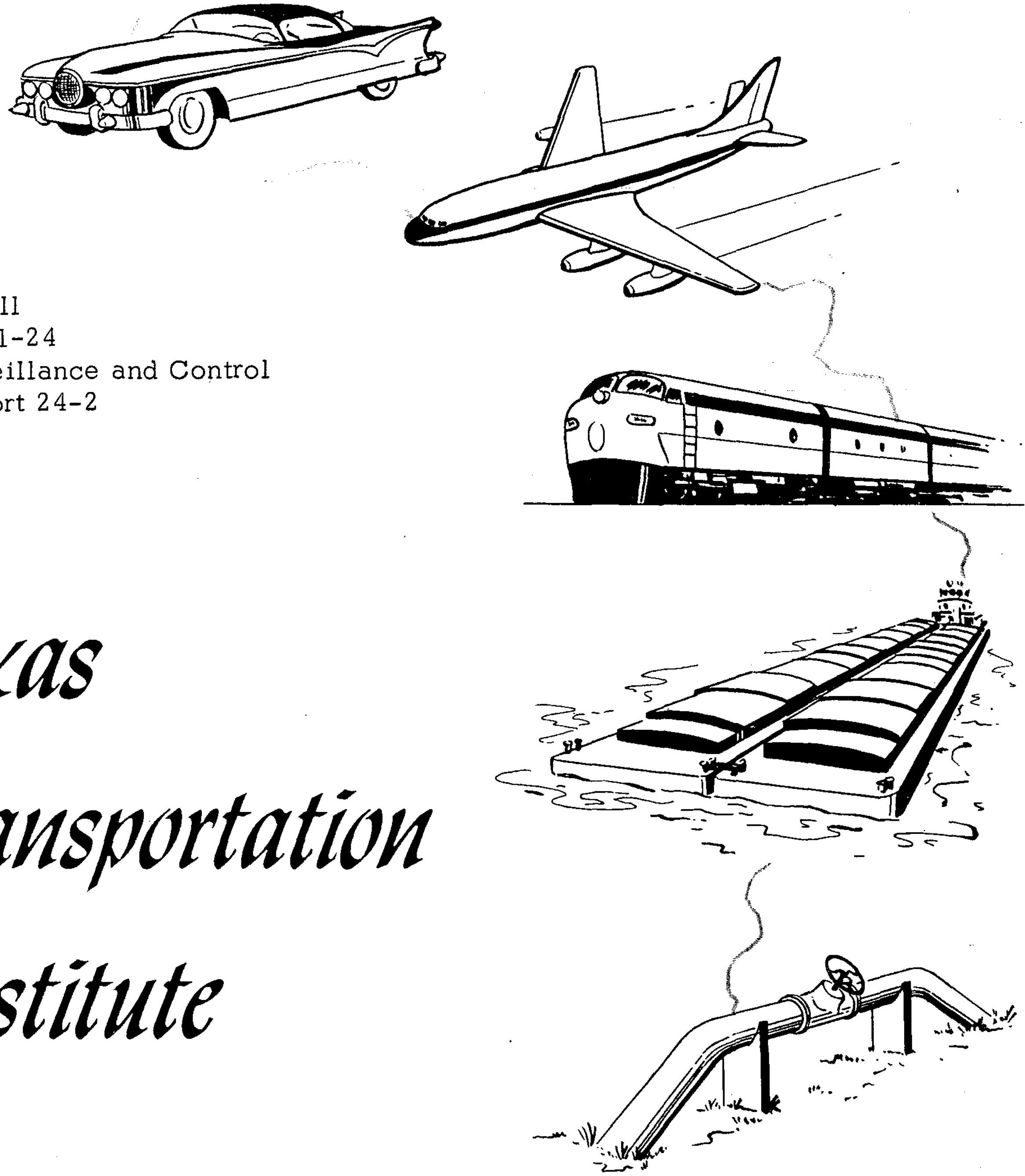
OPTIMUM DISTRIBUTION OF TRAFFIC OVER A CAPACITATED STREET NETWORK



By

Charles Pinnell

Project 2-8-61-24
Freeway Surveillance and Control
Research Report 24-2

Texas Transportation Institute

OPTIMUM DISTRIBUTION OF TRAFFIC OVER A CAPACITATED STREET NETWORK

By

Charles Pinnell

Research Report Number 24-2

Freeway Surveillance and Control
Research Project Number 2-8-61-24

Sponsored by

The Texas Highway Department
In Cooperation with the
U. S. Department of Commerce, Bureau of Public Roads

October, 1964

TEXAS TRANSPORTATION INSTITUTE
Texas A&M University
College Station, Texas

GLOSSARY

CAPACITATED LINK - Network link with specific limitations on the total amount of traffic that can pass over this link.

COPY - Street network having an input traffic volume at a single origin node and corresponding volume outputs at various destination nodes.

E-TYPE CONVERSION - Form of numerical data output by which a decimal fraction is expressed as a power of ten. For example 0.15E 04 equals 0.15×10^{4} and 0.15E-04 equals 0.15×10^{-4}.

FREEWAY - High-type street or highway with full control of access and no intersections at-grade.

ITERATION - Refers to the process of introducing a vector into the basis of the multi-copy linear programming model.

LINK - One-way portion of street network connecting two adjacent nodes.

LINK TRAVEL COST - Time required to move over a given link from one node to another.

MAJOR ARTERIAL - Urban street with intersections at-grade whose function is to provide the through movement of traffic.

MINIMUM PATH ROUTE - Route through a network from a given origin node to a given destination node that requires the least amount of travel time.

NODE - Point of intersection in a street network.

STREET NETWORK - System of nodes and links representing an urban street system.

There are numerous linear programming terms used in the dissertation which are difficult to define in a brief manner. A reader not familiar with this terminology is directed to references (10), (11), (20) and (21) in the reference list.

INTRODUCTION

One of the most significant problems facing those responsible for traffic movement in major urban areas is that of peak hour congestion. Plans for adequate arterial street systems to cope with this problem have been drafted but the required facilities are very costly and difficult to provide. It will require many years to provide the necessary facilities and during this time severe peak hour congestion will continue unless other means are developed to cope with this problem.

A freeway is the major traffic carrier in an arterial street system and is the facility which suffers most from peak hour congestion. Only a fraction of an over-all freeway system has been developed in most urban areas and this partial system contributes greatly to the peak hour overload. The high level of operation provided by a freeway often results in it attracting heavier traffic loads than it will ultimately be required to handle when the entire freeway system is developed.

In order to ease the peak hour congestion problem there is a need to effect maximum utilization of urban arterial systems which include both freeways and at-grade facilities. Thus there is a need for operational control during peak hours which would "spread" the traffic load over the entire arterial system.

Present operational technology in the traffic field does not extend to systems in a broad sense but rather is limited to individual facilities. This approach is inadequate as it does not consider the interaction of the entire system or develop control measures which would achieve optimum system operation. Thus there is a great need to develop a systems analysis approach to the problem of operating arterial street networks.

Systems Analysis Problem

Figure 1 illustrates a typical master plan for a freeway system of a large urban area. As previously noted, such a system exists only on paper and the actual system presently being utilized may resemble that shown in Figure 2. Thus a single freeway facility may serve a rather large area of a city and as a result experience serious peak hour congestion. The area S shown in Figure 2 represents a typical "corridor area" for which peak hour operational controls are needed.

If an expanded view of area A of Figure 2 is taken and the combination of freeway and at-grade arterials is shown as in Figure 3, a basic network to which operational controls may be applied is obtained. There are two basic approaches to operational controls which might be as follows:

1. Traffic could be controlled along the freeway proper by regulating input and output volumes at the various ramps and by controlling the traffic on the freeway through signing and/or control measures.
2. Traffic could be regulated over the entire system which would include both the freeway and the at-grade arterials. With this approach, traffic

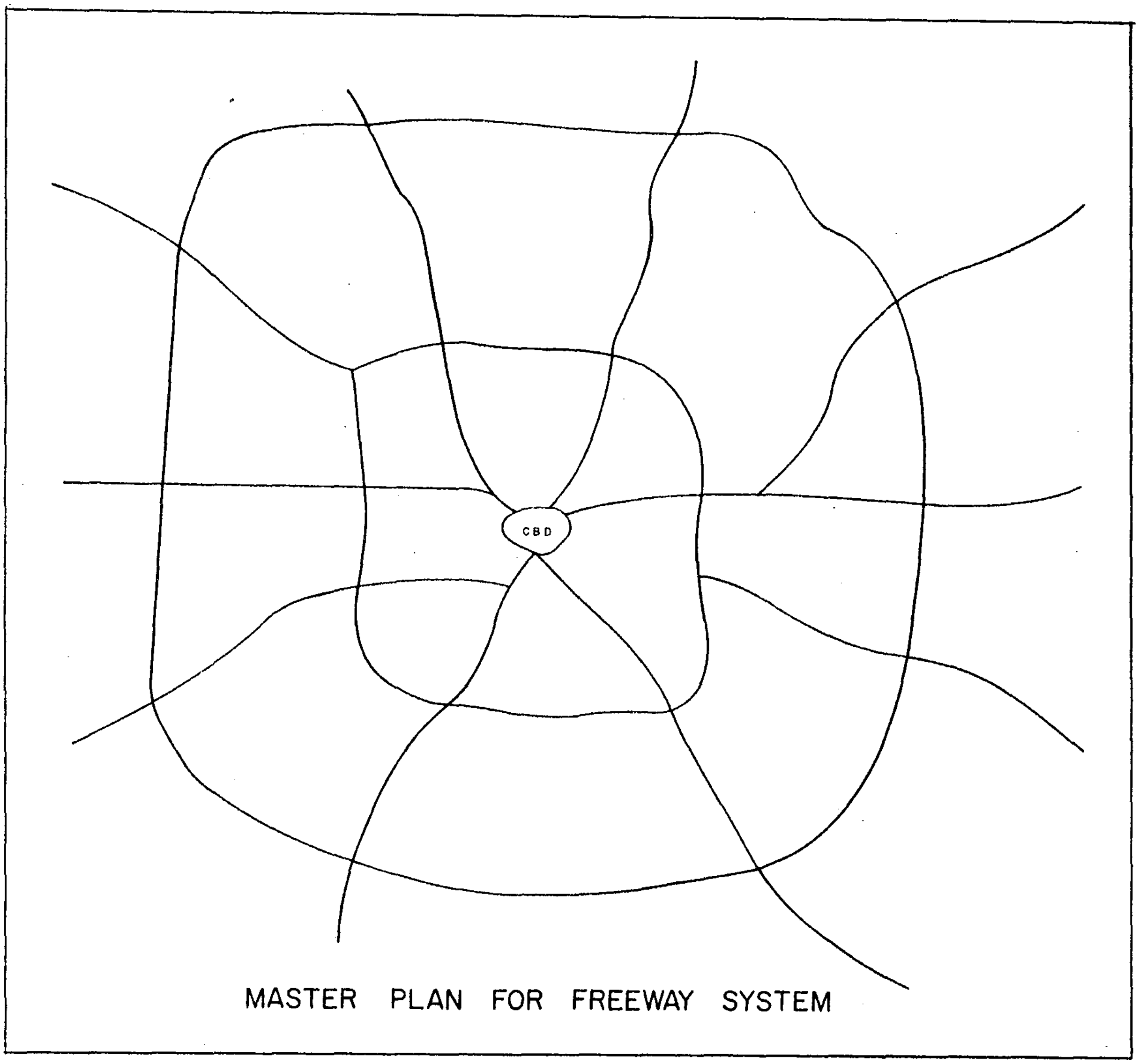


Figure 1

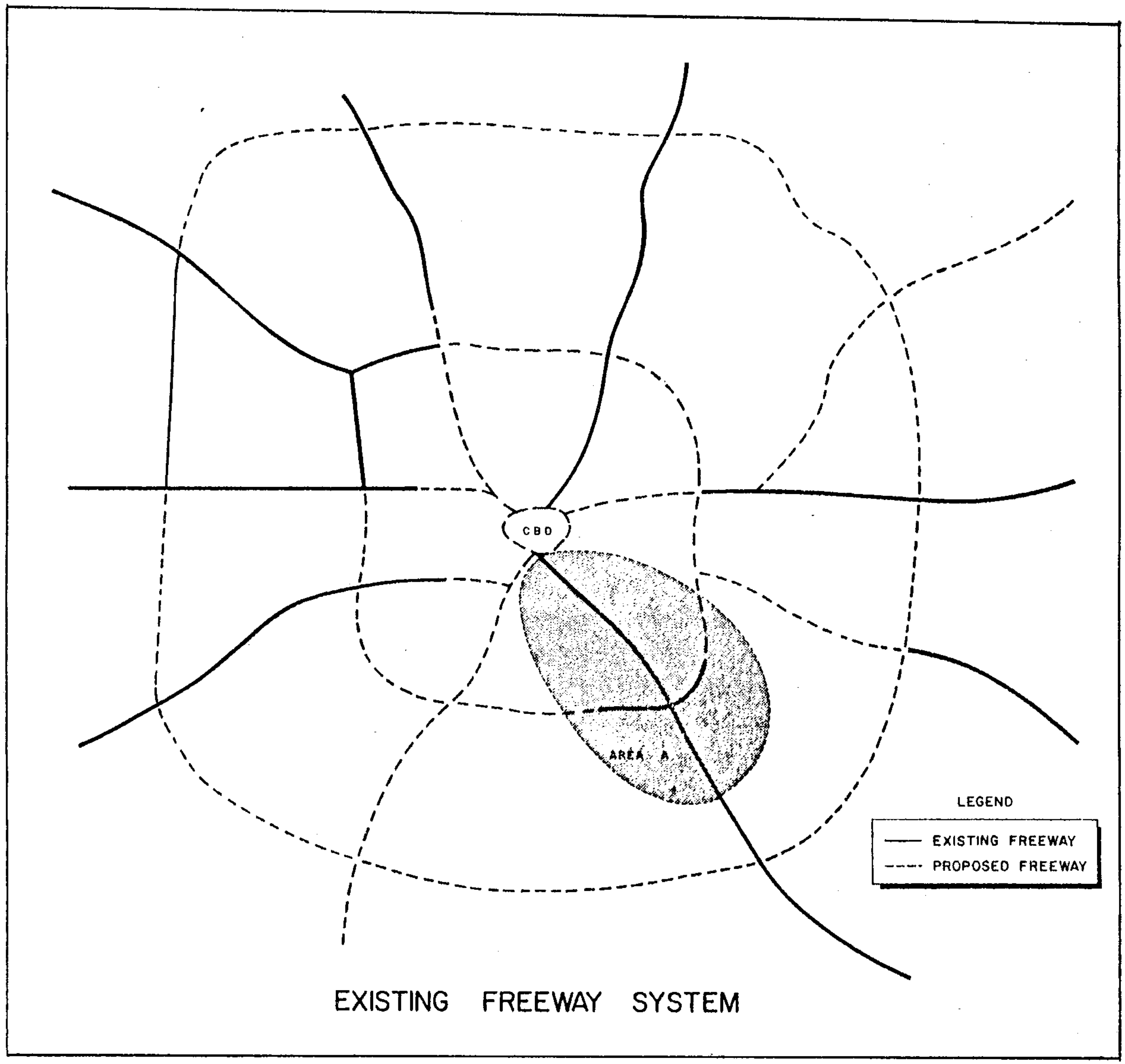
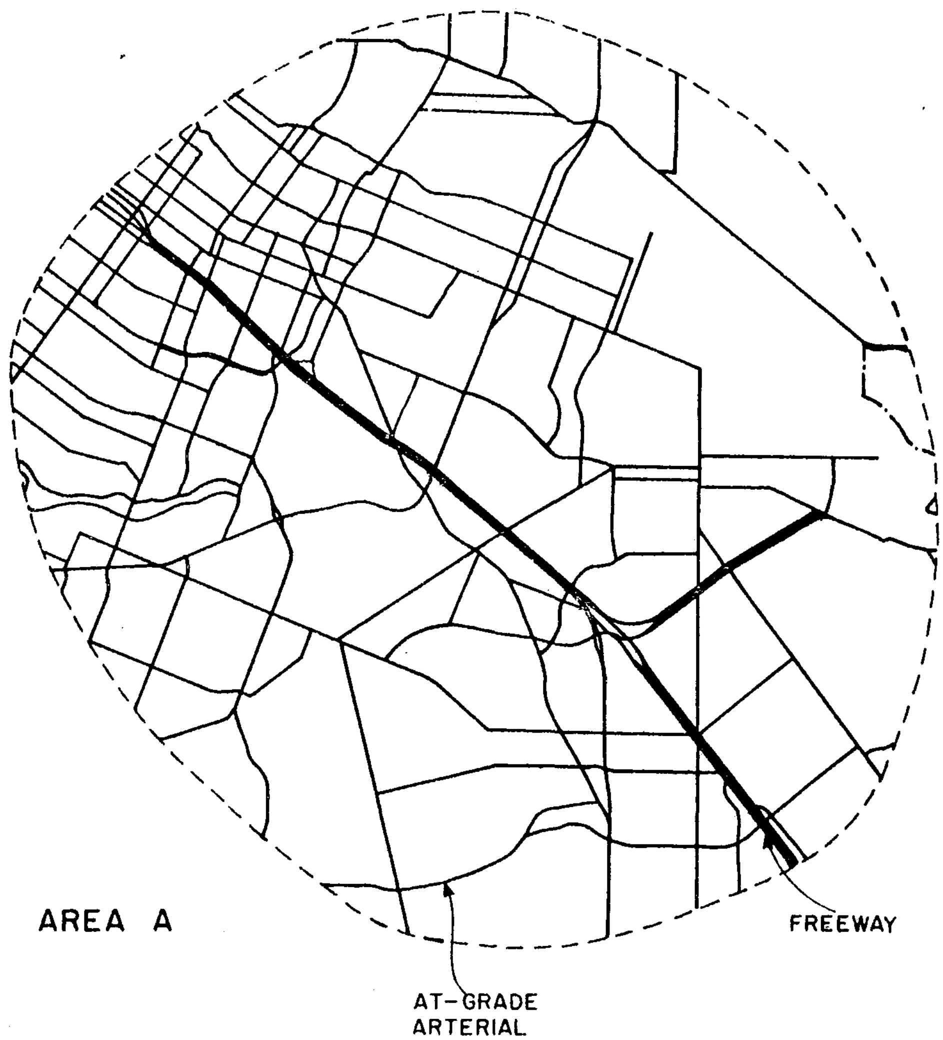

CBD
AREA A
LEGEND
EXISTING FREEWAY
PROPOSED FREEWAY
EXISTING FREEWAY SYSTEM

Figure 2



STREET NETWORK
FOR
SYSTEM ANALYSIS

Figure 3

might be controlled or routed at or near its origin rather than at the freeway and the operational controls required would be for the entire system.

The first technique is already being applied in several surveillance projects across the country and offers a feasible approach to the problem. The second technique would be a more difficult one but would result in better utilization of the entire system of streets. In order to design a control system for an over-all network, however, it would be necessary to know something of the desirable or optimum operations of traffic over the given network. The basic question that must be answered initially is as follows:

> Given a set of traffic inputs and the corresponding outputs for a street network (i.e. - origins and destinations) how should traffic be routed over this network to obtain optimum system operations.

This is the basic problem which will be considered in this report.

Existing Techniques

The need for adequate data from which freeway systems could be planned and designed has led to the development of traffic assignment techniques. These techniques deal with trip desires expressed in terms of origin and destination requirements and a traffic network description defined by nodes, links and impedances. Through the use of high speed digital computers, it is possible to assign traffic volumes to a street network in a manner simulating the daily loading of the system by individual drivers.

The technique of traffic assignment has advanced rapidly since 1952 at which time it was stated (28)* that traffic assignment is considered to be more of an art than a science. At the present time numerous sophisticated assignment programs exist which form a basis for an approach to a systems analysis study of arterial street operation.

The biggest problem in utilizing existing network assignment techniques, however, is that they are designed to work with large systems and necessarily lack some of the sensitivity that would be desired in a critical operational analysis of a comparatively small system.

The basic procedure in an assignment method is to find a minimum path (in terms of travel impedance) through the network from a given origin to a given destination and then to assign the interzonal volumes to this path in some manner. At the present, assignment methods fall into three general categories (28) which are as follows:

1. "All or Nothing" Assignment - all vehicles assigned to the path with the least travel resistance.
2. Diversion Curve Assignment - total number of trips divided between two routes depending upon the relative values of travel resistance on the two routes.
3. Proportional Assignment - total number of trips divided between several

---

* Numbers refer to articles in Reference List.

routes depending upon the relative values of travel resistance for the several routes.

All of the above procedures may utilize capacity restraints which serve to limit the amount of traffic which may be assigned to any given link. The manner in which these restraints are applied varies greatly and is a controversial subject (31).

Requirements for Analysis Technique

A desirable network analysis technique should possess the following characteristics:

1. It should have the ability to determine an optimum means of distributing traffic over a network as well as the ability to simulate network traffic flow.
2. It should provide the ability to place specific capacity limitations on any link in the system.

Critical study of existing assignment techniques indicates that they do not have these desirable characterisitcs. A traffic assignment model developed by the Traffic Research Corporation for Metropolitan Toronto (25) selects as many as four alternate routes between a given origin and destination. Vehicles are then assigned to those routes on a proportional basis which varies as the inverse of the travel time on each route. A capacity restraint feature is provided which increases link travel time as a function of assigned volume. This feature would not enable one to hold link volumes at or below prescribed values.

The procedure used by the Chicago Area Transportation Study (13) might be termed an "all or nothing assignment." After finding a minimum time path between an origin and destination, all trips between the origin and destination are assigned to this path. After all trips in the system have been assigned, individual link travel times are adjusted, new minimum paths computed, and interzonal volumes are again assigned. This iterative process continues until the system stabilizes. There is no provision for minimizing over-all system cost or for holding link volumes to predetermined levels.

The diversion curve technique of traffic assignment (14) also fails to provide a means of assuring optimum distribution of traffic or of holding link volume at or below a predetermined level. In this technique a minimum time path using only freeway links is computed for a trip between a given origin and destination. This same information is computed for a trip moving over the at-grade arterial street system. A ratio of freeway travel time to at-grade arterial travel time is determined and this value is used to select a percent assignment to each type facility from a diversion curve.

Linear Programming Approach

The technique of linear programming which optimizes a linear criterion function subject to a set of linear constraints appears to offer the required approach to the operational analysis of traffic assignments. Such a method offers an exact mathematical procedure for determining traffic distributions which would minimize travel costs and permit restricting volume levels on network links to desired values.

Two approaches to traffic network analysis using a linear programming technique were found in the literature. Heanue (23) proposed a linear programming model which sought to minize system travel time subject to two primary constraints. These constraints were:

(a) Total volume assigned to a network link must be less than or equal to link capacity.

(b) Trips assigned to alternate routes between an origin and destination must sum to total trip desire between the origin and destination.

The primary disadvantage of this model is that a set of alternate routes between each origin and destination must be selected manually. Since an extremely large number of alternate routes exist between each origin and destination in a large system, this choice would be difficult and would introduce an arbitrary procedure into an otherwise exact method.

Charnes and Cooper (11) discuss a class of "coupled" linear programming models which they propose has application to the capacitated transportation problem with specific destinations. This model is discussed further as a "multicopy" model in a presentation made by Charnes (8) at a symposium on the Theory of Traffic Flow conducted by General Motors in 1959. Additional analyses of this model were presented by Pinnell and Satterly (34).

This model satisfies the previously discussed requirements for a satisfactory network analysis method. The model associates a network description or "copy" with each traffic origin yielding a "multi-copy" description of the traffic input data.

The transportation problem when formulated as a general linear programming problem has the disadvantage of requiring an extremely large number of constraint equations and is not practical for solution by present computers. To avoid this problem, Charnes states (11) that a change of variable is possible which permits the large problem to be broken up into two smaller problems. The variable for this problem is the convex combination of copy extreme points and the solution becomes one of determing the proper "mixture" of individual copy extreme point solutions.

The extreme points for each copy can be determined by a minimum path procedure and the proper convex combination of these points is found by a modified simplex technique.

By utilizing this mixing technique, it becomes possible to convert the linear programming model to a practical computer program. This program could utilize existing techniques such as the modified simplex program and the minimum path algorithm. These two techniques could be combined so that the minimum path algorithm furnishes individual copy solutions and the modified simplex technique method solves the "mixing problem."

It is felt that the linear programming model proposed by Charnes fits all the requirements previously discussed and permits a very critical analysis of optimum network traffic flow. It was decided therefore to utilize this model as an approach to the study of optimum distribution of traffic through a capacitated network.

Once some basic knowledge is obtained regarding the optimum distribution of traffic through a network then it will be possible to consider the operational controls necessary to affect such optimum operations. The research work reported herein consisted of developing a computer program for the linear programming model and of utilizing the model to study optimum traffic distribution over a street network.

MATHEMATICAL FORMULATION

General Linear Programming Model

The linear programming technique for traffic network analysis as proposed by Charnes and Cooper (8) has been termed a multi-copy model. This terminology arises from the association of a network copy with each traffic origin on the network. Figure 4A shows a sample traffic network with traffic inputs and corresponding outputs (origins and destinations). Figures 4B and 4C illustrate individual copies associated with the given network.

