. NETWORK AGGREGATION AND TRANSPORTATION PLANNING.

I. THEORY OF NETWORK AGGREGATION

1. The primary tool that has been developed for the bounding of errors due to aggregation is the judicious application of duality concepts from mathematical programming. The usefulness of this tool has been shown for the Hitchcock-Koopmans transportation model combined with appropriate aggregation methods. For the equilibrium model, we can, at present, merely give miniature examples of the application of duality to estimate aggregation error. We propose to continue this research, extending the results to more general networks and to aggregation methods that reflect transportation planning practice.

2. Little or no research has been performed on the "average" or "typical" performance of aggregation methods. The results of such research are needed to make statements about such statistics as the mean and variance of errors due to aggregation. It is proposed that such research be undertaken. To be successful, it needs several essential components. The experimental design of a network model, an aggregation method, and a mathematical technique must be done in such a way to reflect useful applications, current aggregation practice, and realistic computational goals. It appears that these features are available to make such research promising.

II. ALGORITHMS RELATED TO NETWORK AGGREGATION

As part of the past year's research, an algorithm for the user optimized equilibrium model was developed that involves path extraction aggregation. A detailed outline of the algorithm is provided in the final report. The situation with regard to this algorithm is somewhat different from other new algorithms since, although it has not yet been programmed, the structure is so close to known and tested algorithms that past computational experience (reported in "The Sandwich Method," Journal of Optimization Theory, November 1975) is certain to be an accurate guide to its performance.

We propose to program, test, and study the applications of this algorithm. Two primary questions would guide this work: (a) What are the best uses that can be made of an algorithm of this kind in transportation planning? (b) What are the best rules for path extraction to implement the algorithm for large networks? A subsidiary use of the algorithm would be to complement the studies of bounds and biases described under (I) above.

III. NETWORK AGGREGATION AND TRANSPORTATION PLANNING

In our summary report, we have indicated that a complete analysis of an instance of aggregation involves the interaction of four factors: aggregation method, network model, mathematical technique or algorithm, and model application. It was noted that our research to date has slighted the uses of the answers in decision making. However, it is crucial that the impact of network aggregation is measured in terms of the improved performance of network algorithms and in the changes in transportation planning decisions. These decisions are usually not based directly on outputs of the network model solutions, but rather are based on function of these outputs. Typically equilibrium model results, for example, are combined with other data in a cost-benefit/impact model to determine system costs, environmental impacts, energy consumptions, accidents, land uses, displacements, etc. Often these latter measures are presented as overall or slightly broken-out results (e.g. by node). The experience of optimization practitioners suggests that these results will tend to be less sensitive to model changes such as those that are introduced through aggregation.

These observations suggest the following line of inquiry:

- Identify the uses of transportation planning.
- Determine the variables which affect the decisions in these uses.

- Determine the levels of detail at which these variables must be obtained.
- Determine the functions which relate these variables to the outputs of network models.
- Categorize these functions according to the number of outputs involved.
- Assess, thereby, the sensitivity of decision making variables to aggregation.

Alternately, it is desirable to tie down this investigation to the typical applications at one end and to the specific network models at the other end.

REPORT OF INVENTIONS APPENDIX

The work performed under this contract comprised a review, evaluation, and extension of mathematical techniques for network aggregation. A diligent review of the results has shown that the new techniques developed, while an extension of knowledge in the field, do not represent potentially patentable items.

U.S. DEPARTMENT OF TRANSPORTATION
RESEARCH AND SPECIAL PROGRAMS ADMINISTRATION
TRANSPORTATION SYSTEMS CENTER
KENDALL SQUARE, CAMBRIDGE, MA. 02142

OFFICIAL BUSINESS
PENALTY FOR PRIVATE USE, $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF TRANSPORTATION
613

U.S. MAIL

Mr. L. Rose, 831
Transportation Systems Center
Room 271A Building 3