By selecting the minimization of over-all travel time as the basic criterion and utilizing the network copy concept, the network analysis problem can be formulated as a linear programming problem. Considering an individual copy such as copy 1 of Figure 4, traffic distribution over this network can be formulated as a linear programming problem as follows:

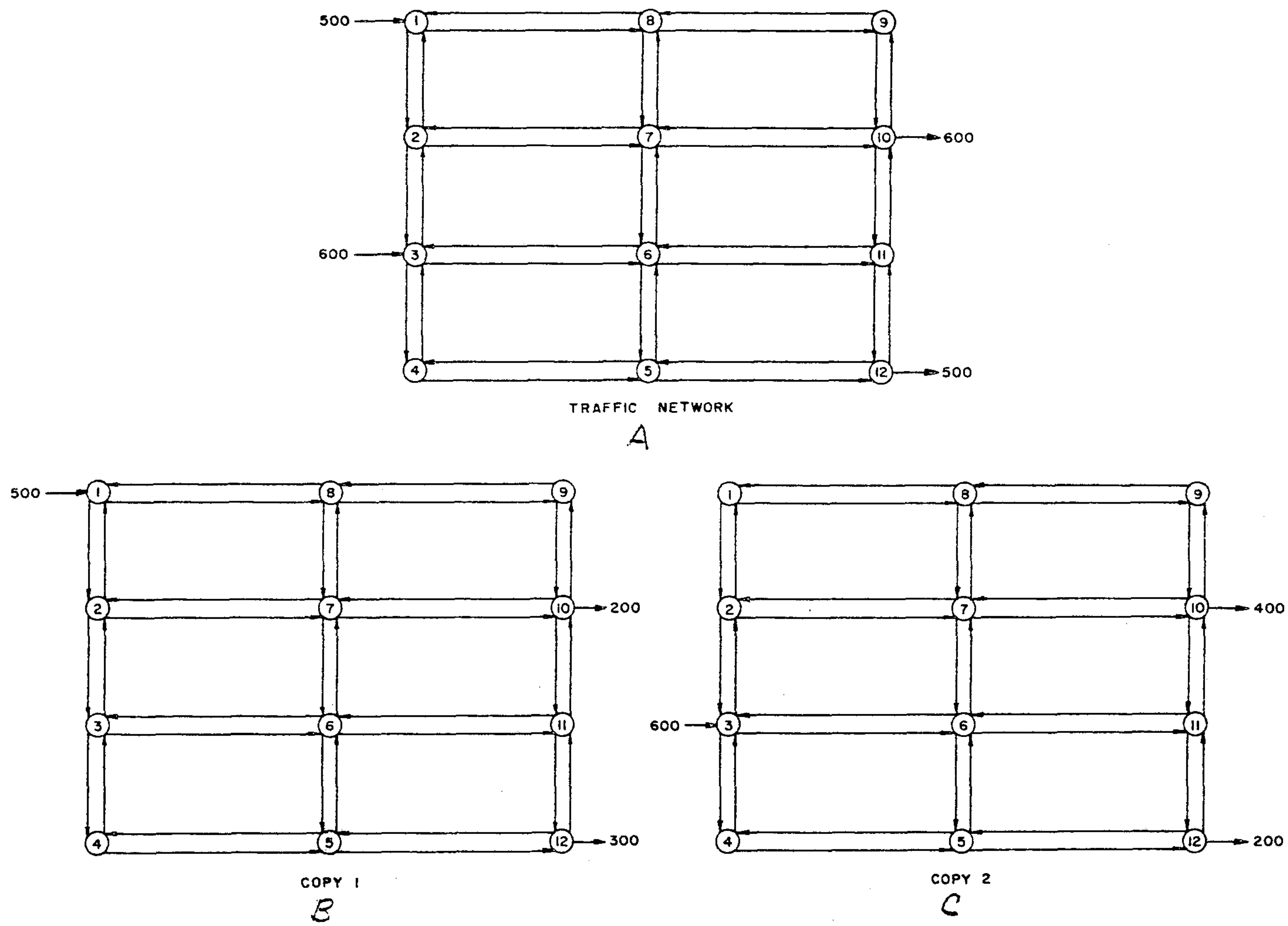
Minimize $$\sum_j c_j x_j \qquad (1)$$

where c_j = Travel cost (in time units) on link j

x_j^{α} = Amount of traffic (number of vehicles) assigned to link j on copy α

subject to $$\sum_j \mathcal{E}_{ij} x_j^{\alpha} = E_i^{\alpha} \qquad (2)$$

where $\mathcal{E}_{ij}$ = Incidence number for the j-th branch at the i-th node (+1 for input, -1 for output, zero if not connected to node)



SAMPLE TRAFFIC NETWORK WITH COPY DESIGNATIONS

FIGURE 4

E_i^α = Influx or efflux at the i-th node on copy α.

The previous constraint equations specify the Kirchhoff node conditions (node input equals node output) for a given copy and assure that the desired origin-destination requirements for the copy are satisfied.

A link in each direction (only one in the case of one-way streets) between adjacent nodes is provided to permit traffic flow in both directions and to permit study of directional flow. Similar formulations can be made for each copy of the network.

Since each of the "copies" utilizes the same network (hence the name "copy"), the traffic flow for the several copies will be superimposed on various links of the system. In order to insure that no single link of the system will be overloaded, individual capacity limitations can be placed on any specific link of the system. This restraint is formulated as follows:

$$\sum_{\alpha} x_j^\alpha \leq \Delta_j \tag{3}$$

where Δ_j = Capacity limitation (number of vehicles) on link j.

The inclusion of a capacity limitation on various branches of the system provides a "coupling constraint" which ties the copies together. The network distribution problem can then be formulated as a general linear programming problem as follows:

$$\text{Minimize} \quad \sum_{\alpha} \sum_{j} c_j \, x_j^\alpha \tag{4}$$

subject to $$\sum_{j} \mathcal{E}_{ij} \; x_j^{\alpha} = E_i^{\alpha} \tag{5}$$

$$\sum_{\alpha} x_j^{\alpha} \leq \Delta_j \tag{6}$$

$$x_j^{\alpha} \geq 0 \tag{7}$$

The structure of this formulation is shown in Figure 5.

Multi-Copy Mixing Model

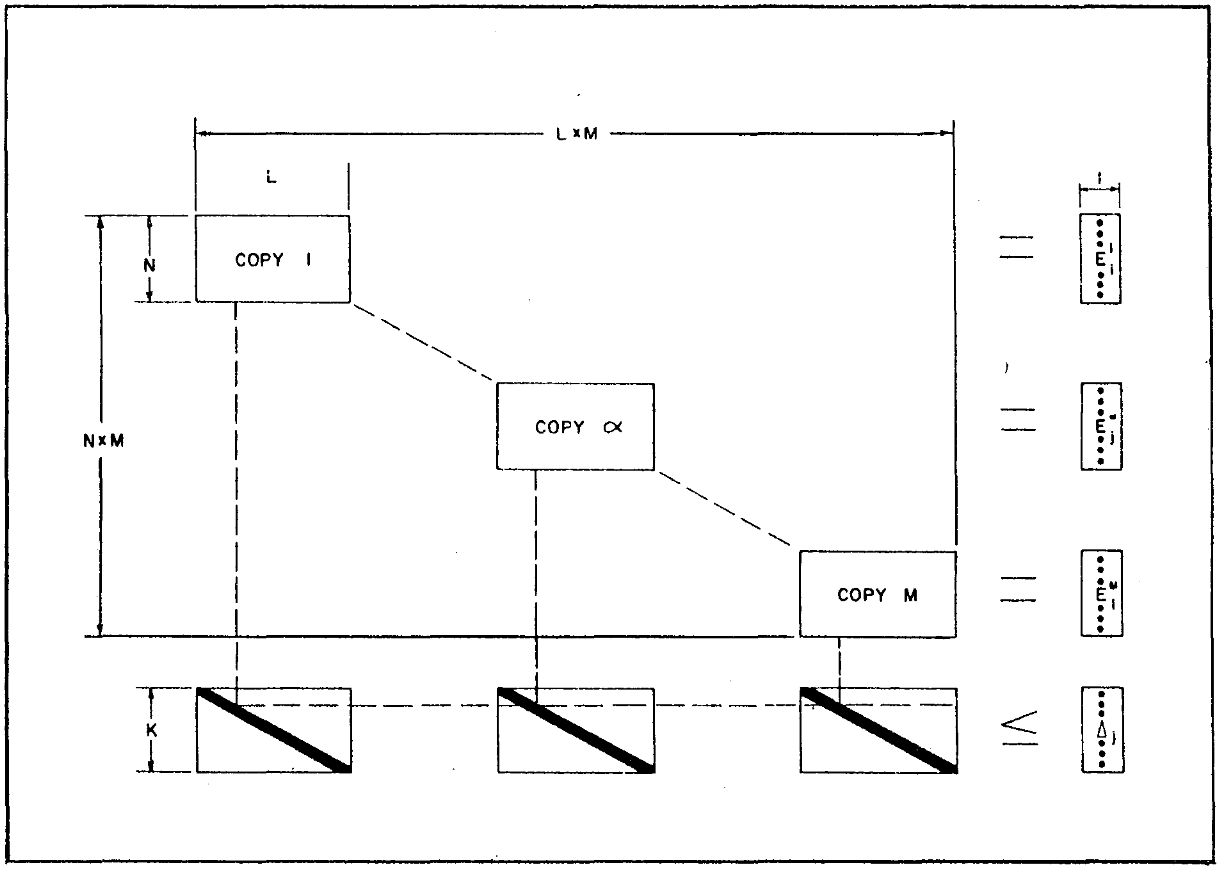
A problem of model size is encountered at once when the distribution problem is considered as a general linear programming problem. This can be readily illustrated by considering a sample network problem. Assume (see Figure 5) that the number of nodes (N) = 100, the number of links (L) = 300 and that 20 copies (M) are required to describe the origin-destination requirements. If 30 links are capacitated, then the total size of the matrix involved is:

$$(N \times M + K) \text{ by } L \times M$$

or

$$(100 \times 20 + 30) \text{ by } 300 \times 30 = 2030 \text{ by } 9000.$$

The magnitude of these numbers readily indicates that it is not practical to treat the problem as a general linear programming problem. In order to reduce the size of the problem from a computational standpoint, Charnes and Cooper (11) have devised a special method which is



M = NUMBER OF COPIES
N = NUMBER OF NODES

L = NUMBER OF LINKS
K = NUMBER OF CAPACITATED LINKS

PROBLEM STRUCTURE
TRAFFIC NETWORK
LINEAR PROGRAMMING PROBLEM

Figure 5

termed a multi-copy "mixing" technique. This technique provides a feasible approach to the solution of the traffic network linear programming problem.

By using a change of variable, the unknown in the linear programming problem is changed from the amount of traffic on a given branch for a specific copy (x_j^α) to the percent of a given extreme point solution. Using vector and matrix notation, one may formulate the general linear programming problem as follows:

$$\text{Minimize} \qquad \sum_{\alpha=1}^{m} C^{\alpha\,T} \lambda^{\alpha} \tag{8}$$

$$\text{subject to} \qquad A^{\alpha} \lambda^{\alpha} = b^{\alpha} \tag{9}$$

$$\sum_{\alpha=1}^{m} K^{\alpha} \lambda^{\alpha} \leq d \tag{10}$$

$$\lambda^{\alpha} \geq 0 \qquad (\alpha = 1,2,\ldots m)$$

where C^α = Cost vector for copy α. Individual element C_j^α is the cost on link j for copy α.

λ^α = Solution vector for copy α. Individual element λ_j^α is the number of vehicles assigned to link j on copy α.

A^α = Matrix of incidence numbers for copy α.

b^α = Vector of node influxes or effluxes for copy α. Individual element b_j^α is the influx or efflux at node j on copy α.

K^α = Matrix of structural coefficients (1 or 0) which specifies the capacitated links on copy α.

d = Vector of capacity constraints. Individual element d_j is the limiting capacity on link j.

Note - Vectors are considered as column vectors unless otherwise indicated. For example $C^{\alpha\, T}$ is a row vector obtained by the transpose of C^{α}.

Thus λ^{α} represents a solution to a given copy α and satisfies the constraints

$$A^{\alpha}\ \lambda^{\alpha} = b^{\alpha}$$

$$\lambda^{\alpha} \geq 0.$$

Therefore, λ^{α} is one element of the total solution vector λ where

$$\lambda^{T} = (\lambda^{1T}, \lambda^{2T}, \cdots \lambda^{\alpha\, T} \cdots \lambda^{mT}). \qquad (11)$$

Assuming that the λ^{α} (individual copy solutions) form a bounded non-empty convex set, one can then express λ^{α} as a convex combination of extreme points as follows:

$$\lambda^{\alpha} = \sum_{\beta=1}^{n(\alpha)} e^{\alpha\beta} \mu_{\alpha\beta} \qquad (12)$$

with the condition

$$\sum_{\beta=1}^{n(\alpha)} \mu_{\alpha\beta} = 1 \qquad (13)$$

$$\mu_{\alpha\beta} \geq 0 \qquad (14)$$

where

$\rho^{\alpha\beta}$ = The β-th extreme point solution on copy α

$\mu_{\alpha\beta}$ = Percentage of the β-th extreme point solution on copy α.

It can then be seen that the total solution λ consists of convex combinations or "mixtures" of individual extreme point copy solutions.

The problem

$$\text{Minimize} \quad \sum_{\alpha=1}^{m} c^{\alpha T} \lambda^{\alpha} \tag{15}$$

$$\text{subject to} \quad A^{\alpha} \lambda^{\alpha} = b^{\alpha} \tag{16}$$

$$\sum_{\alpha=1}^{m} K^{\alpha} \lambda^{\alpha} = d \tag{17}$$

$$\lambda^{\alpha} \geq 0 \tag{18}$$

can be reformulated as a "mixing" problem as follows:

$$\text{Minimize} \quad \sum_{\alpha=1}^{M} \sum_{\beta=1}^{n(\alpha)} c^{\alpha T} \rho^{\alpha\beta} \mu_{\alpha\beta} \tag{19}$$

$$\text{subject to} \quad \sum_{\alpha=1}^{m} \sum_{\beta=1}^{n(\alpha)} K^{\alpha} \rho^{\alpha\beta} \mu_{\alpha\beta} = d \tag{20}$$

$$\sum_{\beta=1}^{n(\alpha)} \mu_{\alpha\beta} = 1 \tag{21}$$

$$\mu_{\alpha\beta} \geq 0. \tag{22}$$

In the preceding formulation, $\sum_{\beta=1}^{n(\alpha)} \rho^{\alpha\beta}\mu_{\alpha\beta}$ has been substituted for λ^{α} and the conditions

$$\sum_{\beta=1}^{n(\alpha)} \mu_{\alpha\beta} = 1 \tag{23}$$

$$\mu_{\alpha\beta} \geq 0 \tag{24}$$

have been incorporated as part of the constraint equations. The constraints

$$A^{\alpha} \lambda^{\alpha} = b^{\alpha} \tag{25}$$

$$\lambda^{\alpha} \geq 0 \tag{26}$$

are redundant and omitted since $\rho^{\alpha\beta}$ is a particular λ^{α} which satisfies these conditions.

The variable for the mixing problem is $\mu_{\alpha\beta}$ with the cost

$$\bar{C}_{\alpha\beta} = C^{\alpha T} \rho^{\alpha\beta}. \tag{27}$$

The vectors from the constraint equations will be made up of

$$L^{\alpha\beta} = K^{\alpha\beta} \rho^{\alpha\beta} \tag{28}$$

where

$L^{\alpha\beta}$ = Vector of traffic assignments on capacitated links for copy α and the β-th extreme point solution. An individual element $L_k^{\alpha\beta}$ is the amount of traffic on capacitated link k on copy α for the β-th extreme point solution and the unity element from

$$\sum_{\beta=1}^{n(\alpha)} \mu_{\alpha\beta} = 1. \qquad (29)$$

Solution Technique

The initial tableau for the problem can be set up by obtaining an extreme point solution from each copy ($\rho^{\alpha\beta}$) and by adding the necessary slack or artificial vectors. An example of the initial tableau is shown in Figure 6. This tableau contains the basis vectors for a feasible solution to the problem.

If the inverse of the basis is obtained (B^{-1}), the initial basic feasible solution (P_0') is found as follows:

$$P(0)' = B^{-1} P(0). \qquad (30)$$

The mixing problem is now in the form of a simplex tableau and optimality checks can be obtained by considering individual "$Z_j - \bar{C}_j$" expressions. If a modified simplex method as developed by Charnes and Lemke (12) is used to progress toward an optimum solution, then an individual vector (P_j) to be checked as "come-in" vector may be expressed as follows:

$$\begin{bmatrix} P_j \\ Z_j - \bar{C}_j \end{bmatrix} = \begin{bmatrix} B^{-1} P_j \\ \omega P_j - \bar{C}_j \end{bmatrix} \qquad (31)$$

The vector ω is an "evaluation vector" and is computed as follows:

COST VECTOR	$\bar{C}_{11}$	$\bar{C}_{21}$	$\bar{C}_{31}$	•	•	•	$\bar{C}_{M1}$	0	∞	•	•	•	0	
SOLUTION VECTORS	P_{11}	P_{21}	P_{31}	•	•	•	P_{M1}	S_1	A_1	•	•	•	S_K	$P_{(0)}$
CAPACITATED LINK 1	V_{11}	V_{21}	V_{31}	•	•	•	V_{M1}	1						Δ_1
CAPACITATED LINK 2	V_{12}	V_{22}	V_{32}	•	•	•	V_{M2}		1					Δ_2
•	•	•	•				•			•				•
•	•	•	•				•				•			•
•	•	•	•				•					•		•
CAPACITATED LINK K	V_{1K}	V_{2K}	V_{3K}	•	•	•	V_{MK}						1	Δ_K
COPY 1	1													1
COPY 2		1												1
COPY 3			1											1
•				•										•
•					•									•
•						•								•
COPY M							1							1

$\bar{C}_{M1}$ — COST OF FIRST SOLUTION ON COPY M

P_{M1} — VECTOR FOR SOLUTION 1 ON COPY M

S_1 — SLACK VECTOR

A_1 — ARTIFICIAL VECTOR

V_{ij} — VOLUME ON LINK (j) FROM COPY (i)

INITIAL TABLEAU MULTI-COPY MODEL

FIGURE 6

$$\omega^T = \bar{C}\, B^{-1} \tag{32}$$

where

$\bar{C}$ = Total cost vector. An individual element is $\bar{C}_{\alpha,\beta}$ as previously defined.

The ω vector will be made up as follows:

$$\omega^T = (U^T, \sigma_1, \ldots \sigma_\alpha \ldots \sigma_m) \tag{33}$$

where

U^T = Vector of individual ω elements associated with the constraint

$$\sum \sum K^{\alpha,\beta} \varrho^{\alpha\beta} \mu_{\alpha,\beta} = d \tag{34}$$

and σ_α = Individual ω element associated with the constraint

$$\sum \mu_{\alpha,\beta} = 1. \tag{35}$$

If an individual vector is checked as a "come-in" vector by considering its "$Z_j - \bar{C}_j$" value, then

$$Z_j - \bar{C}_j = \omega^T P_j - \bar{C}_j. \tag{36}$$

Letting

$$j = \alpha,\beta \tag{37}$$

and since

$$P_{\alpha,\beta} = [L^{\alpha\beta}, 0, \ldots 0, 1, 0, \ldots 0] \tag{38}$$

and

$$\omega^T P_{\alpha\beta} = U^T L^{\alpha\beta} + \sigma_\alpha \tag{39}$$

then

$$Z_{\alpha\beta} - \bar{C}_{\alpha\beta} = U^T L^{\alpha\beta} + \sigma_\alpha - \bar{C}_{\alpha\beta} \, . \tag{40}$$

Rearranging (40) and using the relationships

$$L^{\alpha\beta} = K^{\alpha\beta} \rho^{\alpha\beta} \tag{41}$$

and

$$\bar{C}_{\alpha\beta} = C^{\alpha T} \rho^{\alpha\beta} \tag{42}$$

then

$$Z_{\alpha\beta} - \bar{C}_{\alpha\beta} = (U^T K^\alpha - C^{\alpha T}) \rho^{\alpha\beta} + \sigma_\alpha \, . \tag{43}$$

For optimality, it is necessary and sufficient that for all α

$$\text{Max over } \beta \left\{ (U^T K^\alpha - C^{\alpha T}) \rho^{\alpha\beta} + \sigma_\alpha \right\} \leq 0. \tag{44}$$

For non-optimality and the indication of a vector ($P_{\alpha\beta}$) as a "come-in" vector, it is necessary and sufficient that for some α

$$\text{Max over } \beta \left\{ (U^T K^\alpha - C^{\alpha T}) \rho^{\alpha\beta} + \sigma_\alpha \right\} > 0. \tag{45}$$

Thus at each stage, the value for

$$\text{Max over } \beta \left\{ (U^T K^\alpha - C^{\alpha T}) \rho^{\alpha\beta} + \sigma_\alpha \right\} \tag{46}$$

must be obtained.

Since σ_α is not a function of β, the problem of finding

$$\text{Max over } \beta \quad \left\{ (U^T K^\alpha - C^{\alpha T}) \rho^{\alpha\beta} + \sigma_\alpha \right\} \tag{47}$$

for a particular copy (α) is equivalent to solving the following:

$$\text{Max} \quad (U^T K^\alpha - C^{\alpha T}) \rho^{\alpha\beta} \tag{48}$$

$$\text{subject to} \quad A^\alpha \rho^{\alpha\beta} = b^\alpha \tag{49}$$

$$\rho^{\alpha\beta} \geq 0 \tag{50}$$

or

$$\text{Min} \quad (C^{\alpha T} - U^T K^\alpha) \rho^{\alpha\beta} \tag{51}$$

$$\text{subject to} \quad A^\alpha \rho^{\alpha\beta} = b^\alpha \tag{52}$$

$$\rho^{\alpha\beta} \geq 0 \,. \tag{53}$$

The solution to the above formulation, however, is a minimum-path solution to the copy network with the link costs given by

$$\text{Link Cost Vector} \quad = (C^{\alpha T} - U^T K^\alpha). \tag{54}$$

Thus obtaining a minimum-path solution for a particular copy supplies a $\rho^{\alpha\beta}$ and a value for $Z_{\alpha\beta} - \bar{C}_{\alpha\beta}$. If this value is greater than zero, then the column vector for this extreme point may be used as a "come-in" vector. If the value is less than or equal to zero, another copy solution may be considered. If a vector is brought into the basis, then new values for ω^T and the "dummy" costs $(C^{\alpha T} - U^T K^\alpha)$ may be obtained. Using these new costs, a new $\rho^{\alpha\beta}$ is generated and the corresponding vector may be checked for "come-in".

Thus the large over-all problem may be broken down into two smaller problems for computational efficiency. These are:

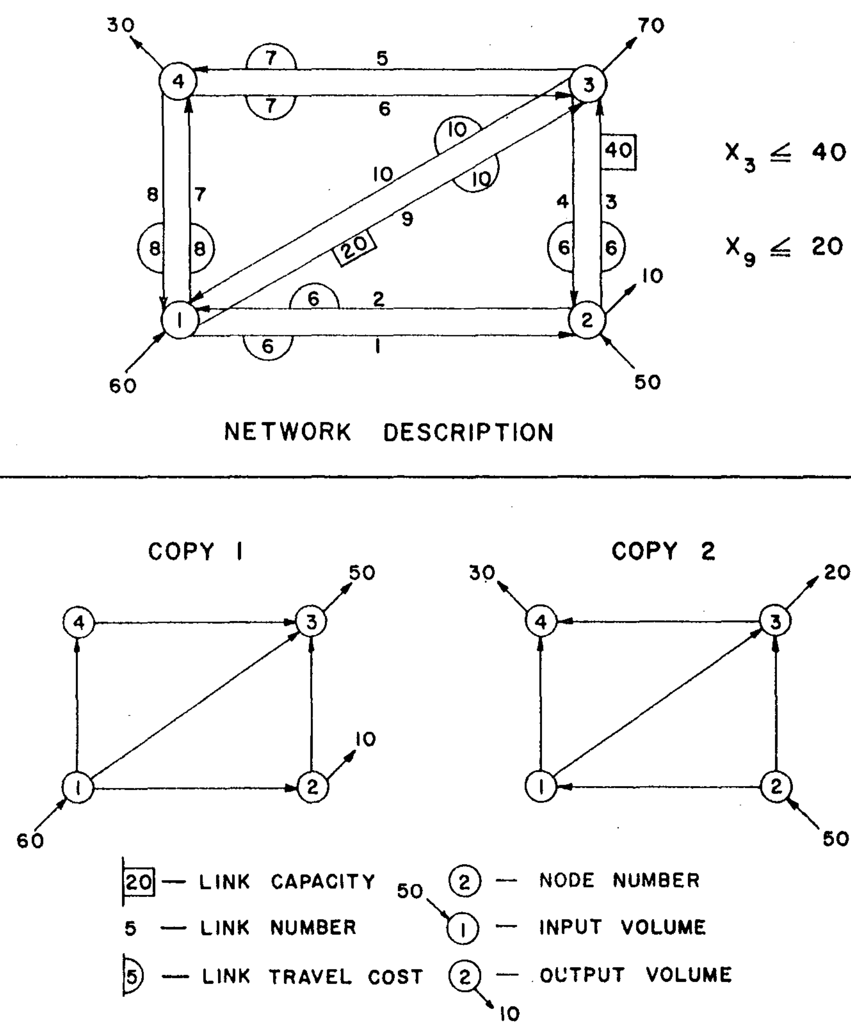
1. The solution of a minimum-path problem for a given copy network.
2. The modified simplex procedure.

One other condition that must be considered is the occurrence of a positive element in the U^T vector. This condition indicates that a "slack vector" (a unity vector with a one in the position corresponding to the positive U^T element) would improve the solution and such a vector must be entered at this time.

The efficiency of the "mixing" technique is illustrated by considering the problem discussed on page 15. Whereas the solution of this example as a general linear programming problem would require working with a 2030 by 9000 matrix, the "mixing" model technique would require only a 50th order "working" matrix. The order of the mixing model matrix is the number of copies plus the number of capacitated links.

<u>Example Problem</u>

In order to illustrate the model and to demonstrate the application of the mathematics previously developed, the step-by-step solution of a small network distribution problem will be presented and discussed. The simple four-node network with directional links and traffic requirements shown in Figure 7 will be used for analysis. The problem consists of determining the optimum routing of traffic over this network with



NETWORK DESCRIPTION AND TRAFFIC DATA EXAMPLE PROBLEM

FIGURE 7

capacity restrictions on links 3 and 9.

A step-by-step solution of this problem is as follows:

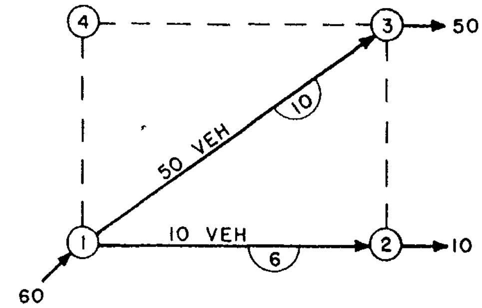
Step 1 - Determine minimum path solutions for each copy using actual link costs. Determine solution costs and solution vectors. Set up initial basis and P(0) vector. See Figure 8. An extremely high cost denoted by the letter M is assigned to the artificial vectors.

Step 2 - Find inverse of initial basis and compute ω^T and P_o'. Set up this information in a tableau as shown in Figure 9.

Step 3 - Compute new "dummy costs" for the network and find new minimum path solution. (Figure 10). Note that the overload of links 3 and 9 results in these links receiving high travel costs and that solution ρ^{12} does not utilize these links. The vector P_{12} is formed and checked for "come-in." The large "$Z_j - C_j$" indicates it would improve the solution and a standard simplex computation for θ indicates its "come-in" point.

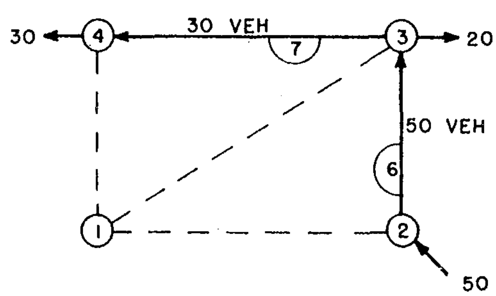
Step 4 - Vector P_{12} is brought into the solution and the tableau shown in Figure 11 results. Dummy costs are computed and the solution ρ^{13} is obtained with the corresponding vector P_{13}. The "$Z_j - C_j$" for this vector is negative, which indicates it will not improve the solution and therefore copy 2 is now considered. Vector P_{22} has a large positive "$Z_j - C_j$" and will be brought into the solution.

FIRST MINIMUM PATH — COPY 1 — ρ_{11}



COST ρ_{11} = 50 (10) + 10 (6) = 560

FIRST MINIMUM PATH — COPY 2 — ρ_{21}



COST ρ_{21} = 30 (7) + 50 (6) = 510

INITIAL BASIS AND P(o) VECTOR

P_{11}	P_{21}	A_1	A_2	
560	510	M	M	P (o)
0	50	-1	0	40
50	0	0	-1	20
1	0	0	0	1
0	1	0	0	1

INITIAL BASIS EXAMPLE PROBLEM

FIGURE 8

INVERSE OF INITIAL BASIS

$B^{-1'} =$

0	0	1	0
0	0	0	1
−1	0	0	50
0	−1	50	0

$$\omega \text{ VECTOR} = \bar{C}\,B^{-1'} = [\,560, 510, M, M,\,] \begin{bmatrix} B^{-1} \end{bmatrix}$$

$$= [\,-M, -M, 560 + 50M, 510 + 50M\,]$$

$$P(0') = B^{-1}\,P(0) = \begin{bmatrix} B^{-1} \end{bmatrix} \begin{bmatrix} 40 \\ 20 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 10 \\ 30 \end{bmatrix}$$

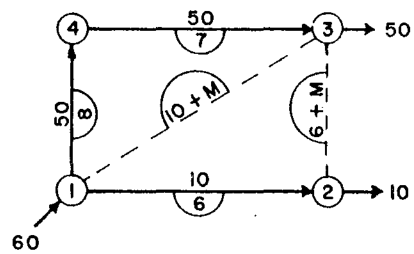
	P_{11}	P_{21}	A_1	A_2	$P(0')$	BASIS	
	0	0	1	0	1	P_{11}	
	0	0	0	1	1	P_{21}	
	−1	0	0	50	10	A_1	
	0	−1	50	0	30	A_2	
ω^T →	−M	−M	560 + 50M	510 + 50M	1070 + 40M		← CURRENT COST

INVERSE OF INITIAL BASIS EXAMPLE PROBLEM

FIGURE 9

FOR LINK 3 DUMMY COST = 6 − (−M 6 + M

FOR LINK 9 DUMMY COST = 10 − (−M) = 10 + M



ρ_{12}

$P_{12}^T = [0, 0, 1, 0]$

COST = 50 (8) + 50 (7) + 10 (6) = 810

$\omega P_{12} - \bar{C} = 560 + 50M - 810 = -250 + 50M$

				$P'(0)$	BASIS	P_{12}	P'_{12}	
0	0	1	0	1	P_{11}	0	1	$\theta = 1/1 = 1$
0	0	0	1	1	P_{12}	0	0	
−1	0	0	50	10	A_1	1	0	
0	−1	50	0	30	A_2	0	50	$\theta = 30/50 = 3/5$ ←
−M	−M	560 + 50M	510 + 50M	1070 + 40M		−250 + 50M		← $Z_j - \bar{C}_j$

INITIAL TABLEAU EXAMPLE PROBLEM

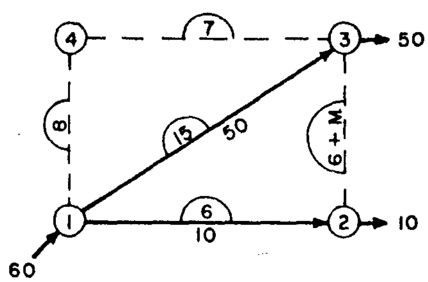
FIGURE 10

				$P(0)'$	BASIS	P_{13}	P_{22}	P_{22}'
0	1/50	0	0	2/5	P_{11}	0	0	2/5
0	0	0	1	1	P_{21}	50	20	1
-1	0	0	50	10	A_1	1	0	50
0	-1/50	1	0	3/5	P_{12}	0	1	-2/5
-M	-5	810	510 + 50 M	1220 + 10 M		-75	-330 + 50 M	

$\theta = \frac{2}{5} \Big/ \frac{2}{5} = 1$

$\theta = 1/1 = 1$

$\theta = 10/50 = 1/5$ ←

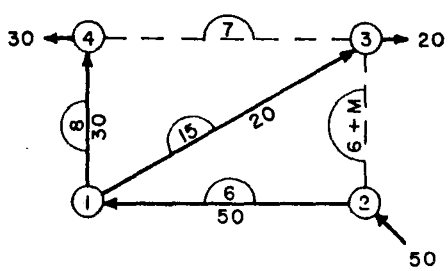


$\underline{\rho_{13}}$

$P_{13}^T = [0, 50, 1, 0]$

$\bar{C}_{13} = 15(50) + 6(10) = 810$

$\omega P_{13} - \bar{C}_{13} = -75 + 810 - 810 = -75$



$\underline{\rho_{22}}$

$P_{22}^T = [0, 20, 0, 1]$

$\bar{C}_{22} = 50(6) + 30(8) + 10(20) = 740$

$\omega P_{22} - \bar{C}_{22} = -100 + 510 + 50M - 740$
$= -330 + 50M$

INVERSE OF SECOND BASIS EXAMPLE PROBLEM

FIGURE 11

Step 5 - Vector P_{22} is brought into the solution and the tableau shown in Figure 12 is obtained. Solution ρ^{23} is developed and the corresponding vector P_{23} has a positive "$Z_j - C_j$".

Step 6 - Vector P_{23} is brought into the solution and the tableau shown in Figure 13 obtained. Solution ρ^{24} is determined and vector P_{24} is found to be an alternate solution. Copy 1 is considered again and vector P_{14} is found to have a positive "$Z_j - C_j$."

Step 7 - Vector P_{14} is brought into the solution and the tableau shown in Figure 14 is obtained. Vector P_{15} is found to be an alternate solution as is vector P_{25}. Thus an optimum solution to the problem has been reached.

The final solution to the example problem is given by the P_o' vector in the tableau of Figure 14 and is as follows:

$$10/25\,\rho^{11} + 2/3\,\rho^{14} + \rho^{23} + 1/5\,\rho^{12}$$

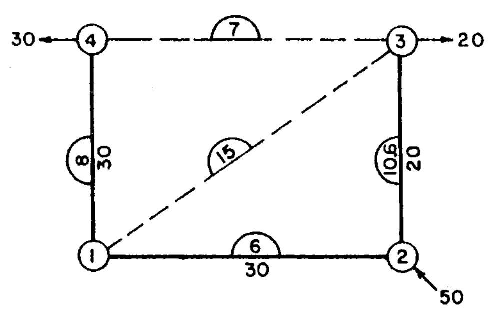
Taking the indicated percentages of the copy solutions in the final basis and combining them yields the final optimum solution shown in Figure 15.

				$P_{(0)}'$	BASIS	P_{23}	P_{23}'
$2/250$	$1/50$	0	$-2/5$	$8/25$	P_{11}	20	$-6/25$
$1/50$	0	0	0	$4/5$	P_{21}	0	$2/5$
$-1/50$	0	0	1	$1/5$	P_{22}	0	$3/5$
$-2/250$	$-1/50$	1	$2/5$	$17/25$	P_{12}	1	$6/25$
-4.6	-5	810	740	1266		108	

$\theta = \frac{4/5}{2/5} = 2$

$\theta = \frac{1/5}{3/5} = 1/3$ ←

$\theta = \frac{17/25}{6/25} = 17/6$



ρ_{23}

$P_{23}^T = [\,20, 0, 0, 1\,]$

$\bar{C}_{23} = 8(30) + 6(30) + 20(6) = 540$

$\omega P_{23} - \bar{C}_{23} = -4.6(20) + 740 - 540 = 108$

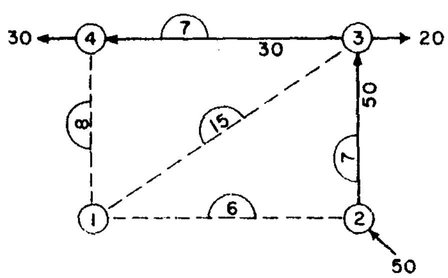
INVERSE OF THIRD BASIS EXAMPLE PROBLEM

FIGURE 12

				$P'(0)$	BASIS	P_{24}	P_{14}	P'_{14}
0	$1/50$	0	0	$10/25$	P_{11}	50	50	0
$1/30$	0	0	$-2/3$	$10/15$	P_{21}	0	0	$5/3$
$-1/30$	0	0	$5/3$	$1/3$	P_{23}	0	1	$-5/3$
0	$-1/50$	1	0	$15/25$	P_{12}	1	0	1
−1	−5	810	560	1230		0		

$\theta = \frac{10/15}{5/3} = 2/5$ ←

$\theta = \frac{15/25}{1} = 3/5$



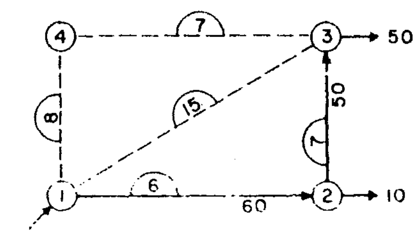
ALTERNATE SOLUTION

ρ_{24}

$P_{24}^T = [\,50, 0, 0, 1\,]$

$\bar{C}_{24} = 7(30) + 6(50) = 510$

$\omega P_{24} - \bar{C}_{24} = -50 + 560 - 510 = 0$



ρ_{14}

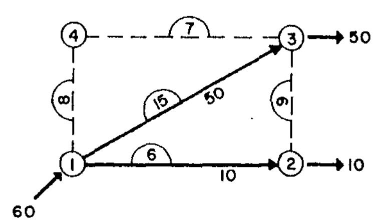
$P_{14}^T = [\,50, 0, 1, 0\,]$

$\bar{C}_{14} = 6(60) + 6(50) = 660$

$\omega P_{14} - \bar{C}_{14} = -50 + 810 - 660 = +100$

INVERSE OF FOURTH BASIS
EXAMPLE PROBLEM
FIGURE 13

				$P(0')$	BASIS	P_{15}	P_{25}
0	1/50	0	0	10/25	P_{11}	0	20
1/50	0	0	-2/5	2/5	P_{14}	50	0
0	0	0	1	1	P_{23}	1	0
-1/50	-1/50	1	2/5	1/5	P_{12}	0	1
-3	-5	810	600	1190		0	0



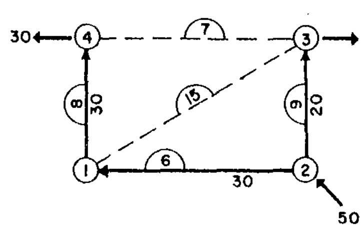
$\underline{P_{15}}$

$P_{15}^{T} = [0,50,1,0]$

$\bar{C}_{15} = 10(50) + 6(10) = 560$

$\omega P_{15} - \bar{C}_{15} = -250 + 810 - 560 = 0$

ALTERNATE SOLUTION



$\underline{P_{25}}$

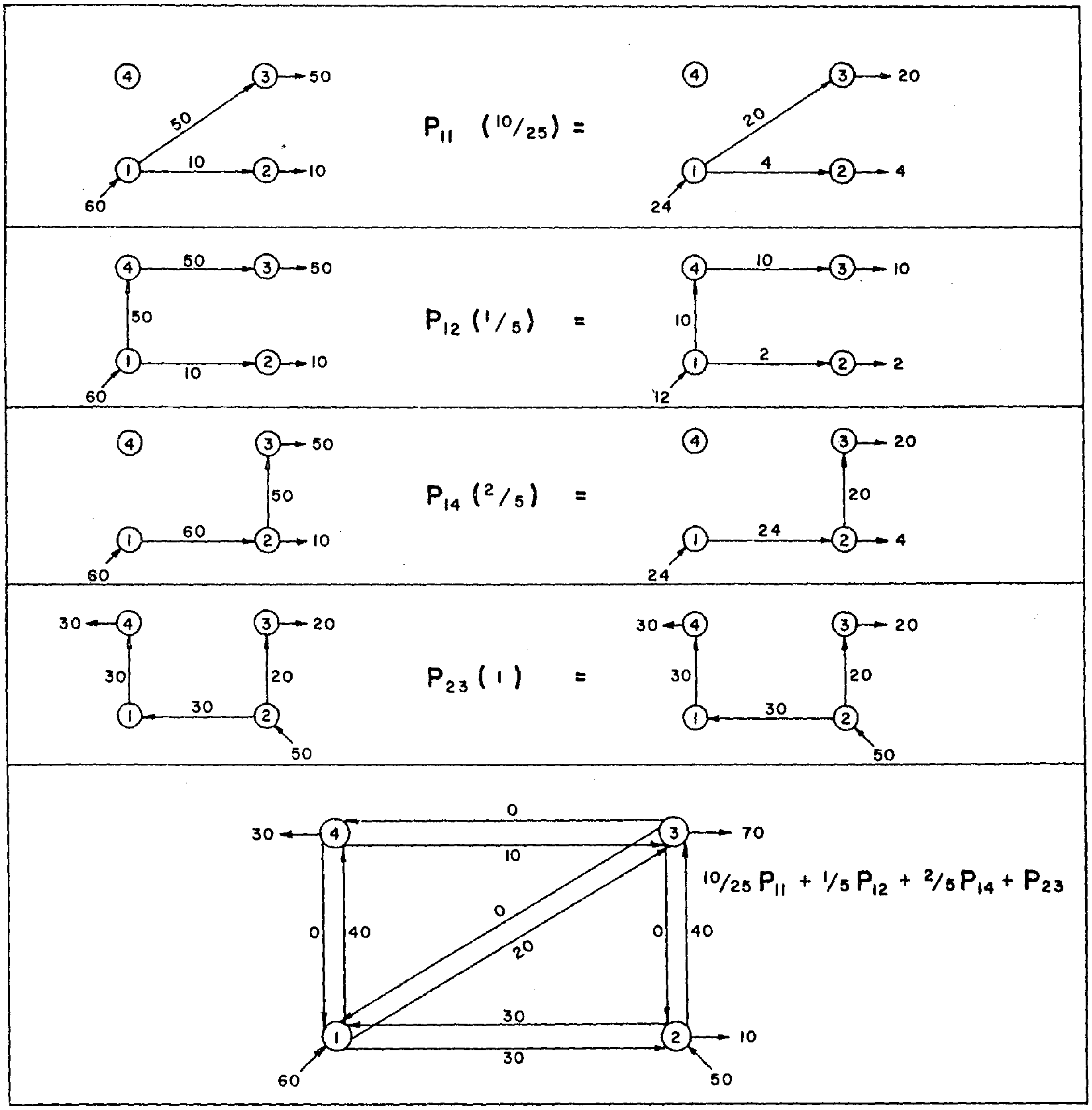
$P_{25}^{T} = [20,0,0,1]$

$\bar{C}_{25} = 20(6) + 6(30) + 8(30) = 540$

$\omega P_{25} - \bar{C}_{25} = -60 + 600 - 540 = 0$

ALTERNATE SOLUTION

INVERSE OF FINAL BASIS
EXAMPLE PROBLEM
FIGURE 14



FINAL SOLUTION
EXAMPLE PROBLEM
FIGURE 15

COMPUTER PROGRAM DEVELOPMENT

In order to study the optimum distribution of traffic using a linear programming model, it was necessary to develop a computer program which would perform the arithmetic operations of the problem. A computer program to perform this technique was coded by staff members of the Data Processing Center of Texas A&M University. This coding was written in Fortran II with the exception of one subroutine which was written in FAP (Fortran Assembly Program). The logic, testing and operation of this program will be discussed in the following sections of this chapter.

Program Logic

The basic logic of the program is illustrated in the block diagram shown in Figure 16. The program utilizes three basic operations in proceeding toward an optimum solution. These are as follows:

(1) Computation of a minimum time path through the network.

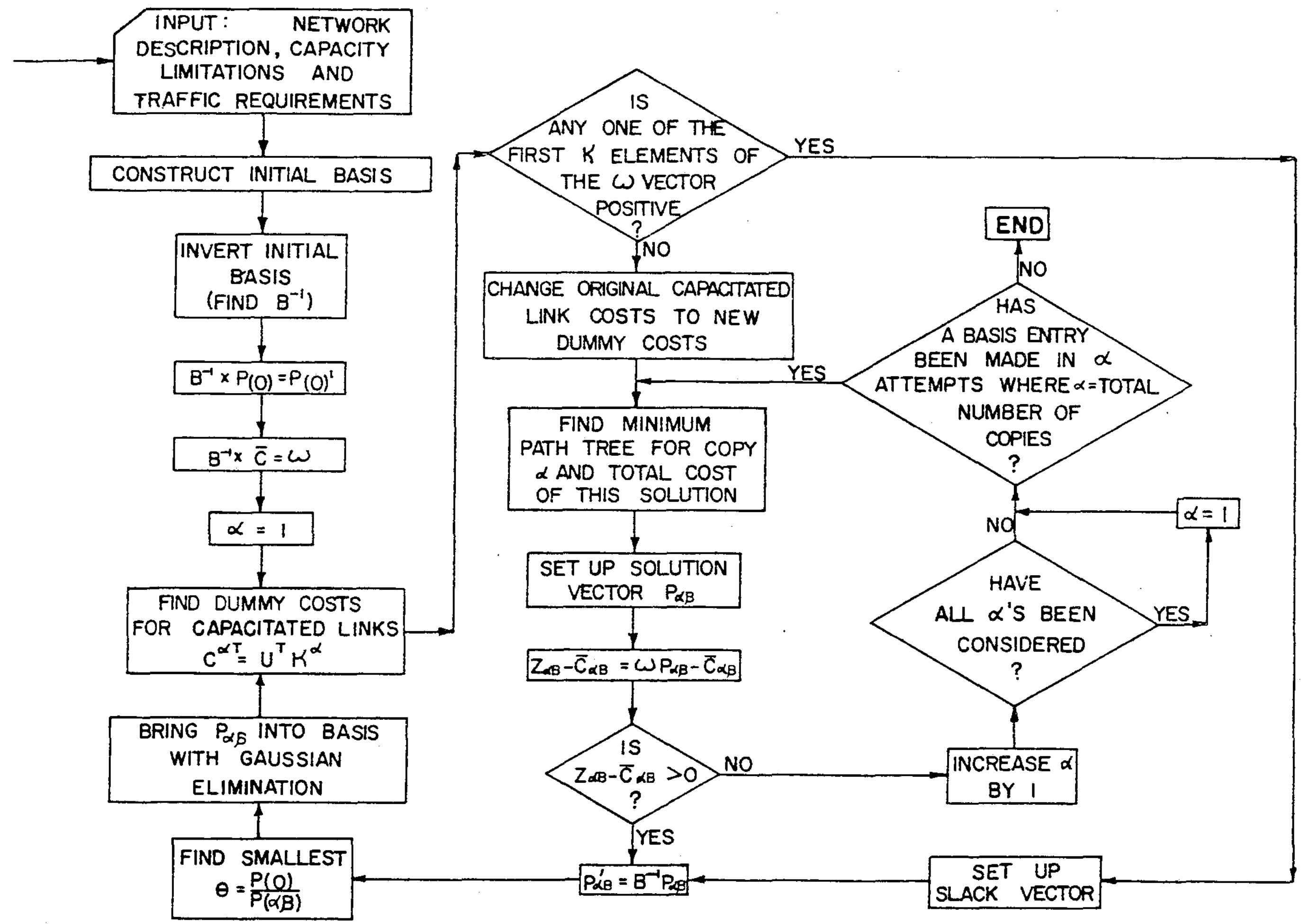
(2) Modified simplex technique of checking optimality at each stage.

(3) Gaussian elimination technique of bringing new vectors into the basis.

These three basic steps are repeated until an optimum solution is reached on all copies.

Initial Program

The initial program was not designed to handle a large network in



PROGRAM LOGIC

FIGURE 16

order to insure that all main functions of the program could be performed within core storage of the I.B.M. 709, a stored program digital computer with 32,768 words of core storage. The following size limitations were established for the initial program:

1. The main matrix was limited to an order of 20, which is determined by the sum of the number of copies plus the number of capacitated links.
2. The number of copies was restricted to 15.
3. The network was limited to a total of 400 links.
4. The number of output nodes per copy was limited to 15.

These limitations did not seriously restrict the size of test problem that could be run and avoided the possibility of exceeding core storage.

The initial computer program had 12 subroutines which performed the following functions:

<u>SUBROUTINE START</u> - Subroutine to initialize all arrays and read all input data. Input data consist of a complete network description, individual copy flow data, capacitated link numbers and their capacities. See Figure 17 in Appendix A for flow diagram.

<u>SUBROUTINE TREE</u> - Subroutine to assign traffic on one or all copies. It is called initially with key of 1 and then finds minimum paths and makes traffic assignments on all copies. When called with a key of 2, the subroutine finds a minimum path for a specific copy and assigns traffic to this path. A flow diagram for this subroutine is shown in Figure 18 in Appendix A.

SUBROUTINE PATH - This subroutine finds the minimum path route through a network from a given input node to all other nodes. The technique used in this subroutine is a modification of the procedure first described by Moore (32). Subroutine PATH is called by TREE in performing its functions. A flow diagram of this subroutine is shown in Figure 19 in Appendix A.

SUBROUTINE KOST - This subroutine sets up the initial main matrix, the true cost vector and the P(o) vector. It also computes the indices which specify the (+) or (-) ones to be placed in the initial matrix. See Figure 20 in Appendix A.

SUBROUTINE MULT - Subroutine to accomplish the various matrix and vector multiplications. See Figure 21 in Appendix A.

SUBROUTINE CHKINT - Subroutine to compute the P(o) Prime and Omega vector for a copy solution being checked for entry into the basis. This subroutine utilizes subroutine MULT in this computation. After computation of the Omega vector, this vector is checked for a positive element and the need to enter a slack vector into the basis. See Figure 22 in Appendix A for a flow diagram.

SUBROUTINE LOGIC - Subroutine to compute the "dummy costs" for the capacitated links of the network and to transfer these costs to the corresponding links. See Figure 23 of Appendix A.

SUBROUTINE NEWVAL - Subroutine to utilize the "dummy costs" found by subroutine LOGIC in computing a new minimum path solution for a specific copy. After finding the minimum path and assigning traffic to it, the solution cost, the solution vector P(j) and Z(j) are computed. See Figure 24 of Appendix A.

SUBROUTINE OTHVAL - If a check of Z(j) - C(j) in the main program indicates a "come-in" vector, this subroutine finds the entry point of this vector into the basis by comparing elements of a computed P(j) prime vector and the P(o) prime vector to determine a minimum Theta value. It then brings the new vector into the basis by a Gaussian Elimination technique and computes a new cost vector.

Alternately, subroutine OTHVAL is called when the $Z_j - C_j$ check indicates an alternate solution. This subroutine then finds the entry point into the basis for the alternate solution. See Figure 25 of Appendix A.

SUBROUTINE SLACK - Whenever a positive element is detected in the Omega vector by subroutine CHKINT, this subroutine is called to bring a slack vector into the basis. The slack vector comes into a position corresponding to the positive Omega element. See Figure 26 of Appendix A.

SUBROUTINE SHELL - This subroutine computes the convex combinations (or percentages) of all copy solutions in the basis. It then combines the traffic volumes from the various copies for final output. See Figure 27 of Appendix A.

SUBROUTINE PRNT - This subroutine outputs the answers in their final form. See Figure 28 in Appendix A.

A flow diagram of the main control program illustrating the calling of the various subroutines is shown in Figure 29 in Appendix A.

Program Tests

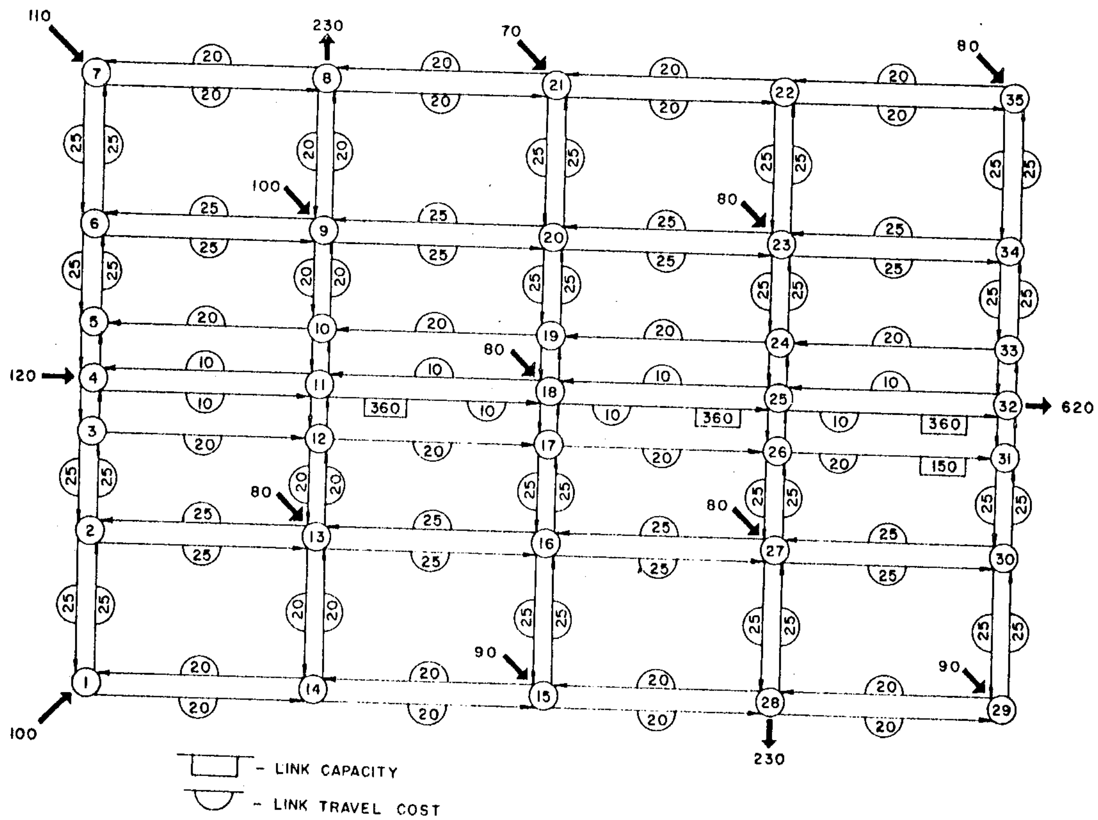
Two basic test problems were utilized in the initial checkout of the computer program. The first problem run (Test Problem 1) was the network problem previously illustrated in Figure 7. Test Problem 2 consisted of a 12-copy, 35-node network problem with 4 capacitated links. The network description and data on traffic inputs and outputs are shown in Figure 30.

These problems offered the advantage of a known solution in each case. Test Problem 1 was worked out manually as an example and Test Problem 2 was solved manually in a paper by Pinnell and Satterly. These problems provided good checkout material and after some relatively minor modifications to the original program it was possible to duplicate the manual results obtained for both of the problems.

After initial checkout of the program, a second phase of testing was initiated to ascertain the ability of the program to handle a wide range of input data over networks of various descriptions and to study the optimum distribution of traffic. The work of this phase was accomplished by the use of six additional problems numbered for reference as Test Problems 3, 4, 5, 6, 7, and 8.

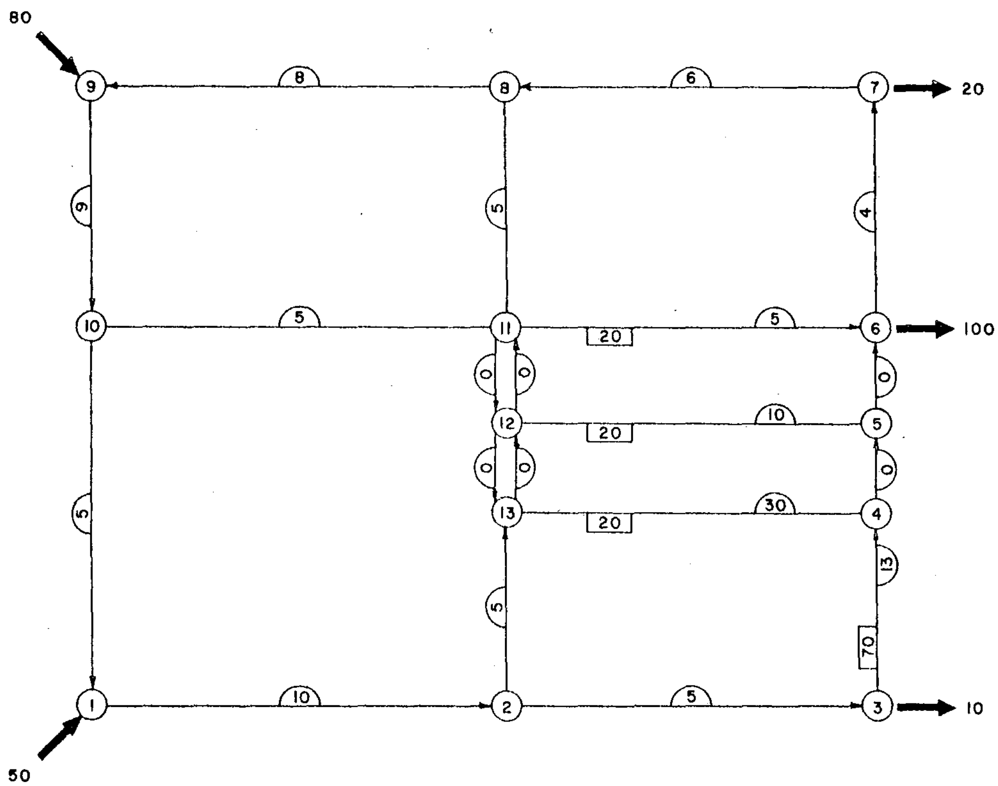
Test Problem 3 - This problem was devised principally to test the ability of the program to handle the case where a network link was capacited at several different levels. A two-copy problem with four capacitated links was developed for this purpose. Figure 31 illustrates the network description and traffic requirements for this problem.

Capacitating a link at several levels is necessary to provide a piecewise linear approximation to a travel time. This curve as



NETWORK DESCRIPTION AND TRAFFIC DATA
TEST PROBLEM 2

FIGURE 30



NETWORK DESCRIPTION AND TRAFFIC DATA
TEST PROBLEM 3

FIGURE 31

developed for freeways resembles the one shown in Figure 32. The piecewise linear approximation to this nonlinear curve is slightly in error but improves in accuracy as the number of linear approximations is increased.

As one level of capacity is satisfied, each vehicle utilizing the next level of capacity must be given a time penalty large enough to represent its effect on the entire traffic stream. As an explanation of this consider the following example:

Assume the following conditions exist:

For volumes from 0 to 100 vehicles per hour ---

Travel time is 5 minutes per vehicle

For volumes between 100 and 300 vehicles per hour ---

Travel time is 8 minutes per vehicle

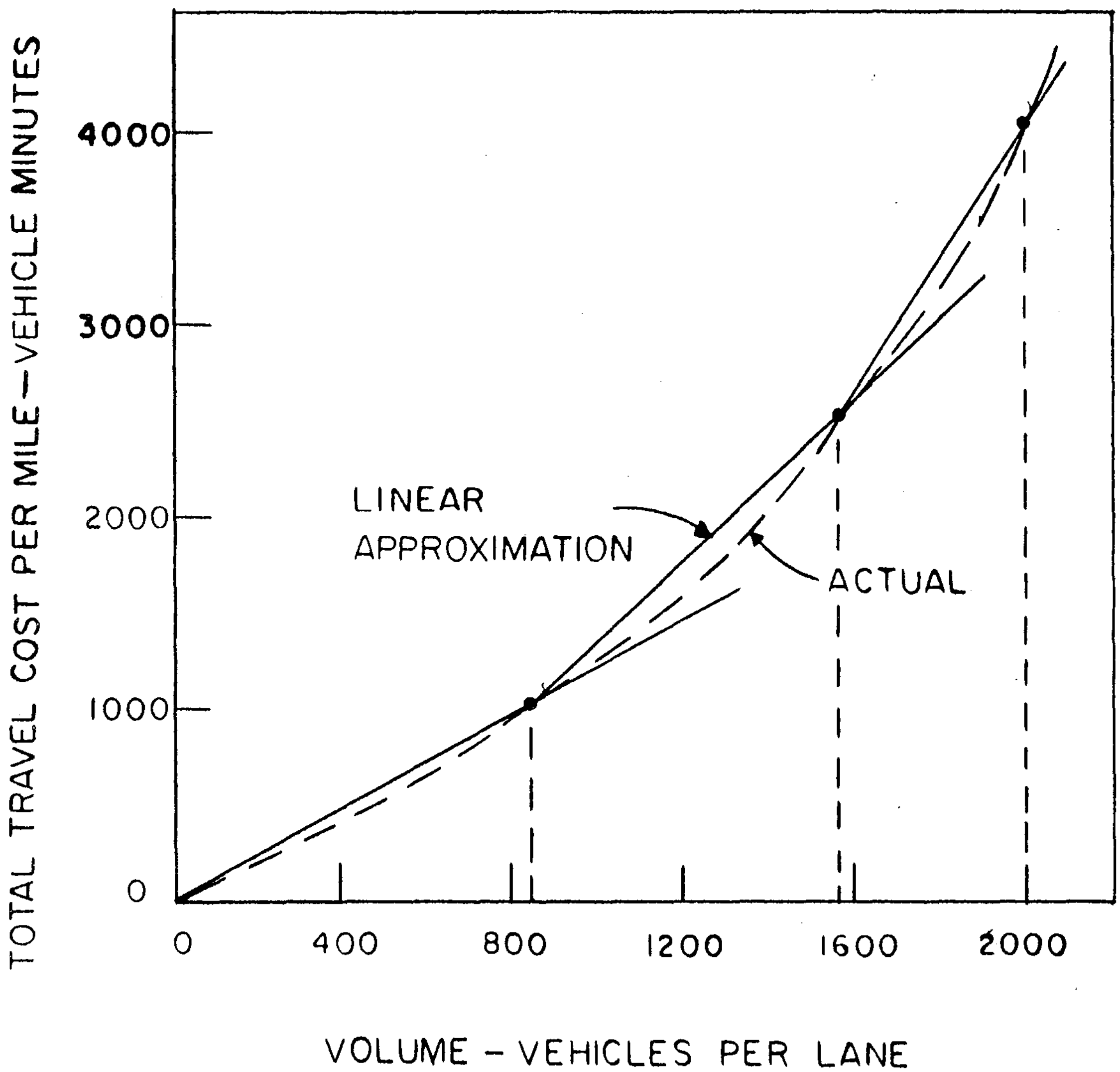
For volumes between 300 and 500 vehicles per minute ---

Travel time is 15 minutes per vehicle

At 100 vehicles per hour total travel time is 500 veh/min while at 300 vehicles per hour the total travel time is 2400 veh/min. Thus vehicles added in the 100-300 volume range should have a time cost of 2400-500/200 or 9.5 minutes per vehicle.

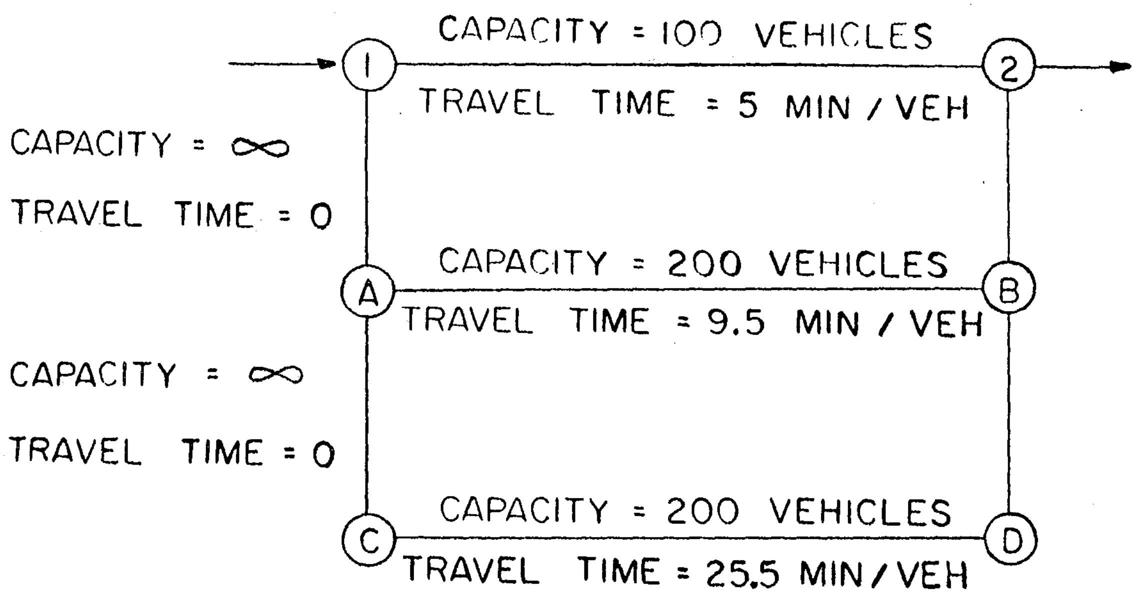
The vehicles added in the 300-500 volume range will be assigned a time cost of 7500-2400/200 or 25.5 minutes per vehicle.

The previous example would be represented by a network description like that shown in Figure 33.



PIECEWISE LINEAR APPROXIMATION
OF TRAVEL COST CURVE

FIGURE 32



NETWORK CODING FOR LINK WITH THREE LEVELS OF CAPACITY

FIGURE 33

Since Test Problem 3 involved numerous calculations when solved manually, a solution was obtained by use of the LP/90 program of the Texas A&M University Data Processing Center. The LP/90 program is a comprehensive system for linear programming developed by C.E.I.R., Inc. (7). In order to utilize this system, it was necessary to provide a general linear programming formulation of the problem. This formulation is as follows:

Minimize $$\sum_{j=1}^{20} C_j X_j$$

subject to

(1) $X_{11} - X_1 = -50$

(2) $X_1 - X_{15} - X_2 = 0$

(3) $X_2 - X_3 = 10$

(4) $X_3 + X_{14} = 0$

(5) $X_4 + X_{13} = 0$

(6) $X_5 + X_{12} = 100$

(7) $X_6 - X_7 = 20$

(8) $X_7 + X_{20} - X_8 = 0$

(9) $X_8 - X_9 = -80$

(10) $X_9 - X_{10} - X_{11} = 0$

(11) $X_5 - X_{20} - X_{18} + X_{19} = 0$

(12) $X_{18} - X_{19} - X_{13} - X_{16} + X_{17} = 0$

(13) $X_{16} - X_{17} - X_{14} + X_{15} = 0$

(14) $X_3 \leq 70$

(15) $X_{12} \leq 20$

(16) $X_{13} \leq 20$

(17) $X_{14} \leq 20$

(18) $X_j \geq 0$

The criterion function is a summation of individual link costs times the number of vehicles assigned to that link, which produces a total travel cost. Equations (1) through (13) are node equations, equations (14) through (17) represent capacity limitations and equation (18) imposes non-negativity restraints on the variables.

The results of the LP/90 solution to this problem were identical to those obtained with the Multi-Copy program.

Test Problem 4 - Test Problem 4 was obtained by adding three additional copies to Test Problem 2. The same 35 node network was utilized but additional traffic inputs were added. This provided an extension of the number of copies considered and considerable overload of the capacitated links. The network and traffic data are shown in Figure 34.

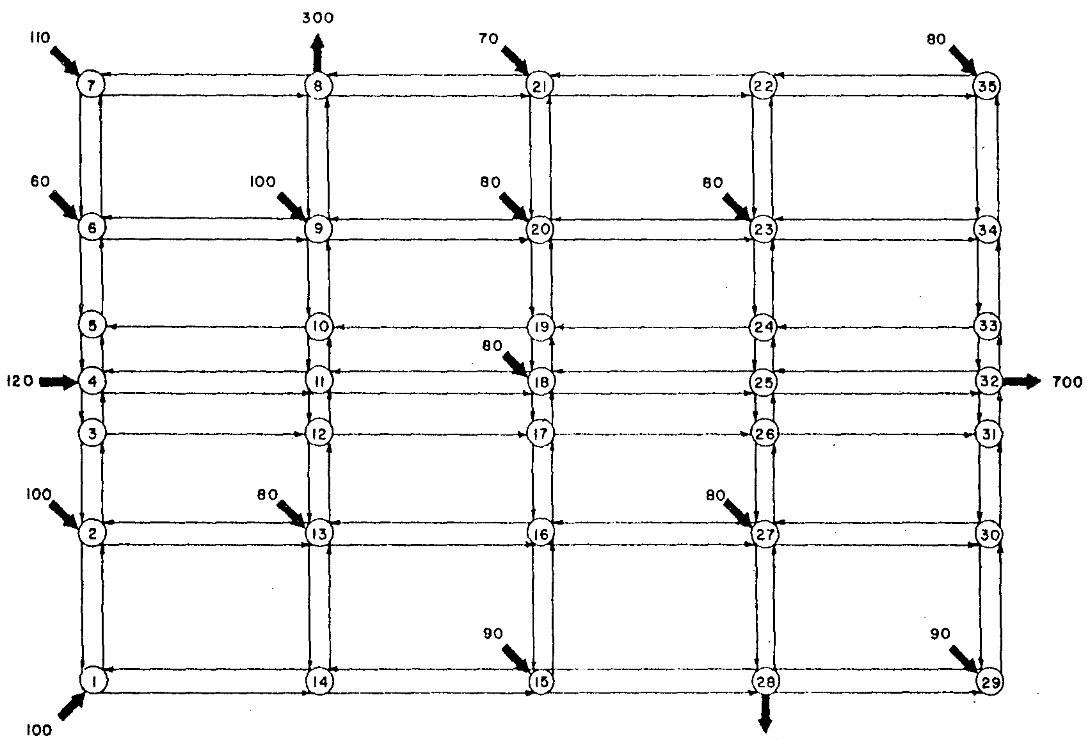
Test Problem 5 - Problem 5 again utilized the 35 node network and its basic objective was to provide a larger study problem which could be verified by LP/90. The 8 copy, 10 capacitated link problem used is shown in Figure 35.

Solution of this problem by LP/90 presented some difficulty because of the number of constraint equations involved when formulated as a general linear programming problem. For each copy, 35 node equations are required and the network size introduces 108 unknowns. The main matrix then must have an order of 280 by 864. When the capacitated link restrictions are added the total formulation has 291 rows and 875 columns.

The coding of this information required more than 3000 cards which presented a very sizeable card punching job. This job was simplified, however, by the use of a data conversion program and the IBM 1401.

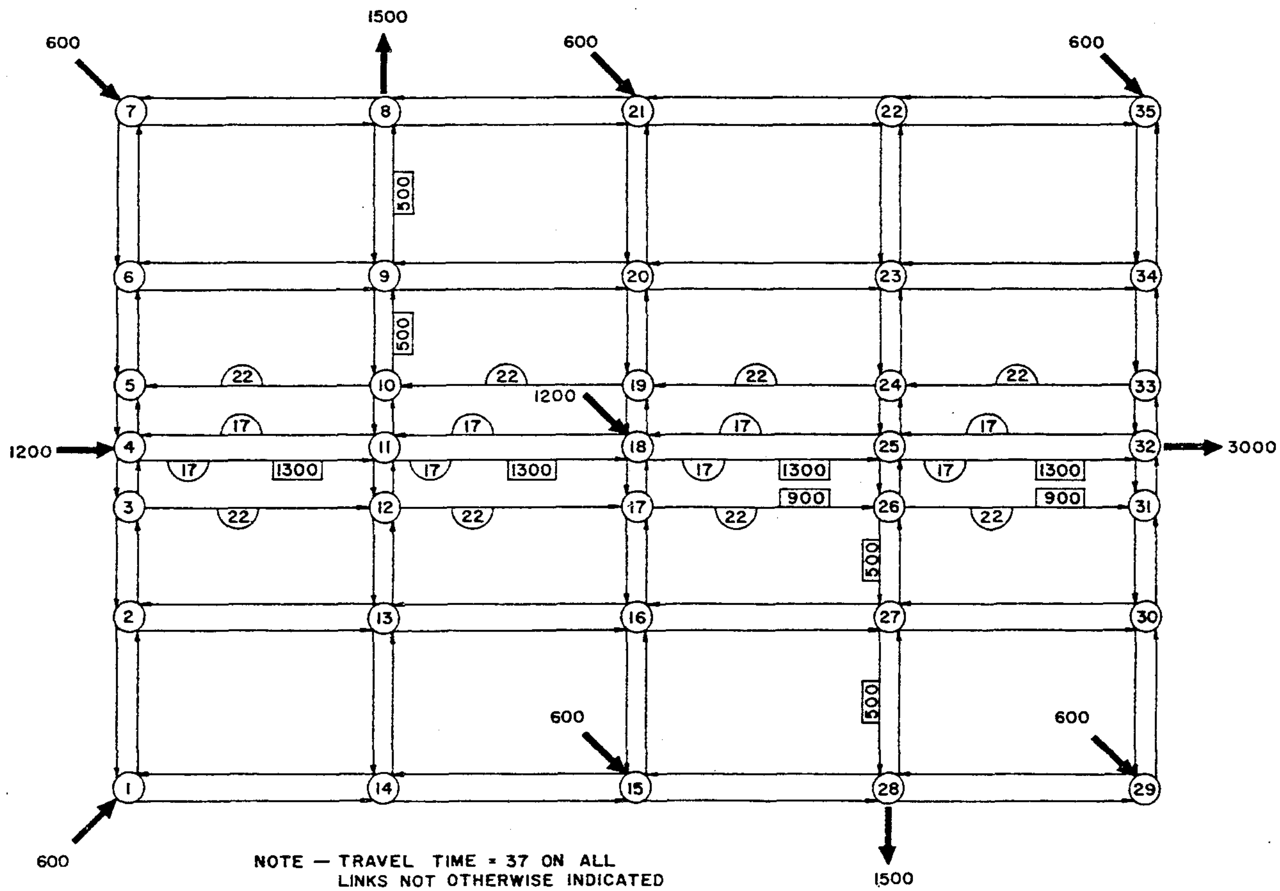
Test Problem 6 - This problem (see Figure 36) was used to study special effects of data scaling on the linear programming solution.

Test Problem 7 - This problem provided a larger system than could be handled by the initial program and was used as a test problem for the final program. The problem consisted of 10 copies and 20 capacitated links and required a 30 by 30 basic matrix. The network description and



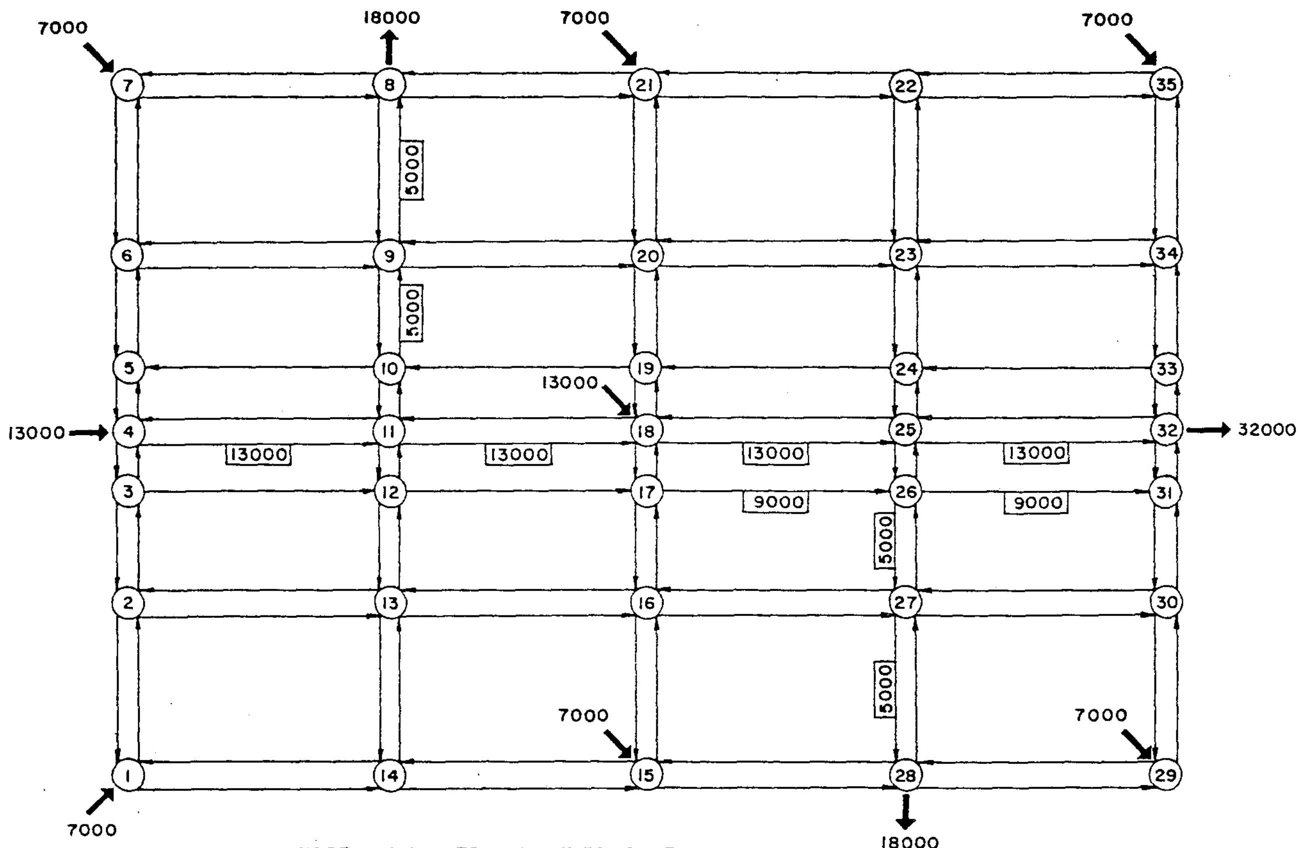
NOTE—TRAVEL TIMES AND CAPACITY RESTRAINTS ARE THE SAME AS TEST PROBLEM 2

NETWORK DESCRIPTION AND TRAFFIC DATA
TEST PROBLEM 4
FIGURE 34



NETWORK DESCRIPTION AND TRAFFIC DATA
TEST PROBLEM 5

FIGURE 35



NOTE — LINK TRAVEL TIMES SAME AS FOR PROBLEM 5

NETWORK DESCRIPTION AND TRAFFIC DATA TEST PROBLEM 6

FIGURE 36

traffic inputs for this problem are shown in Figure 37.

Test Problem 8 - This problem was designed to study run times for a large network using the final program. The network consisted of 256 nodes and 960 links, which approaches the network size limitations of the final program.

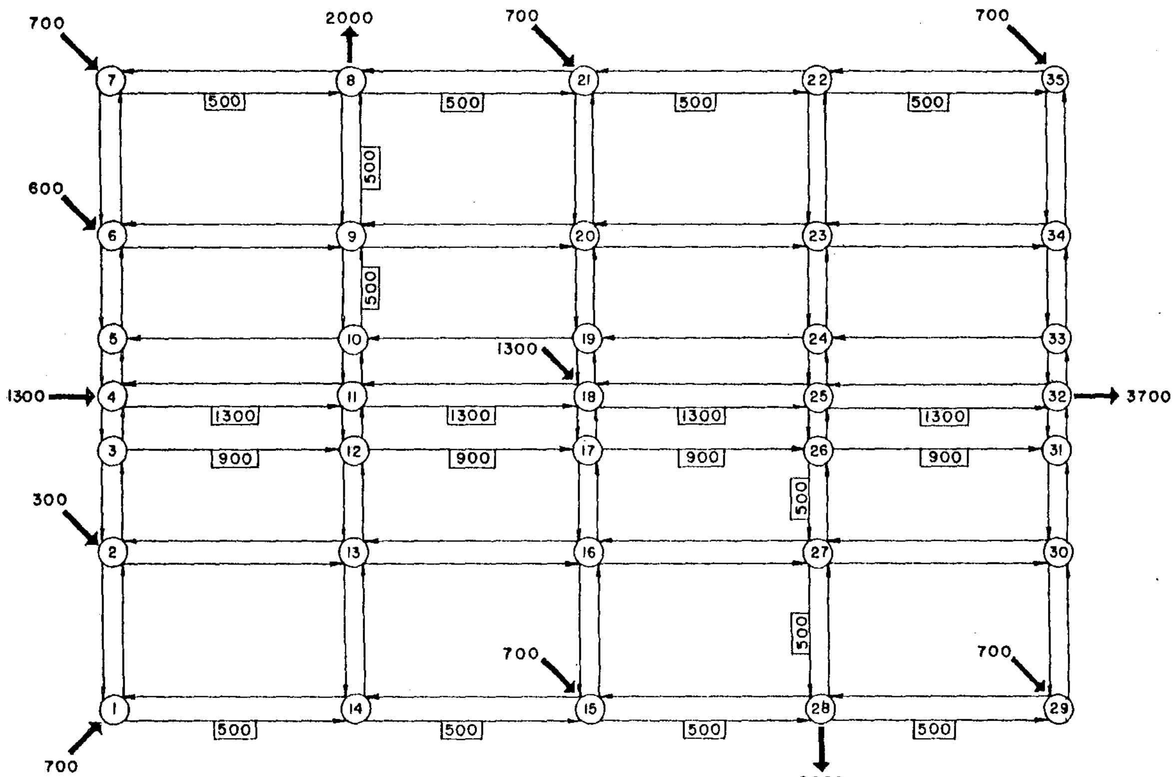
Problems Encountered

In developing the computer program for the linear programming model, problems of machine roundoff and slow conversions were encountered. These problems affected the accuracy of the output and the machine time required for problem solution. A detailed discussion of how these problems were treated is presented in reference (35).

Final Program

In order to handle a network analysis problem of the type encountered in a real system, it was necessary to make provisions in the initial program to handle much larger problems. This was done by placing the link data for individual copy loadings on tape instead of storing it in core. This provided more room in core for elements of a larger matrix and permitted the description of larger networks.

In addition, a terminate-restart procedure was incorporated to allow a break in execution on a long problem. With this technique available, it was possible to divide an extremely long problem into several short runs on the computer.



NOTE — TRAVEL TIMES SAME AS FOR TEST PROBLEM 5

NETWORK DESCRIPTION AND TRAFFIC DATA
TEST PROBLEM 7

FIGURE 37

Except for the above noted changes, the logical flow of the prog is essentially the same. The program divisions ;ame as t ..., i.e., a main program and 12 subroutines. Many of the original subroutines required no alternation other than changes in DIMENSION and COMMON statements, which were changed uniformly throughout the program.

The data input method was changed slightly to facilitate an improved minimum path procedure. The format of the output remained the same as for the initial program.

A total of four scratch tapes (three for the initial program) are used in the normal operation of the final program in addition to the regular input-output tapes. An additional tape is required for the intermediate dump when the terminate-restart option is utilized. The use of one additional tape operation by the final program did not appreciably affect run time on the Test Problems considered.

The limitations of the final program were as follows:

110th	order basic matrix
50	copies
60	capacitated links
1000	network links

An analysis of the core storage required for a problem of the above magnitude explains these limitations. A total of 29,151 storage locations are needed for such a problem compared to the 32,768 locations available on a machine with a 32K memory. The 3616 unused storage locations were reserved for additional modifications or additions to the program which might later be necessary.

Run Time

One of the most significant items for evaluation is that of the amount of computer run time required to obtain the solution to a given problem. This run time is related to a number of variables such as size of network, number of copies, number of capacitated links, and degree of overloading capacitated links. The most significant of these variables are (1) size of the network and (2) the number of copies.

The most time consuming operation in the program, because of its repetitive nature, is that of determining a minimum path from an input node to all output nodes. As the network size increases, the time required for this operation increases also. Similarly, as the number of copies increases, the minimum path determinations also increase. Large networks with a great number of copies thus require a considerable amount of machine time to compute the necessary minimum paths.

The 256 node and 960 link network of Test Problem 8 was utilized in connection with a two copy and five capacitated link system to study the run time on a large system. The minimum path calculations were timed to obtain a measure of run time required for this operation. Table 14 shows the findings of this run.

Thus it was found that for a 256 node system, approximately one minute of machine time is required for each minimum path. The number of minimum paths that must be calculated for any given program is a function of the number of copies and the degree to which the capacitated links are overloaded and is impossible to predict accurately.

The time required to determine a minimum path becomes a problem as the network size, the number of copies and the number of capacitated links increase. Improvement of computation speed on the path subroutine would significantly reduce over-all run time.

TABLE 14

RUN TIME FOR MINIMUM PATH DETERMINATIONS - TEST PROBLEM 8

Path Number		Time Required
1		1.04 minutes
2		1.02 minutes
3		1.04 minutes
4		1.04 minutes
5		1.04 minutes
6		1.02 minutes
7		1.01 minutes
8		1.01 minutes
9		1.04 minutes
	Avg. Time =	1.03 minutes

PROGRAM OUTPUT

On a normal run, four separate types of data are output as follows:

1. Data on the progress of the problem solution.
2. Final solution data.
3. Individual traffic loadings for all copies in the final basis.
4. Link volumes for loaded links in the network.

Solution Progress

Three types of program output are possible as a means of following the progress of the solution. The first type, called normal output is illustrated in Figure 38. Data on the following items are furnished:

1. Current Basis - Both the copy and extreme point solution number are provided. Thus, for example 6002 means 6-th copy and 2nd extreme point solution. A positive integer less than 1001 represents a slack vector and a negative integer represents an artificial vector.
2. Z(j) minus C(j) values - Separate values of Z(j) - C(j) are listed along with the copy and extreme point solution number.
3. Type of Vector Entered - This will be either a solution vector or slack vector. The entry position in P(o)' is also indicated.
4. Current Cost - The current cost of the solution after the entry of the indicated vector is provided.

In order to locate sources of program error it was necessary to output more information than the normal output. Two other types of

```
BEGINNING IBAS
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2     3     4     5     6     7     8     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   15          CURRENT COST  0.10840800E 05           Z(J) - C(J) =  0.47599992E 03
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2     3     4  2002     6     7     8     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   17          CURRENT COST  0.10819499E 05           Z(J) - C(J) =  0.55379996E 03
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2     3     4  2002     6  2003     8     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   16          CURRENT COST  0.10787333E 05           Z(J) - C(J) =  0.53074992E 03
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2     3     4  2002  2004  2003     8     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   15          CURRENT COST  0.10723078E 05           Z(J) - C(J) =  0.65266653E 03
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2     3     4  2005  2004  2003     8     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   18          CURRENT COST  0.10668666E 05           Z(J) - C(J) =  0.68655416E 03
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2     3     4  2005  2004  2003  2006     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   13          CURRENT COST  0.10554999E 05           Z(J) - C(J) =  0.10229997E 04
 1001  2001  3001  4001  5001  6001  7001  8001  9001 10001      1     2  2007     4  2005  2004  2003  2006     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO    2          CURRENT COST  0.10554999E 05           Z(J) - C(J) =  0.31100000E 03
 1001  2006  3001  4001  5001  6001  7001  8001  9001 10001      1     2  2007     4  2005  2004  2003  2006     9    10
   11    12    13    14    15    16    17    18    19    20

VECTOR INTO   15          CURRENT COST  0.10424809E 05           Z(J) - C(J) =  0.53399995E 03
 1001  2006  3001  4001  5001  6001  7001  8001  9001 10001      1     2  2007     4  3002  2004  2003  2006     9    10
   11    12    13    14    15    16    17    18    19    20

SLACK INTO    21
 1001  2006  3001  4001  5001  6001  7001  8001  9001 10001      1     2  2007     4  3002  2004  2003  2006     9    10
   21    12    13    14    15    16    17    18    19    20

SLACK INTO    16
 1001  2006  3001  4001  5001  6001  7001  8001  9001 10001      1     2  2007     4  3002    22  2003  2006     9    10
   21    12    13    14    15    16    17    18    19    20
```

SOLUTION PROGRESS DATA-NORMAL OUTPUT

FIGURE 38

output were used for this purpose - an intermediate output illustrated in Figure 39 and a detailed output as shown in Figure 40.

For the detailed output, the following data are available:

1. Vector entered and position.
2. Current Cost and Z(j) - C(j) value.
3. Minimum paths from input nodes to all other nodes.
4. Elements of the cost vector showing costs of each copy solution in the current basis.
5. Elements of the entering copy or P(j) vector.
6. Elements of the primed entering copy or P(j)' vector.
7. Elements of the P(o)' or solution vector.
8. Elements of the current main matrix.
9. Elements of the Omega vector.

The intermediate output eliminates the minimum path and matrix data.

For all cases the data are output after each iteration of the problem. With detailed output it is possible to follow each specific step of the solution.

Solution Output

After convergence, the final solution data shown in Figure 41 are output. This provides information on the final basis, the cost of each copy solution in the final basis, the solution vector, the Omega vector and the total cost of the optimum solution.

In addition, data on individual copy loadings (Figure 42) and total link loadings (Figure 43) are available. The individual copy

```
BEGINNING IBAS
 1001  2001  3001  4001  5001  6001  7001  8001     -1    -2     1     2    -3    -4    -5    -6    -7    -8

ZJ= 0.16583999E 07           CJ= 0.93000000E 04            ZJMICJ= 0.16490999E 07          1002

VECTOR INTO     1              CURRENT COST  0.42564897E 07            Z(J) - C(J) =  0.16490999E 07          TIME =  0.
 1002  2001  3001  4001  5001  6001  7001  8001     -1    -2     1     2    -3    -4    -5    -6    -7    -8

ENTERING COPY -
  0.                0.                0.30000000E 02   0.30000000E 02   0.                0.                0.
  0.                0.                0.               0.09999999E 01   0.                0.                0.
  0.                0.                0.               0.

PRIMED ENTERING COPY -
  0.10000000E 01  -0.37252903E-08  -0.                -0.54715201E-08  -0.33760445E-08   0.55297272E-09  -0.
 -0.27939676E-07   0.35762787E-06   0.35762787E-06   0.30000000E 02   0.30000000E 02   0.45000001E 02   0.30000000E 02
  0.30000000E 02   0.29999998E 02   0.14999999E 02   0.14999999E 02

PIOI PRIME VECTOR -
  0.10000001E 01   0.99999998E 00   0.09999999E 01   0.99999995E 00   0.99999995E 00   0.99999993E 00   0.09999999E 01
  0.99999970E 00   0.39999998E 02   0.39999998E 02   0.59999996E 02   0.59999996E 02   0.20000603E 02   0.50000012E 01
  0.16999999E 03   0.94999990E 02   0.24999998E 02   0.24999996E 02

OMEGA VECTOR -
 -0.09999999E 05  -0.09999999E 05   0.                0.               -0.09999999E 05  -0.09999999E 05  -0.09999999E 05
 -0.09999999E 05  -0.09999999E 05  -0.09999999E 05   0.93000000E 04   0.48093597E 07   0.13562249E 07   0.90712499E 06
  0.25557899E 07   0.12071249E 07   0.30520499E 06   0.30635998E 06

ZJ= 0.93000000E 04           CJ= 0.93000000E 04            ZJMICJ= 0.                       1003

ZJ= 0.48093597E 07           CJ= 0.11910000E 05            ZJMICJ= 0.47974497E 07          2002

VECTOR INTO    14              CURRENT COST  0.40280396E 07            Z(J) - C(J) =  0.47974497E 07          TIME =  0.
 1002  2001  3001  4001  5001  6001  7001  8001     -1    -2     1     2    -3  2002    -5    -6    -7    -8

ENTERING COPY -
  0.                0.                0.90000000E 02   0.90000000E 02   0.                0.                0.
  0.                0.                0.               0.               0.09999999E 01   0.                0.
  0.                0.                0.               0.

PRIMED ENTERING COPY -
  0.                0.99999999E 00   0.               0.               0.                0.                0.
  0.                0.15000000E 02   0.15000000E 02   0.89999996E 02   0.89999996E 02   0.12000000E 03   0.10500000E 03
  0.10500000E 03   0.89999993E 02   0.14999998E 02   0.14999997E 02

PIOI PRIME VECTOR -
  0.10000001E 01   0.95238092E 00   0.09999999E 01   0.99999995E 00   0.99999995E 00   0.99999993E 00   0.09999999E 01
  0.99999970E 00   0.39285712E 02   0.39285712E 02   0.55714281E 02   0.55714281E 02   0.14285716E 02   0.47619058E-01
  0.16499999E 03   0.90714275E 02   0.24285711E 02   0.24285710E 02

OMEGA VECTOR -
 -0.09999999E 05  -0.09999999E 05   0.                0.               -0.09999999E 05   0.35689997E 05  -0.09999999E 05
 -0.09999999E 05  -0.09999999E 05  -0.09999999E 05   0.93000000E 04   0.11910000E 05  -0.14474982E 05   0.90712499E 06
  0.25557899E 07   0.12071249E 07   0.30520499E 06   0.30635998E 06

SLACK INTO    13
 1002  2001  3001  4001  5001  6001  7001  8001     -1    -2     1     2     3  2002    -5    -6    -7    -8

ENTERING COPY -
  0.                0.                0.               0.               0.                0.09999999E 01   0.
  0.                0.                0.               0.               0.                0.                0.
  0.                0.                0.               0.

PRIMED ENTERING COPY -
  0.                0.95238093E-02   0.               0.               0.                0.                0.
  0.                0.14285714E-00   0.14285714E-00   0.85714281E 00   0.85714281E 00   0.11428571E 01  -0.95238094E-02
  0.99999996E 00   0.85714278E 00   0.14285712E-00   0.14285711E-00

PIOI PRIME VECTOR -
  0.10000001E 01   0.83333328E 00   0.09999999E 01   0.99999995E 00   0.99999995E 00   0.99999993E 00   0.09999999E 01
  0.99999970E 00   0.37499997E 02   0.37499997E 02   0.44999994E 02   0.44999994E 02   0.12500001E 02   0.16666669E-00
  0.15250000E 03   0.79999989E 02   0.22499997E 02   0.22499996E 02

OMEGA VECTOR -
 -0.09999999E 05  -0.09999999E 05   0.                0.                0.21228748E 05   0.               -0.09999999E 05
 -0.09999999E 05  -0.09999999E 05  -0.09999999E 05   0.93000000E 04   0.11910000E 05   0.11936247E 06   0.90712499E 06
  0.25557899E 07   0.12071249E 07   0.30520499E 06   0.30635998E 06

SLACK INTO    11
 1002  2001  3001  4001  5001  6001  7001  8001     -1    -2     4     2     3  2002    -5    -6    -7    -8

ENTERING COPY -
  0.                0.                0.               0.               0.09999999E 01   0.                0.
  0.                0.                0.               0.               0.                0.                0.
  0.                0.                0.               0.

PRIMED ENTERING COPY -
  0.                0.83333331E-02   0.               0.               0.                0.                0.
  0.                0.12500000E-00   0.12500000E-00   0.74999996E 00   0.74999996E 00  -0.87500000E 00  -0.83333331E-02
  0.87499996E 00   0.74999993E 00   0.12499998E-00   0.12499997E-00

PIOI PRIME VECTOR -
  0.10000001E 01   0.33333333E-00   0.09999999E 01   0.99999995E 00   0.99999995E 00   0.99999993E 00   0.09999999E 01
  0.99999970E 00   0.29999997E 02   0.29999997E 02   0.59999996E 02   0.47683716E-06   0.64999998E 02   0.66666664E 00
  0.10000000E 03   0.34999996E 02   0.14999999E 02   0.14999998E 02
```

SOLUTION PROGRESS DATA
INTERMEDIATE OUTPUT

FIGURE 39

```
BEGINNING IBAS
 100'  2001  3001  4001  5001  6001  7001  8001  9001 10001 11001 12001 13001 14001 15001     1    -1    -2    -3

 ZJ= U.21C84999E 07          CJ= 0.10600000E 05          ZJMICJ= 0.20978999E 07          1002

 VECTOR INTO    1          CURRENT COST  0.39790998E 07          Z(J) - C(J) =  0.20978999F 07          TIME =  0.
  1C02  2001  3001  4001  5001  6001  7001  8001  9001 10001 11001 12001 13001 14001 15001     1    -1    -2    -3

 MINIMUM PATHS -
     2    17 77777     3    35    17 77777     4    49    35    17 77777     5    67    49    35    17 77777     6    81
    67    49    35    17 77777     7    99    81    67    49    35    17 77777     8    97    79    65    51    49    35
    17 77777     9    79    65    51    49    35    17 77777    10    65    51    49    35    17 77777    11    51    49
    35    17 77777    12    33    15     1 77777    13    15     1 77777    14     1 77777    15     4     1 77777    16
    22    15     1 77777    17    38    33    15     1 77777    18    45    38    33    15     1 77777    19    63    45
    38    33    15     1 77777    20    77    63    45    38    33    15     1 77777    21   103    97    79    65    51
    49    35    17 77777    22   106   103    97    79    65    51    49    35    17 77777    23    75    61    43    39
    38    33    15     1 77777    24    61    43    39    38    33    15     1 77777    25    43    39    38    33    15
     1 77777    26    39    38    33    15     1 77777    27    11     5     4     1 77777    28     5     4     1 77777
    29     8     5     4     1 77777    30     9     8     5     4     1 77777    31    40    39    38    33    15     1

 COST VECTOR -
    0.10600000E 05   0.58000000E 04   0.81999999E 04   0.45000000E 04   0.43000000E 04   0.51000000E 04   0.30000000E 04
    0.38000000E 04   0.41999999E 04   0.34999999E 04   0.50000000E 04   0.47000000E 04   0.77000000E 04   0.43000000E 04
    0.44000000E 04   0.               0.09999999E 05   0.09999999E 05   0.09999999E 05

 ENTERING COPY -
    0.70000000E 02   0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.

 PRIMED ENTERING COPY -
    0.09999999E 01   0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.70000000E 02   0.70000000E 02   0.69999999E 02   0.69999999E 02

 P(O) PRIME VECTOR -
    0.09999999E 01   0.99999995E 00   0.09999999E 01   0.09999999E 01   0.09999999E 01   0.09999999E 01   0.99999987E 00
    0.09999999E 01   0.99999980E 00   0.09999999E 01   0.09999999E 01   0.09999999E 01   0.09999999E 01   0.09999999E 01
    0.09999999E 01   0.80000000E 02   0.10000002E 02   0.20999999E 03   0.16999999E 03

 MATRIX -
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.99999999E 00   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.

    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.99999999E 00   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.99999999E 00   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.09999999E 01   0.               0.09999999E 01
    0.               0.               0.              -0.70000000E 02   0.               0.               0.
    0.               0.               0.               0.               0.               0.               0.
    0.               0.               0.               0.               0.               0.              -0.09999999E 01
    0.               0.               0.               0.09999999E 03   0.80000000E 02   0.70000000E 02   0.50000000E 02
    0.               0.               0.               0.               0.               0.               0.
    0.30000000E 02   0.40000000E 02   0.               0.               0.               0.              -0.09999999E 01
    0.               0.               0.99999999E 02   0.79999999E 02   0.69999999E 02   0.49999999E 02   0.49999999E 02
    0.59999999E 02   0.40000000E 02   0.               0.               0.               0.               0.30000000E 02
    0.39999999E 02   0.49999999E 02   0.               0.               0.               0.              -0.09999999E 01
    0.               0.79999999E 02   0.69999999E 02   0.59999999E 02   0.49999999E 02   0.49999999E 02   0.39999999E 02
    0.30000000E 02   0.40000000E 02   0.30000000E 02   0.               0.               0.29999999E 02   0.20000000E 02
    0.29999999E 02   0.

 OMEGA VECTOR -
    0.              -0.09999999E 05  -0.09999999E 05  -0.09999999E 05   0.10599999E 05   0.28057999E 07   0.23081999E 07
    0.20044999E 07   0.15042999E 07   0.10050999E 07   0.10029999E 07   0.70379999E 06   0.40420000E 06   0.30349999E 06
    0.50000000E 04   0.47000000E 04   0.90769998E 06   0.10043000E 07   0.80439997E 06

 ZJ= 0.10599999E 05          CJ= 0.10600000E 05          ZJMICJ=-0.36621094E-03          1003

 ZJ= 0.28057999E 07          CJ= 0.84000000E 04          ZJMICJ= 0.27973999E 07          2002
```

SOLUTION PROGRESS DATA - DETAILED OUTPUT

FIGURE 40

LINEAR PRCGRAMMING - MIXING MCDEL TRAFFIC STUCY FCR PRCBLEM NUMBER 8

FINAL EASIS CESCRIPTICN -

2C11	1C13	3C16	4CC5	5CC5	1CC12	7C01	8CC1	2030	1C14	6C12	2	1C11	10C18	1CC15	48	10016	6013	4018	5006
2C29	12	13	14	7C	16	1CC17	18	19	2C										

CCST VECTCR -

C.1129CCCCE C4	C.15539999E C4	C.833CCCCCE C3	C.845CCCCCE C3	C.391CCCCCE 03	C.1C8CCCCCE 04	0.67699999E 03
C.814CCCCCE C3	C.1249CCCCE C4	C.1258CCCCE C4	C.737CCCCCE C3	C.	C.12279999E 04	0.75500000E 03
C.8CCCCCCCE C3	C.	C.775CCCCCE C3	C.777CCCCCE 03	C.925CCCCCE 03	C.9430000CE 03	0.13420000E 04
C.	C.	C.	C.	0.	C.7550000CE 03	C.
C.	C.					

F(C) PRIME VECTCR -

C.66666675E CC	-C.1937151CE-C6	C.10CCCCC5E C1	C.50CC6762E-01	C.1CCCCC17E 01	C.79999949E CC	0.09999999E 01
C.C9999999E C1	C.33333278E-CC	C.5CCCCC08E C0	-0.59604645E-07	0.48CCCC67E C1	C.49999998E-CC	C.66669099E-01
C.18239C43E-C5	C.	-C.35464764E-C5	C.99999966E CC	C.95CCC6C7E CC	C.1CCCCCCCE 01	C.16763806E-06
C.41999946E C1	C.	C.	C.2CCCC66CE-CC	C.5CCCCCCCE C1	C.13333345E-0C	C.5C00000CE 01
C.99999989E CC	C.5CCCCCCCE C1					

CMEGA VECTCR -

-C.74CC1C91E C2	C.	-C.2CCCCCCCE C2	C.	-C.5CCC1964E C1	-C.5C001993E 01	-C.43999591E 02
-C.2CCCC251E C2	-C.38999605E C2	-C.15CCC257E C2	C.	C.	C.	C.
-C.148CCCC9E C3	C.	-C.5CCCC038E C1	C.	C.	C.	0.15640000E 04
C.1568CC55E C4	C.1164CC24E 04	C.925C0C00E C3	C.14740053E 04	C.1221CCC3E C4	C.69699999E 03	C.81400000E 03
C.554CC175E C3	C.1C85CCCCE C4					

THE CVERALL CCST CF THE FINAL SCLUTICN IS C.87769999E 04.

FINAL SOLUTION DATA

FIGURE 41

PAGE 2

THE COPIES IN THE FINAL BASIS ARE AS FOLLOWS -

COLUMN 1 COPY 1001 CORRESPONDING P(0) PRIME ELEMENT = 0.763592 -

LINK NO.	LOAD	LINK NO.	LOAD	LINK NO.	LOAD	LINK NO.	LOAD
3	2	4	2	5	5	6	3
7	3	8	3	13	2	14	2
15	2	26	5	43	5	52	5
68	2						

PAGE 3

COLUMN 7 COPY 7001 CORRESPONDING P(0) PRIME ELEMENT = 0.411094 -

LINK NO.	LOAD	LINK NO.	LOAD	LINK NO.	LOAD	LINK NO.	LOAD
3	2	4	2	24	3	33	4
50	4	60	4	62	2	63	2
64	2	68	2				

PAGE 4

COLUMN 8 COPY 8001 CORRESPONDING P(0) PRIME ELEMENT = 1.000000 -

LINK NO.	LOAD	LINK NO.	LOAD	LINK NO.	LOAD	LINK NO.	LOAD
24	2	32	2	49	2	59	2
73	5	86	5	103	5	106	2
107	2	108	2				

PAGE 5

COLUMN 12 COPY 2 CORRESPONDING P(0) PRIME ELEMENT = 5.000081 -

LINK NO. LOAD LINK NO. LOAD LINK NO. LOAD LINK NO. LOAD

THIS SLACK WAS IN COLUMN 12 OF THE ORIGINAL BASIS.

PAGE 6

COLUMN 22 COPY 12 CORRESPONDING P(0) PRIME ELEMENT = 4.000106 -

LINK NO. LOAD LINK NO. LOAD LINK NO. LOAD LINK NO. LOAD

THIS SLACK WAS IN COLUMN 22 OF THE ORIGINAL BASIS.

PAGE 7

COLUMN 23 COPY 13 CORRESPONDING P(0) PRIME ELEMENT = 3.000003 -

LINK NO. LOAD LINK NO. LOAD LINK NO. LOAD LINK NO. LOAD

THIS SLACK WAS IN COLUMN 23 OF THE ORIGINAL BASIS.

PAGE 8

INDIVIDUAL COPY LOADING DATA

FIGURE 42

TFE FCLLCWING IS TFE CCNVEX CCMBINATICN CF ALL CCPIES IN TFE FINAL BASIS -

LINK NC.	LCAC	LINK NC.	LCAC	LINK NC.	LOAC	LINK NO.	LOAD
1	5	3	5	4	5	5	13
6	13	7	13	8	13	9	9
1C	9	11	9	12	5	13	5
14•	5	15	5	17	5	19	4
24	1C	26	2	3C	7	32	7
33	4	34	1	35	1	36	13
37	8	43	4	46	7	47	2
49	4	5C	9	51	6	52	1
54	4	55	11	56	2	57	9
58	9	59	4	6C	18	62	1
63	2	64	2	65	6	66	1
68	5	7C	5	73	12	79	6
8C	1	83	5	86	12	88	3
89	3	9C	3	91	7	96	4
97	2	1CC	5	1C1	4	103	5
1C6	1C	107	2	1C8	2		

PAGE 33

TFE FOLLCWING ARE TFE ALTERNATE SCLUTICNS THAT CCURREC IN TFE FINAL BASIS WITH TFEIR CCRRESPONDING ENTRY POINTS -

1C16 2 2C32 17 3C2C 17 4019 15 5C24 15 • 6018 2 7006 11 8006 8
9C11 15

TOTAL LINK LOADING DATA

FIGURE 43

loadings show the links in the copy solution, the traffic assigned to each link and the percent of the copy solution that will be utilized. In the case of a slack vector in the final solution, it will be indicated as shown in Figure 42.

The principal answer to a given problem is shown by the total link loading data in Figure 43. The proper percentages of the link loadings on each copy are obtained and combined to give total traffic on each loaded link of the system.

Copy Solutions

Since any network problem whose combined minimum path flow overloads any given capacitated link will produce a "mixture" of copy solutions, it is of interest to consider the multiple copy solutions. These solutions represent a diversion from minimum path flow and correspond to the proportional assignment technique developed by Irwin, et al. From the listing of traffic flow on each of the copy solutions in the final basis (Figure 42), the diverted traffic and the routes it follows can be observed.

Omega Vector

The Primal of the multi-copy problem is formulated as follows:

$$\text{Minimize} \quad \sum_{\alpha=1}^{m} \sum_{\beta=1}^{n(\alpha)} C^{\alpha T} e^{\alpha\beta} \mu_{\alpha\beta}$$

$$\text{subject to} \quad \sum_{\alpha=1}^{m} \sum_{\beta=1}^{n(\alpha)} K^{\alpha} e^{\alpha\beta} \mu_{\alpha\beta} = d$$

$$\sum_{\beta=1}^{n(\alpha)} \mu_{\alpha\beta} = 1$$

$$\mu_{\alpha\beta} \geq 0$$

The criterion function for the dual to the problem is as follows:

$$\text{Max } \omega_1 d_1 + \omega_2 d_2 + \ldots + \omega_n d_n + \omega_{n+1} + \omega_{n+2} + \ldots + \omega_k$$

where d_i = capacity limitation on link (i)

ω_i = the i-th variable of the dual solution

n = number of capacitated links

k = sum of copies plus capacitated links

At an optimum solution the total cost of the solution (G) is given by

$$G = \bar{C}\ P(0)'.$$

Since $P(0)' = B^{-1}\ P(0)$

then $G = \bar{C}\ B^{-1}\ P(0).$

but ω is defined as $\bar{C}\ B^{-1}$.

Therefore $G = \omega\ P(0)$ or

$$G = \omega_1 d_1 + \omega_2 d_2 + \ldots + \omega_n d_n + \omega_{n+1} + \omega_{n+2} + \ldots + \omega_k.$$

Thus the Omega vector contains the dual variables of the optimum solution and provides evaluators for the capacitated links.

Consideration of the first n elements of the Omega vector make it possible to determine which capacitated links contribute the greatest detrimental effect to the network flow. For example, consider the example problem (Figure 7) which had two capacitated links and the following Omega vector when an optimum solution was reached.

$$\omega = [-3, -5, 810, 600]$$

ω_1 is associated with link 3 and ω_2 with link 9. If it were possible to increase the link capacity values, the greatest improvement in the total network flow costs would result from changing link 9. Thus, the final output provides a valuable tool to analyze the capacitated links of a traffic network.

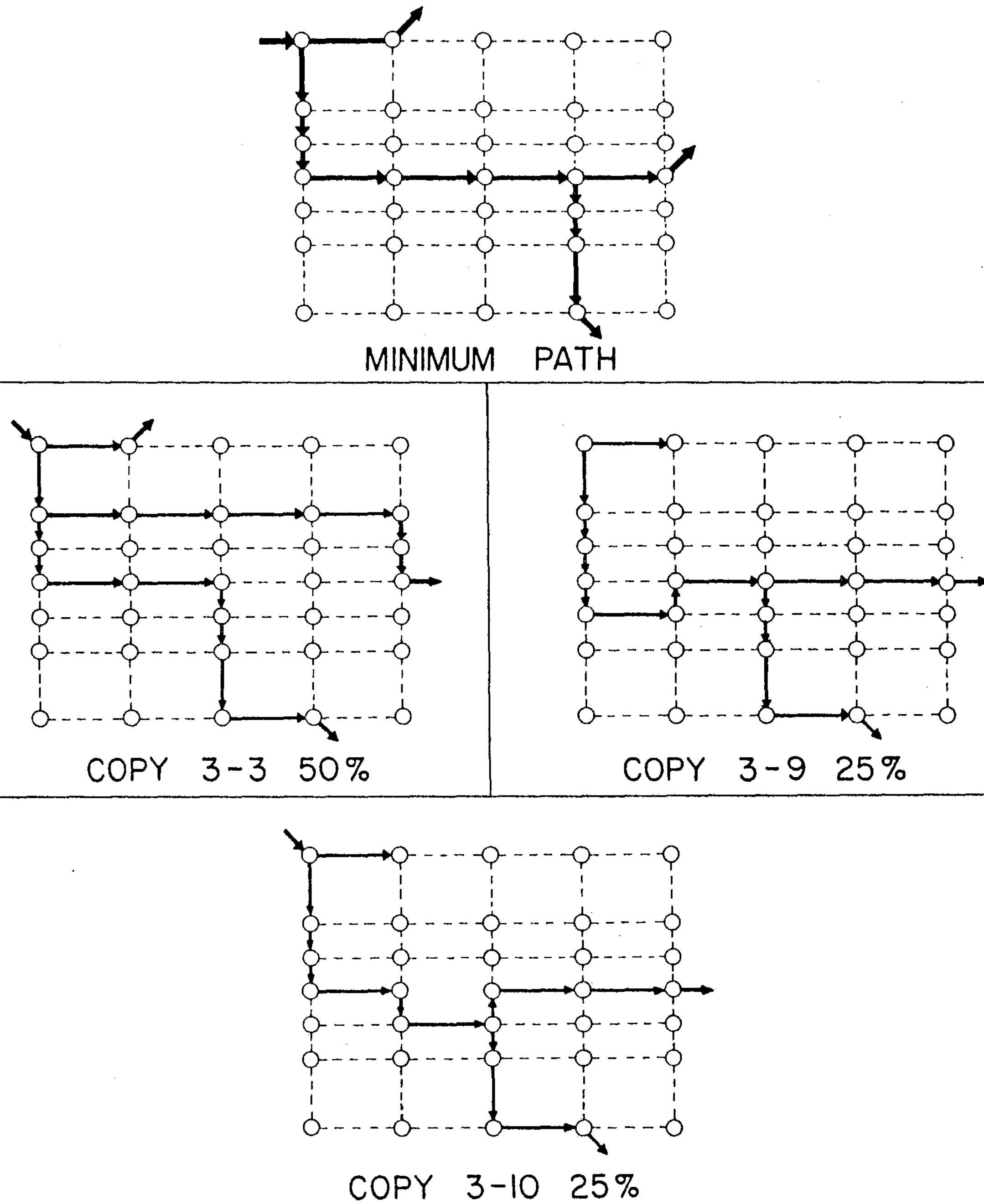
The results obtained from the solution from the various test problems indicate that the desired answer can be obtained by the multicopy program and that it provides insight to the distribution problem. The program considers each origin to destination requirement and the routing of this trip concerning capacity requirements and over-all system travel times.

The output of the computer program for an optimum solution to the problem shows how traffic moving from each input node to all output nodes should be routed. These optimum routings can then be compared to the minimum path routings to find that traffic which should be diverted from the freeway by some type of operational control.

Figure 44 shows an example of the traffic diversion previously discussed. The first drawing in Figure 44 shows the minimum path that traffic would follow from a given input node to the specified output nodes if there were no capacity restraints. However, the solution of the problem by the optimization technique produced the combination of routings shown in the next three drawings.

It would be impossible to route traffic exactly as indicated by the solution shown in Figure 44 but it would be possible to affect some diversion of traffic from the general area of the city represented by the input node in question. Thus from the solution it would be possible to pinpoint the specific origins of traffic which should be diverted. This diversion could possibly be accomplished through ramp closures, one-way street operation, advisory signing, etc.

In most cases the traffic which must be diverted from the freeway is short trip traffic. However, the ability of the at-grade facility to move traffic would also be reflected in the assignment to the freeway. Traffic moving relatively short distances in areas where a poor supporting street system exists might be more adversely affected than traffic moving much greater distances over better at-grade facilities.



TRAFFIC DIVERSION BY COPIES

FIGURE 44

OPTIMUM DISTRIBUTION OF TRAFFIC

Solution Required

Before it is possible to develop a control system to operate a street network, the basic question of which traffic will be controlled must be answered. The freeway links of the network have the greatest potential for traffic movement and thus it is extremely important that their ability to move traffic be maintained at a high level. The basic question can therefore be narrowed to ask what traffic must be diverted from the freeway to maintain its high level of efficiency and to provide for an optimum assignment of traffic over the system.

An algorithmic approach which was tried might be stated as follows:

1. Close off all capacitated links or assign them a very high cost and search a minimum path of each origin to all destinations on all copies.
2. Compute the time cost of each of the above minimum paths and rank these in order with the highest cost path first followed by the second highest cost path and so forth.
3. Assign traffic to the network considering the previously developed ranking. Allow traffic to utilize the capacitated links of the network until the indicated capacity of some link is reached. When the capacity of a link is reached this link is removed from the system and the assignment is continued until all traffic has been assigned.

The advanced procedure was programmed by Blumentritt and it provided a very rapid assignment technique. It was found that this algorithmic technique produced the correct answer to test problem 2 and approximated very closely the correct answer to test problem 4. It was found, however, that it would be necessary to introduce some refinements into the technique in order to duplicate the multi-copy solution over a wide range of problems. Work with this algorithmic approach is being continued and it offers a good approach to studying optimum distribution of traffic over a street network. The advantage of the algorithmic assignment technique is that it can be programmed to run much faster than the multi-copy model and can deal with much larger traffic networks than is possible with a linear programming solution.

Algorithmic Assignment

A technique which shows a great deal of promise for the study of optimum traffic distribution is an algorithmic assignment to a given network which would approach the linear programming solution produced by the multi-copy model. Such techniques are being studied by Blumentritt (2) and some progress has been made toward obtaining successful solutions by this method.

The basic element in the algorithmic assignment is the individual movements between a given origin and destination. It is possible to consider the effect or cost of a trip between a given origin and destination if this trip cannot be made over the capacitated links of the system or, in the case of a real problem, the freeway. Those trips which sustain the most adverse effect are then assigned to the capacitated links in that order until link capacity is reached. The remaining trips must then be diverted from their minimum cost path and would become the subjects of a traffic control program.

CONCLUSIONS AND RECOMMENDATIONS

Conclusions

The research reported herein has concerned itself with the development of a computer program to provide a linear programming solution for the optimum distribution of traffic over portions of a street network. This program was developed, evaluated and its application to the problem of developing operational controls for street networks was studied. The following conclusions were reached as a result of this research.

1. The multi-copy linear programming model can be adapted to a computer program which will provide a solution to the optimum distribution problem for small networks.
2. The solution to the optimum distribution problem provides information which would be of significant value in planning and designing a traffic control system to operate an arterial street network. The origin of trips which should be diverted from the freeway can be determined and this information utilized in the design of the control system.
3. The multi-copy program was found to be limited to the following conditions:

   50 input nodes

   60 capacitated links

   300 network nodes

   1000 links

   In addition to the network limitations the problems of slow convergence and long machine times indicated the undesirability of utilizing the linear programming solution when dealing with large networks.

4. Preliminary investigations indicate that it is possible to duplicate the linear programming solution of the optimum distribution problem by an algorithmic technique. Such a technique offers the decided advantage of a much faster solution on the computer and the ability to handle a large network.

5. The multi-copy program is not applicable to normal traffic assignments for urban planning but is intended only as a tool for the critical analysis of small street networks. In this capacity it could be a useful technique for planning operational controls for freeway-major arterial systems.

The concept of the optimum distribution of traffic is one which might well be applied to the general traffic assignment problem. It appears that the algorithmic technique could be developed to permit traffic assignment to urban street systems on a optimum distribution basis.

Recommendations

The following recommendations for additional research seem pertinent:

1. The development of the algorithmic technique for solving the network problem in a much faster manner appears to have much promise. Studies which would compare the results of multi-copy solutions to algorithmic

solutions and lead to the development of a satisfactory assignment technique seem highly warranted.

2. After development of the algorithmic technique so that large networks could be studied with a reasonable amount of machine time it would be desirable to study the optimum distribution of traffic over actual systems. To do this it would be necessary to break a system up into links and nodes and to arrive at the origin-destination requirements for this network.

REFERENCES

1. Berry, D. S., "Evaluation of Techniques for Determining Overall Travel Time", Proceedings of the Highway Research Board, pp. 429-440, 1952.

2. Blumentritt, Charles W., "Algorithmic Techniques For Traffic Network Analysis," Unpublished Report, Texas A&M University, 1963.

3. Brokke, G. E., "Assigning Traffic to a Highway Network", Public Roads, pp. 77-87, 1958.

4. Brown, R. M., "Expressway Route Selection and Vehicular Usage", Bulletin No. 16, Highway Research Board, pp. 12-21, 1948.

5. Bureau of Public Roads, "Electronic Computer Program for Assignment of Traffic to Street and Freeway Systems", Division of Development, Bureau of Public Roads, 1959.

6. Carroll, J. Douglas, Jr., "A Method of Traffic Assignment to an Urban Network", Bulletin 224, Highway Research Board, pp. 64-71.

7. CEIR Inc., LP/90 Usage Manual - A Comprehensive Computing System for Linear Programming, July, 1961.

8. Charnes, A., and Cooper, W. W., "Multicopy Traffic Network Models", Proceedings of Symposium on Theory of Traffic Flow, General Motors Corporation, pp. 85-96, 1959.

9. Charnes, A., and Cooper, W. W., "External Principles for Simulating Traffic Flow in a Network", Proceedings National Academy of Science, 1958, pp. 201-204.

10. Charnes, A., Cooper, W. W., and Henderson, Introduction to Linear Programming, John Wiley & Sons, Inc., New York, 1953.

11. Charnes, A., and Cooper, W. W., Management Models and Industrial Applications of Linear Programming, Volume II, John Wiley and Sons, Inc., pp. 785-796.

12. Charnes, A., and Lemke, C. E., "A Modified Simplex Method For Control of Round-off Error in Linear Programming," Proceedings of the Association for Computing Machinery, Pittsburgh, May 1952, pp. 97-98.

13. Chicago Area Transportation Study. Final Report, Volume II, Chicago, 1960.

14. Cleveland, D. E., "A Study of Highway Traffic Assignment", Doctoral Dissertation, A&M College of Texas, 1962.

15. Covault, D. O., and Roberts, R. R., "The Influence of On-Ramp Spacing on Traffic Flow on the Atlanta Freeway and Arterial Street System", Paper presented at the 42nd Annual Highway Research Board Meeting, 1963.

16. Dantzig, C. B., "On the Shortest Route Through a Network", Management Science, 1960, pp. 187-190.

17. Dantzig, G. B., "Maximization of a Linear Function of Variables Subject to Linear Inequalities", Chapter XXI of Koopmans (27).

18. DeRose, F., "The Development and Evaluation of the John C. Lodge Freeway Traffic Surveillance and Control Project", presented to the Annual Meeting of the Highway Research Board, 1963.

19. Forbes, T. W., "Speed, Headway and Volume Relationships on a Freeway", Proceedings of the Institute of Traffic Engineers, pp. 103-126, 1951.

20. Fuller, Leonard E., Basic Matrix Theory, Prentice Hall, Inc., 1963.

21. Gass, Saul I., Linear Programming Methods and Applications, McGraw-Hill Book Company, Inc., 1958.

22. Gomory, R. E., and Hu , T. C., "An Application of Generalized Linear Programming to Network Flows", Research Report RC 319, International Business Machines Corporation, September, 1960.

23. Heanue, Kevin E., and Covault, Donald O., "The Development of Criteria for Closing a Freeway Ramp Using Linear Programming", Committee Report, Theory of Traffic Flow, Highway Research Board, January, 1962.

24. Irwin, N. A., and von Cube, H. G., "Capacity Restraint in Multi-Travel Mode Assignment Programs, Paper presented to the Origin and Destination Committee of the Highway Research Board, January, 1962.

25. Irwin, N. A., Dodd, Norman and von Cube, H. G., "Capacity Restraint in Assignment Programs", Bulletin 297, Highway Research Board, pp. 109-127.

26. Keese, C. J., Pinnell, C., and McCasland, W. R., "A Study of Freeway Traffic Operations", Bulletin 235, Highway Research Board, pp. 73-132.

27. Koopmans, T. C., Editor, "Activity Analysis of Production and Allocation", Cowles Commission Monograph 13, John Wiley & Sons, Inc., New York, 1951.

28. Martin, B. V., Memmott, F. W., and Bone, A. J., Principles and Techniques of Predicting Future Demand for Urban Area Transportation, Research Report No. 38, Joint Highway Research Project, Massachusetts Institute of Technology, 1961.

29. May, A. D., Athol, P. J., Parker, W. L. and Rudden, J. B., "Development and Evaluation of Congress Street Expressway Pilot Detection System", presented at the Annual Meeting of the Highway Research Board, 1963.

30. McCracken, Daniel D., A Guide to FORTRAN Programming, John Wiley and Sons, Inc., New York, 1961.

31. Mertz, William L., "Review and Evaluation of Electronic Computer Traffic Assignment Programs", Bulletin 297, Highway Research Board, pp. 94-105.

32. Moore, E. F., "The Shortest Path Through a Maze", Proceedings of an International Symposium on the Theory of Switching Part II, April, 1957.

33. Moskowitz, K., "Research on Operating Characteristics of Freeways", Proceedings of the Institute of Traffic Engineers, pp. 85-110, 1956.

34. Pinnell, Charles and Satterly, G. T., "Systems Analysis Technique for the Evaluation of Arterial Street Operation", Paper presented at the Annual Meeting of the American Society of Civil Engineers, Detroit, Michigan, October, 1962.

35. Pinnell, Charles, "Development of a Linear Programming Technique for a Systems Analysis Study of Arterial Street Operation", Doctoral Dissertation, Texas A&M University, 1964.

36. Pollack, M., "The Maximum Capacity Through a Network", Operations Research, 1960, pp. 733-736.

37. Saaty, Thomas L., Mathematical Methods of Operations Research McGraw-Hill Book Company, Inc., 1959.

38. Schneider, M., "Traffic Distribution on a Road Network", Chicago Area Transportation Study Publication 66538, May, 1961.

39. Smock, Robert B., "An Iterative Assignment Approach to Capacity Restraint on Arterial Networks", paper presented at the Annual Meeting of the Highway Research Board, January, 1962.

40. Texas Highway Department, Highway Planning Survey, Gulf Freeway Traffic Survey, Houston, Texas, 1951.

41. Tomazinis, A. R., "A New Method of Trip Distribution in an Urban Area", Paper Number 15, Penn Jersey Transportation Study, 1962.

42. Voorhees, A. M., "A General Theory of Traffic Movement", Proceedings Institute of Traffic Engineers, 1955.

43. Wattleworth, Joseph A., "Peak-Period Control of a Freeway System-Some Theoretical Considerations", Report 9, Expressway Surveillance Project, Illinois Department of Highways, August, 1963.

44. Wattleworth, Joseph A., and Shuldiner, Paul W., "The Inclusion of Left Turns in Traffic Prediction Models of the Charnes-Cooper Type", Paper presented at the American Society of Civil Engineers Annual Meeting, October, 1962.

45. Webb, G., and Moskowitz, K., "California Freeway Capacity Study - 1956", Proceedings of the Highway Research Board, Volume 36, pp. 587-642, 1957.

A P P E N D I C E S

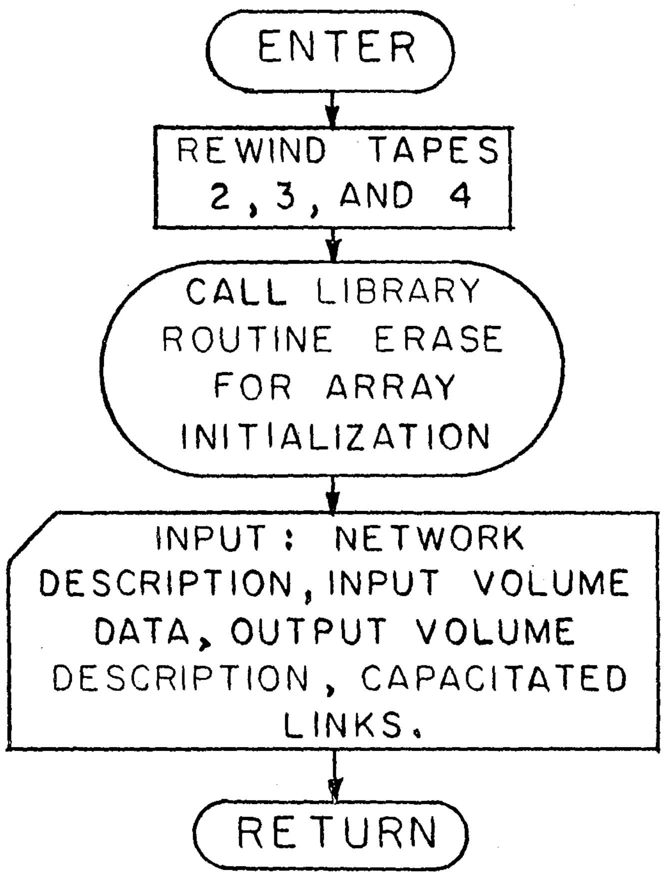
CALLING SEQUENCE:

CALL START (NOV, K, NOKAP)

NOV = NUMBER OF LINKS

K = NUMBER OF COPIES

NOKAP = NUMBER OF CAPACITATED LINKS.



FLOW DIAGRAM
SUBROUTINE START

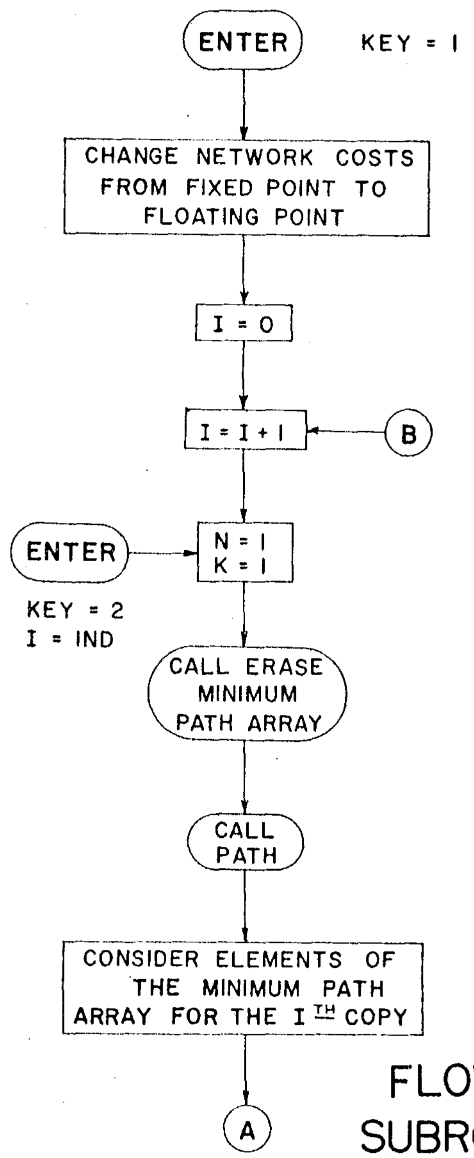
FIGURE 17

CALLING SEQUENCE:

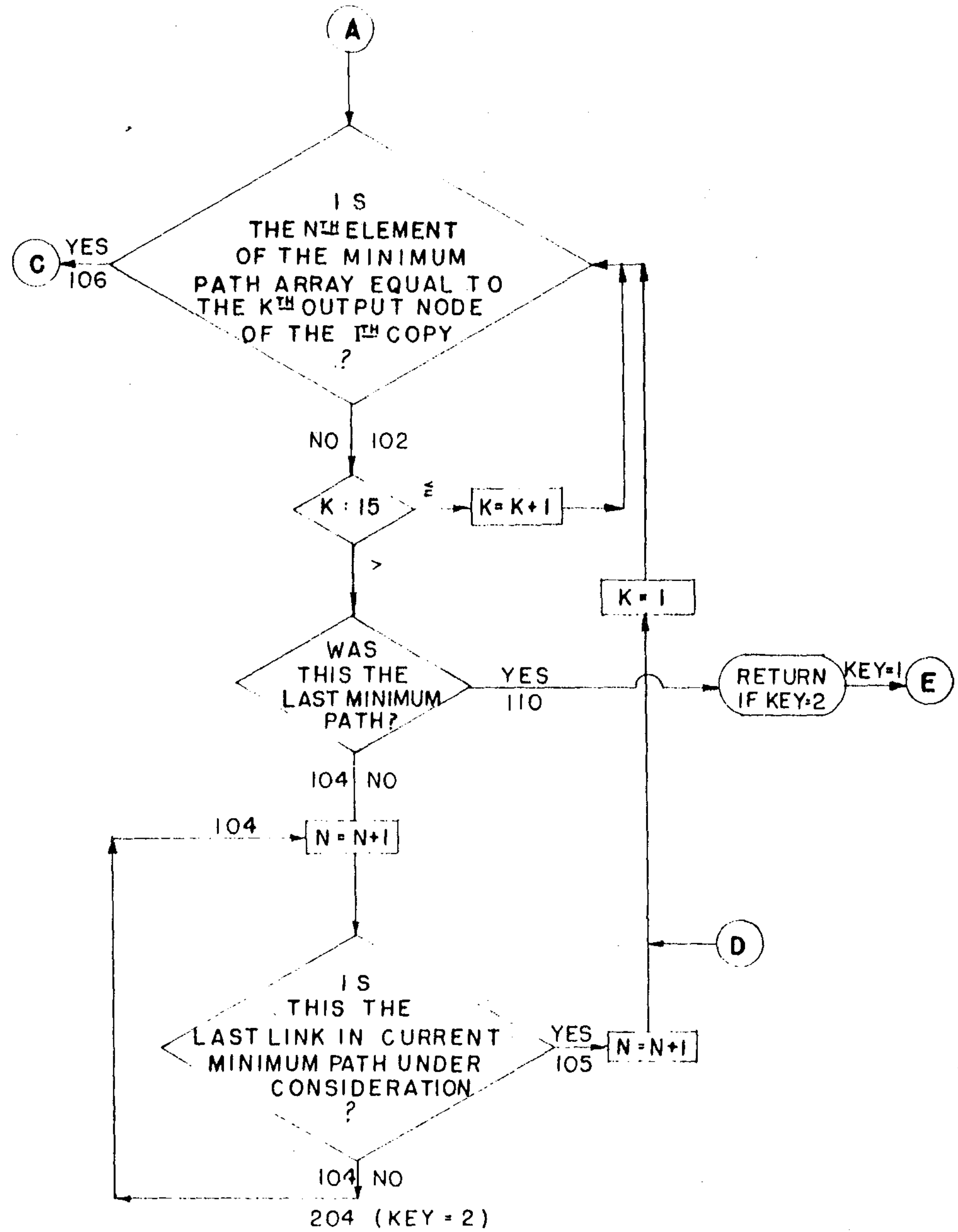
CALL TREE (KOPY, KEY, IND, LINK, ND)

KOPY = NUMBER OF COPIES
KEY = 1 OR 2, WHICH SPECIFIES ENTRY POINT
IND = COPY NUMBER UNDER CONSIDERATION
LINK = NUMBER OF LINKS
ND = NUMBER OF NODES MINUS TWO

KEY = 1 BUILDS MINIMUM PATHS FOR THE INITIAL BASIS
KEY = 2 BUILDS THE MINIMUM PATH FOR THE I TH COPY.

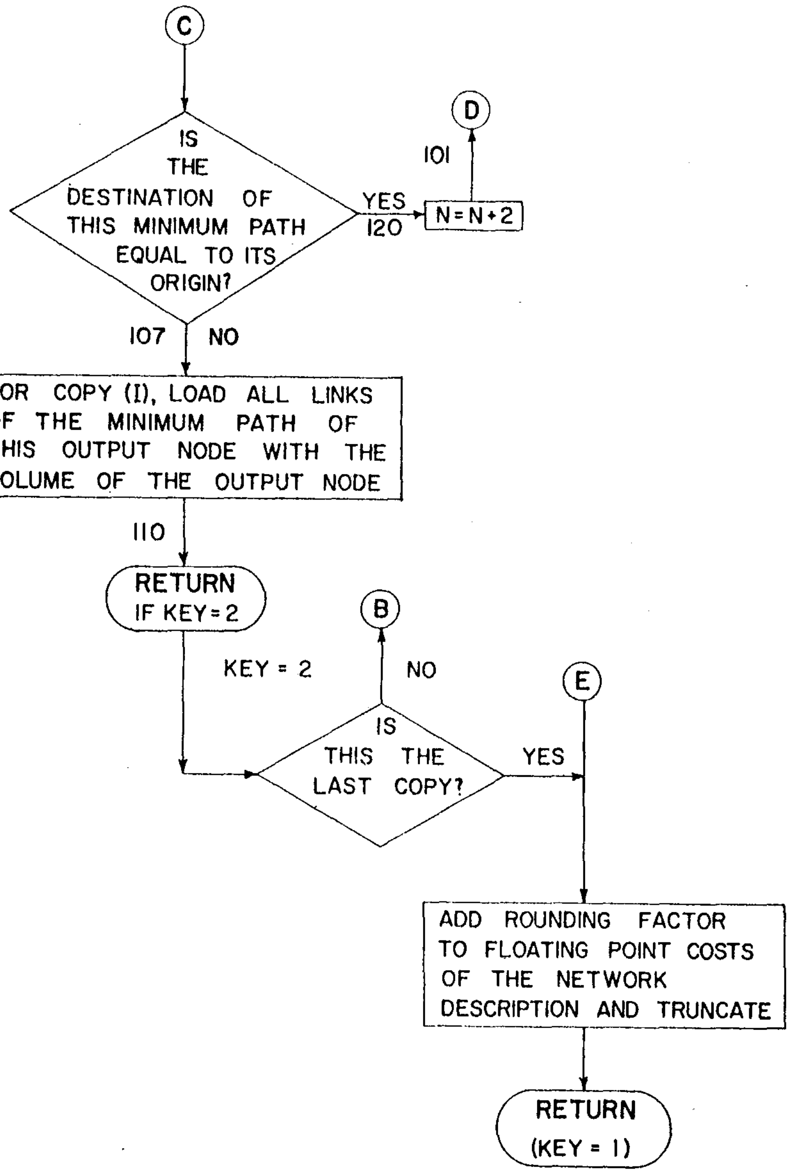


FLOW DIAGRAM
SUBROUTINE TREE
FIGURE 18-A

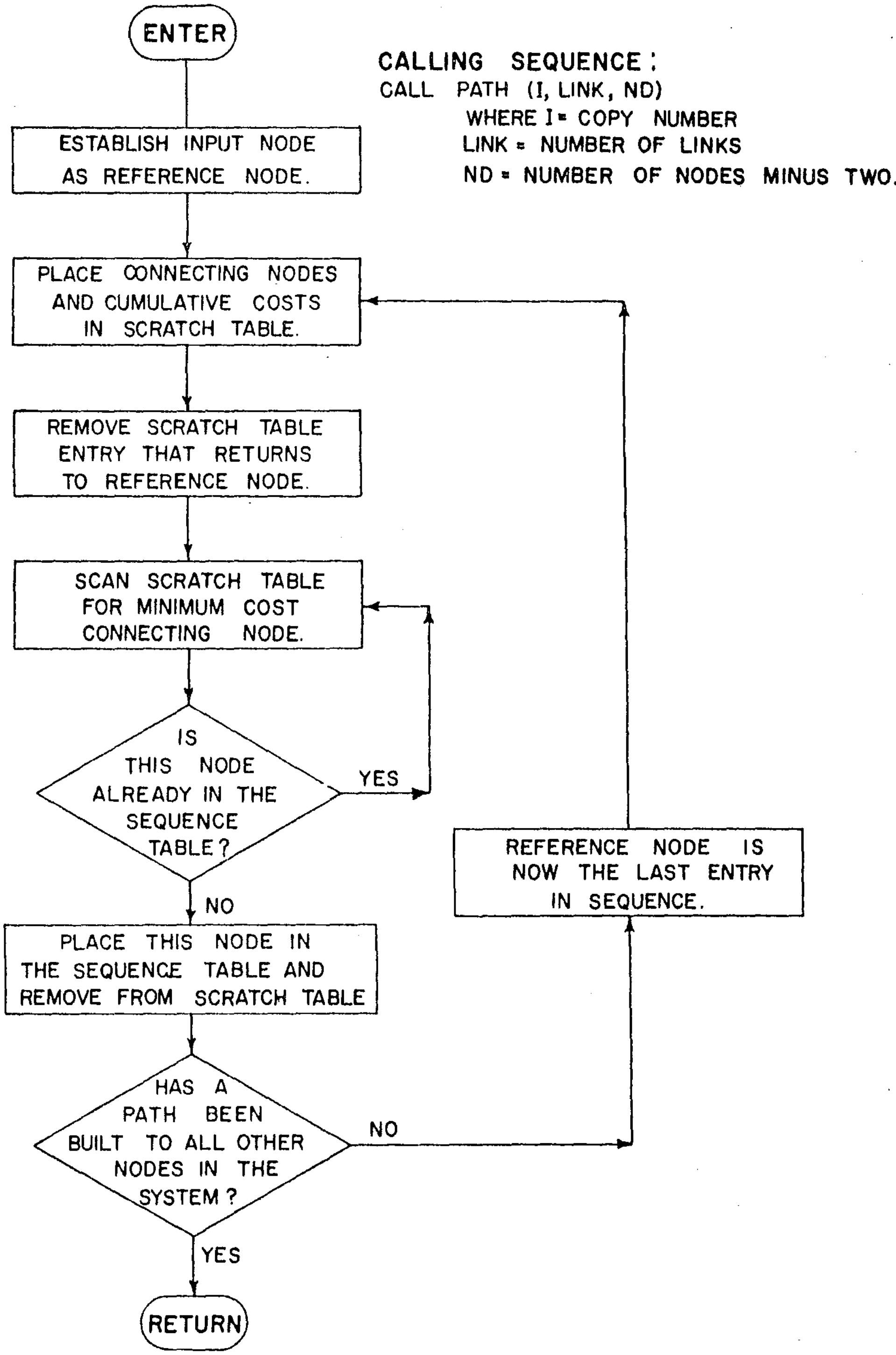


FLOW DIAGRAM
SUBROUTINE TREE

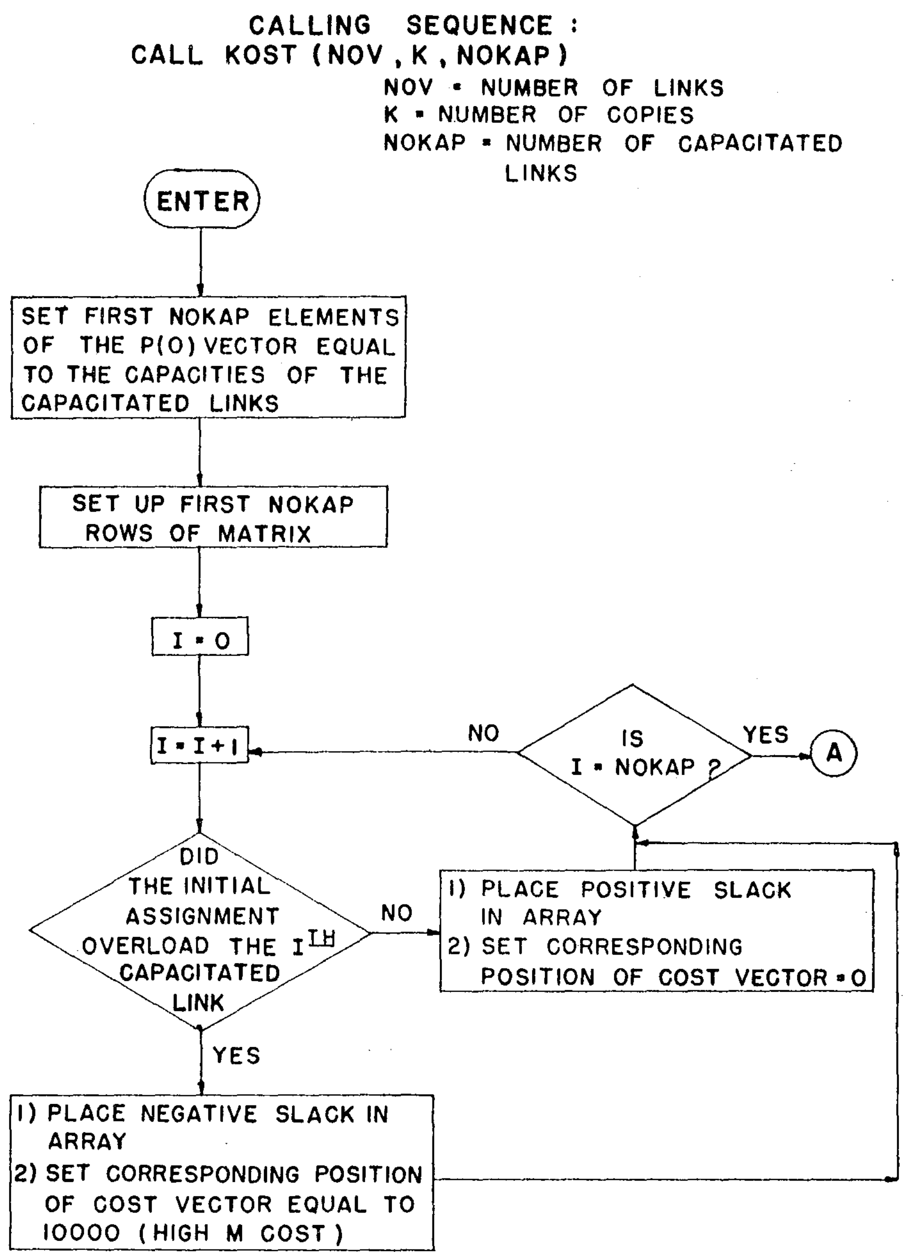
FIGURE 18-B



FLOW DIAGRAM
SUBROUTINE TREE
FIGURE 18-C

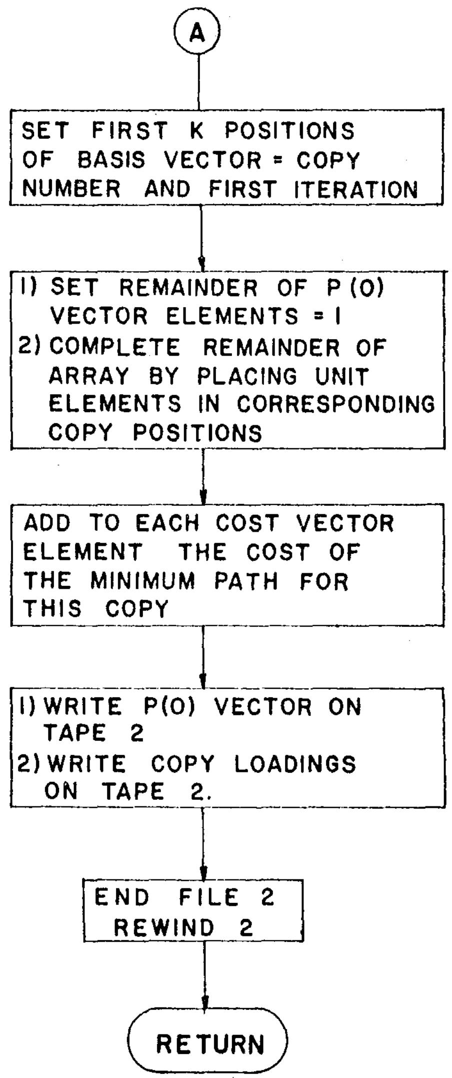


FLOW DIAGRAM
SUBROUTINE PATH
FIGURE 19



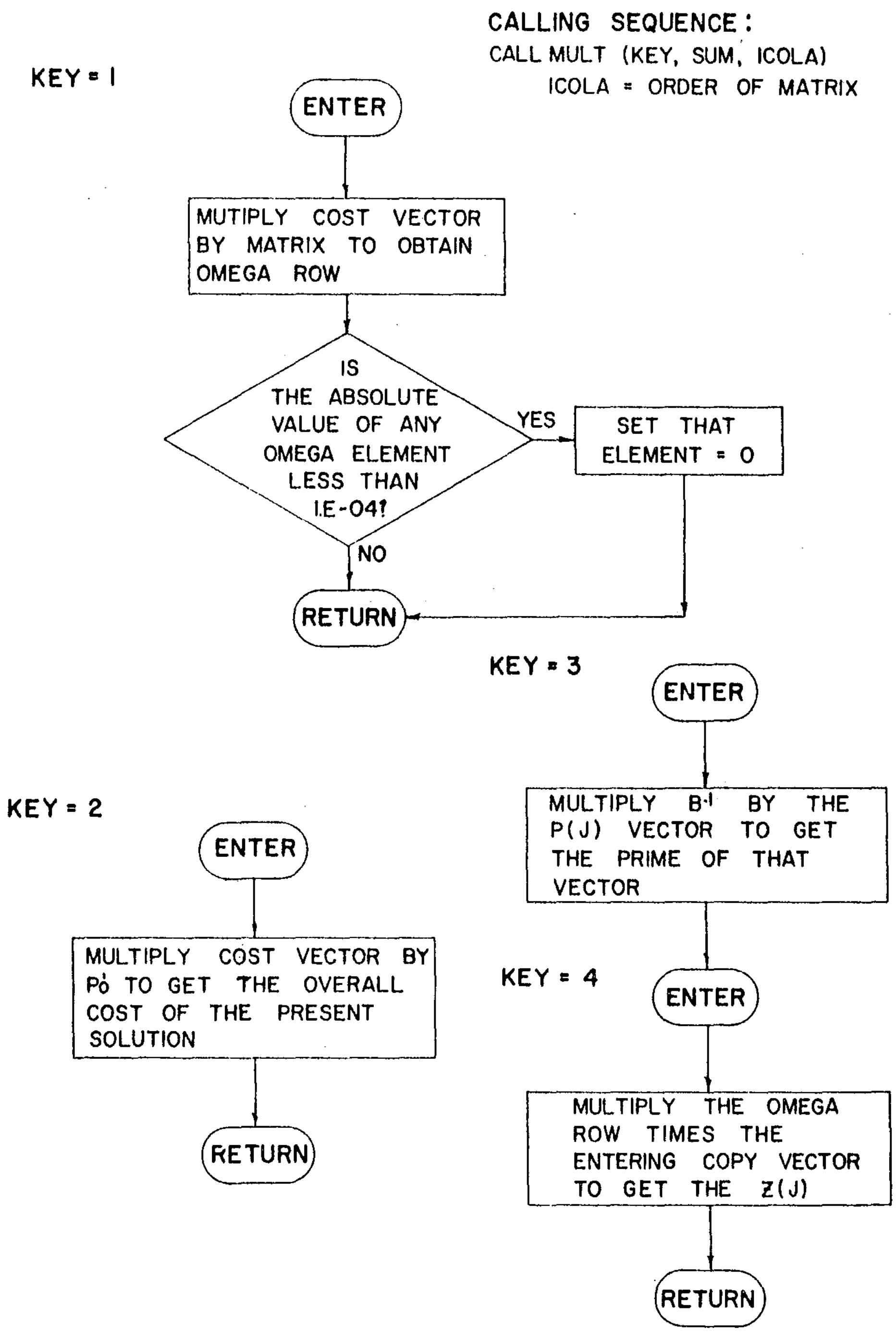
FLOW DIAGRAM
SUBROUTINE KOST

FIGURE 20-A



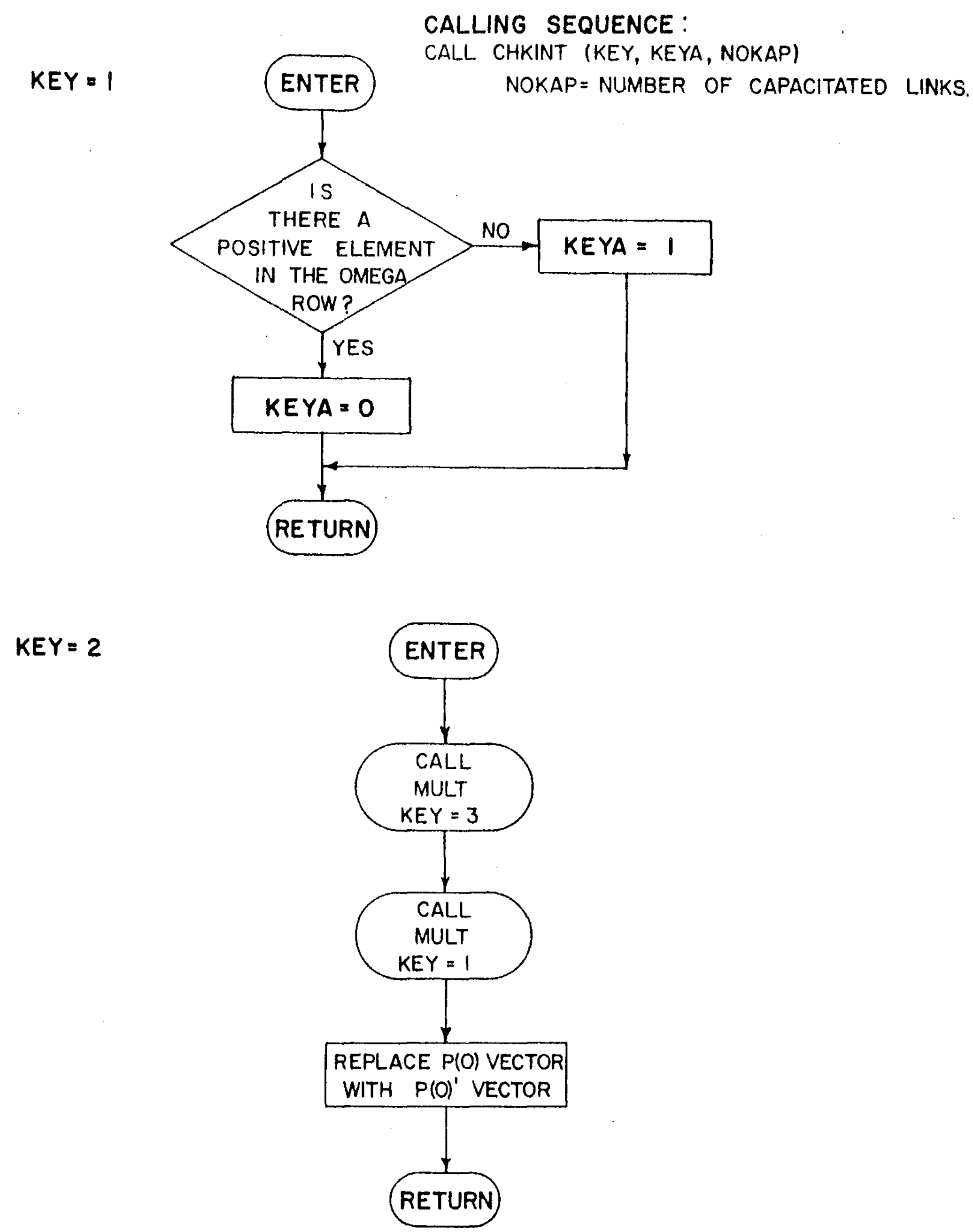
FLOW DIAGRAM
SUBROUTINE KOST

FIGURE 20-B



FLOW DIAGRAM
SUBROUTINE MULT

FIGURE 21



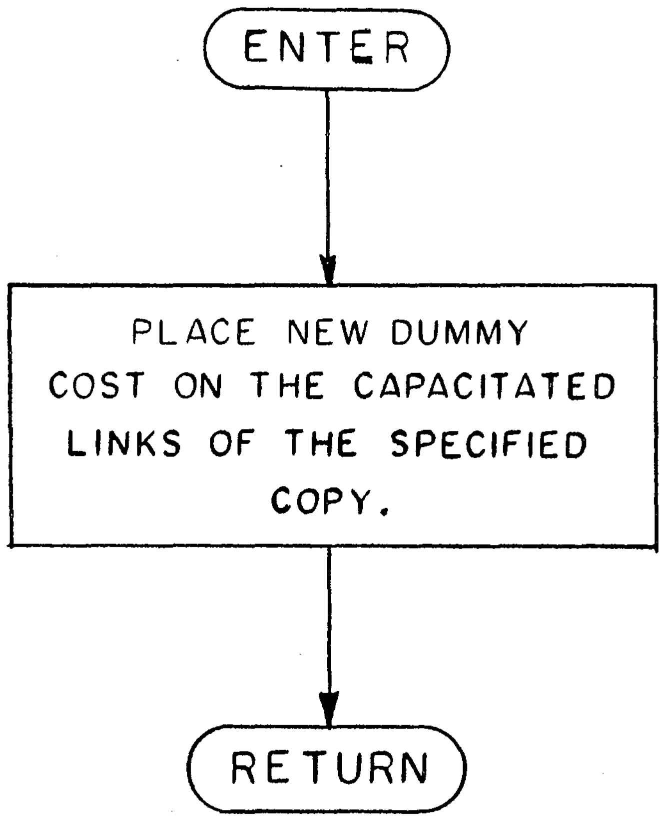
FLOW DIAGRAM
SUBROUTINE CHKINT

FIGURE 22

CALLING SEQUENCE

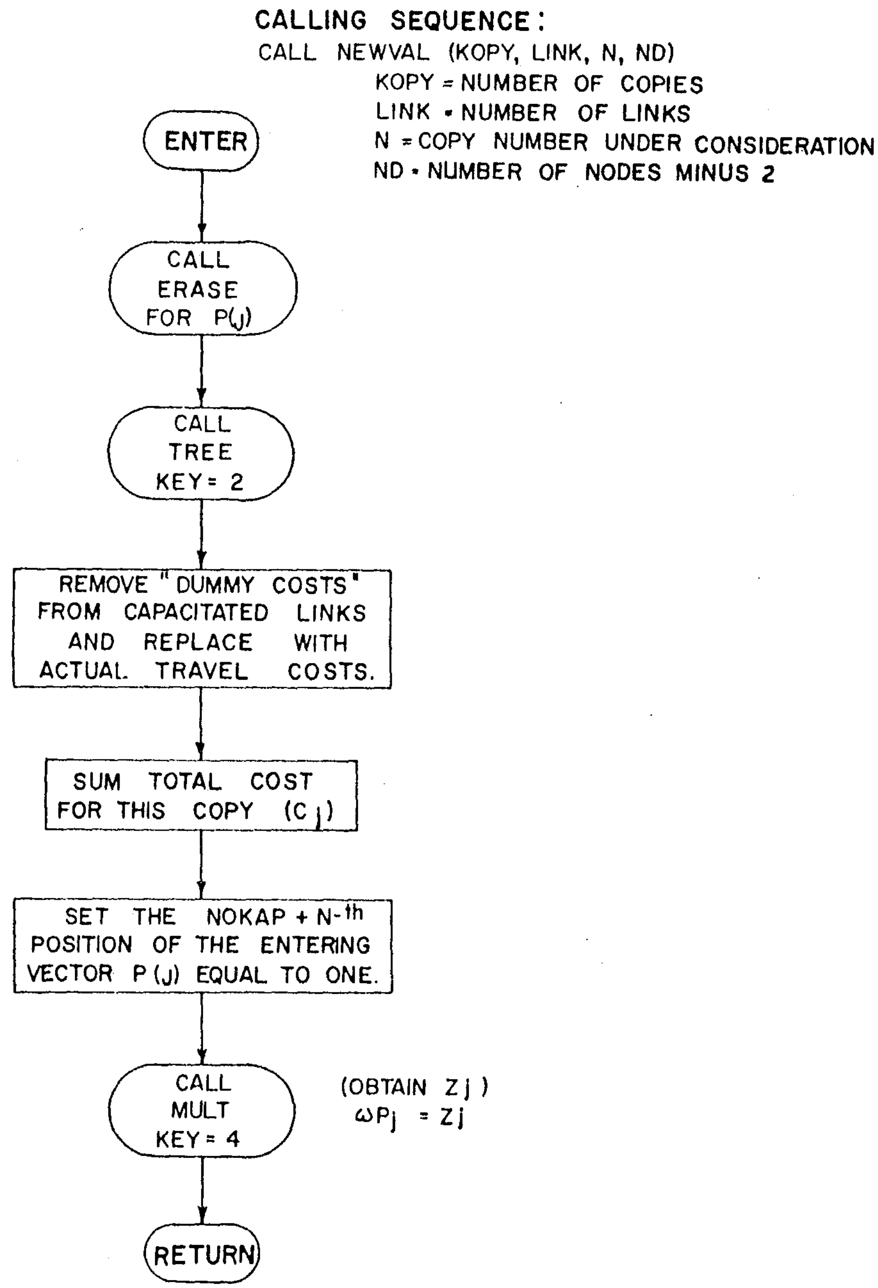
CALL LOGIC (NOKAP)

NOKAP = NUMBER OF CAPACITATED LINKS.

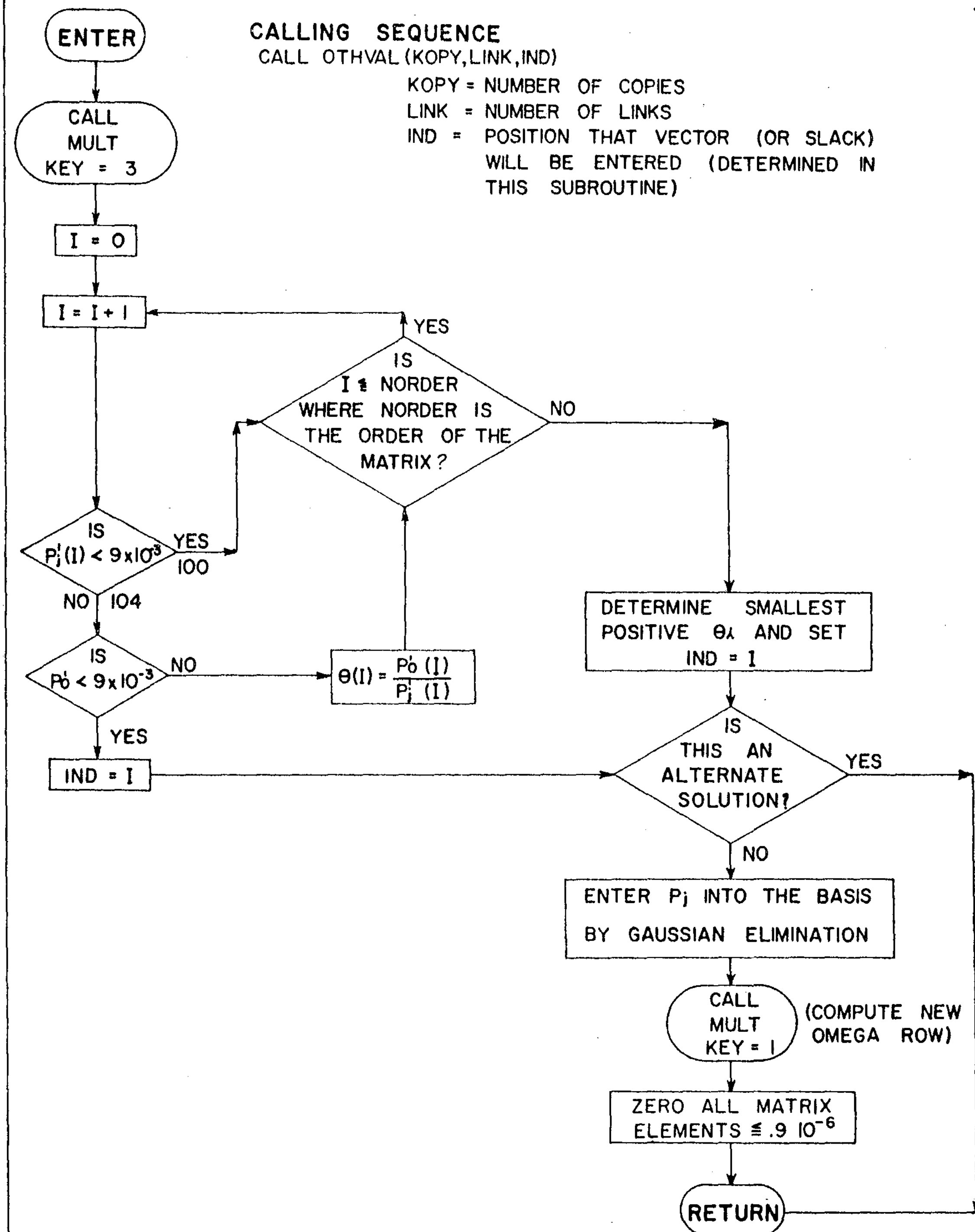


FLOW DIAGRAM
SUBROUTINE LOGIC

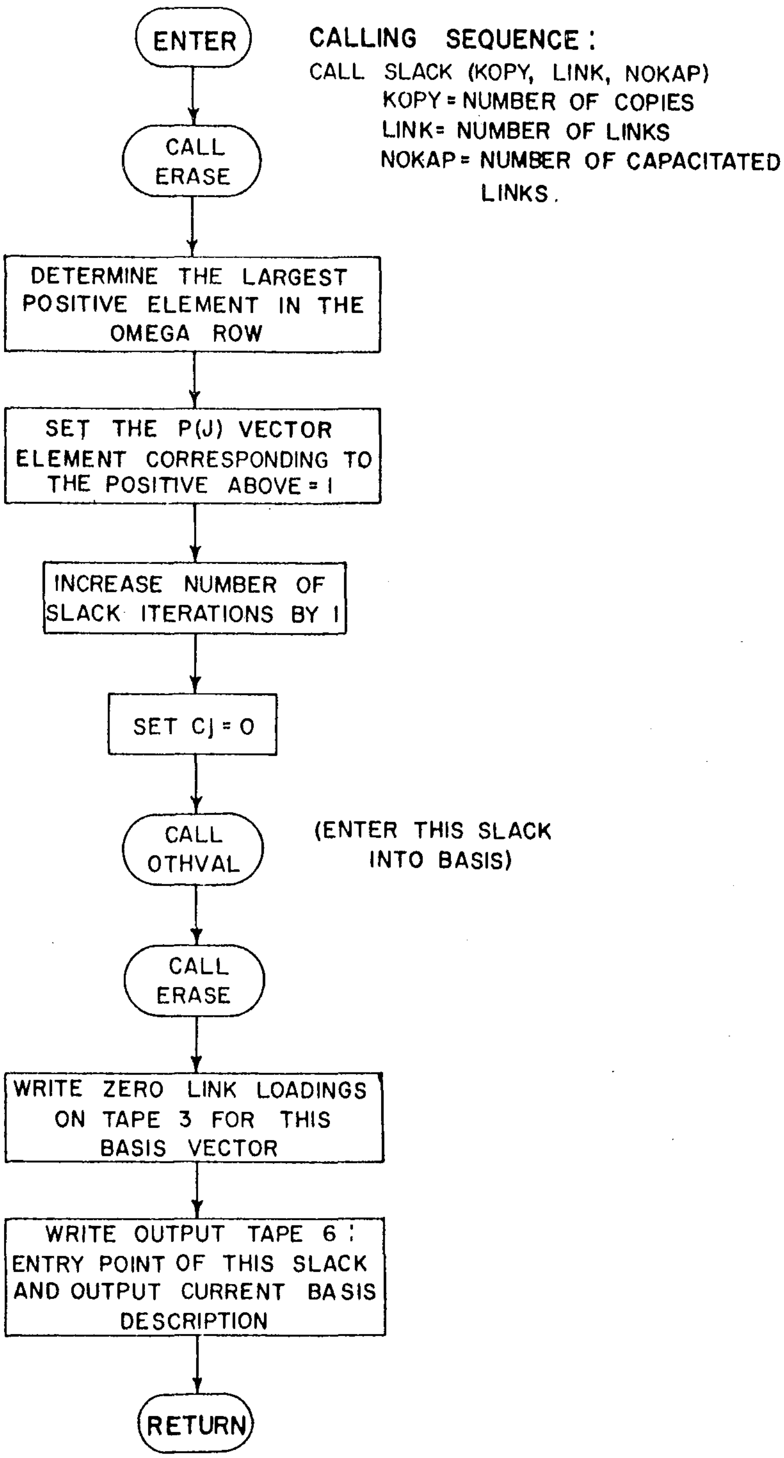
FIGURE 23



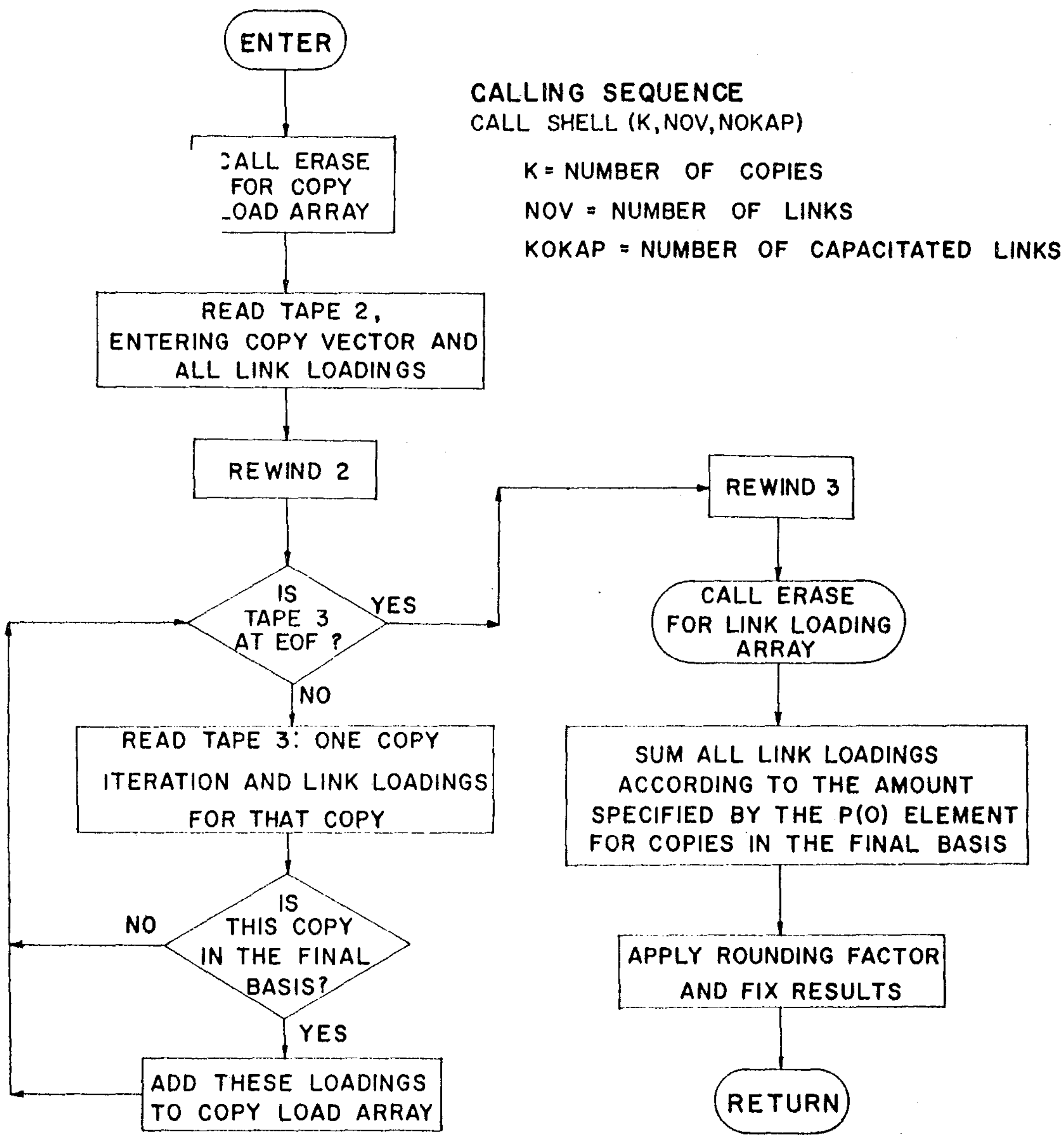
FLOW DIAGRAM
SUBROUTINE NEWVAL
FIGURE 24



FLOW DIAGRAM
SUBROUTINE OTHVAL
FIGURE 25



FLOW DIAGRAM
SUBROUTINE SLACK
FIGURE 26

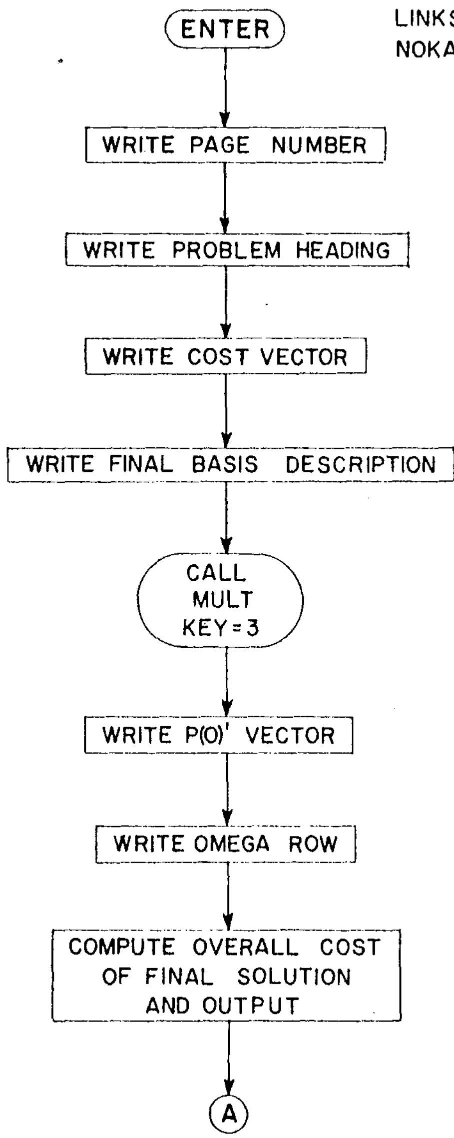


FLOW DIAGRAM
SUBROUTINE SHELL

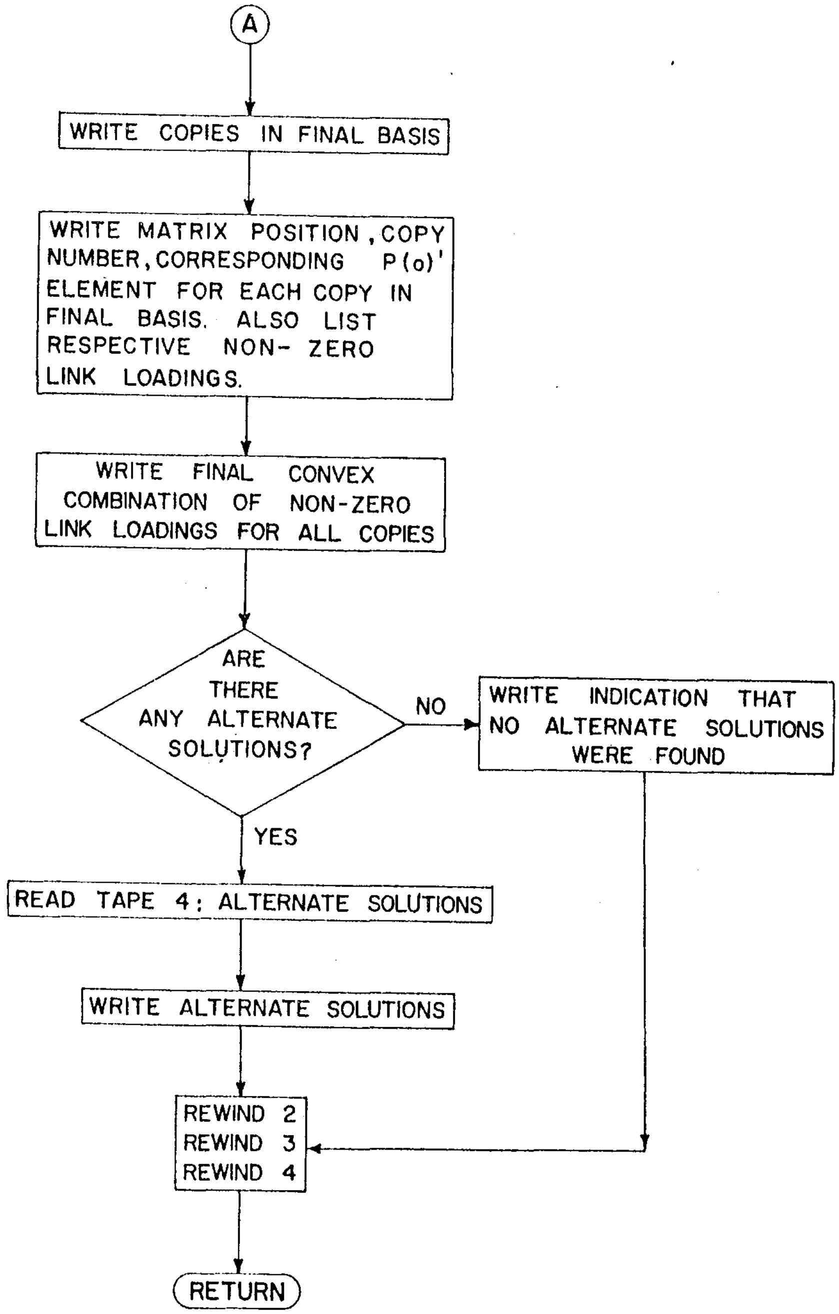
FIGURE 27

CALLING SEQUENCE
(KOPY ,LINK , NOKAP)
KOPY = NUMBER OF COPIES
LINKS = NUMBER OF LINKS
NOKAP = NUMBER OF CAPACITATED LINKS.

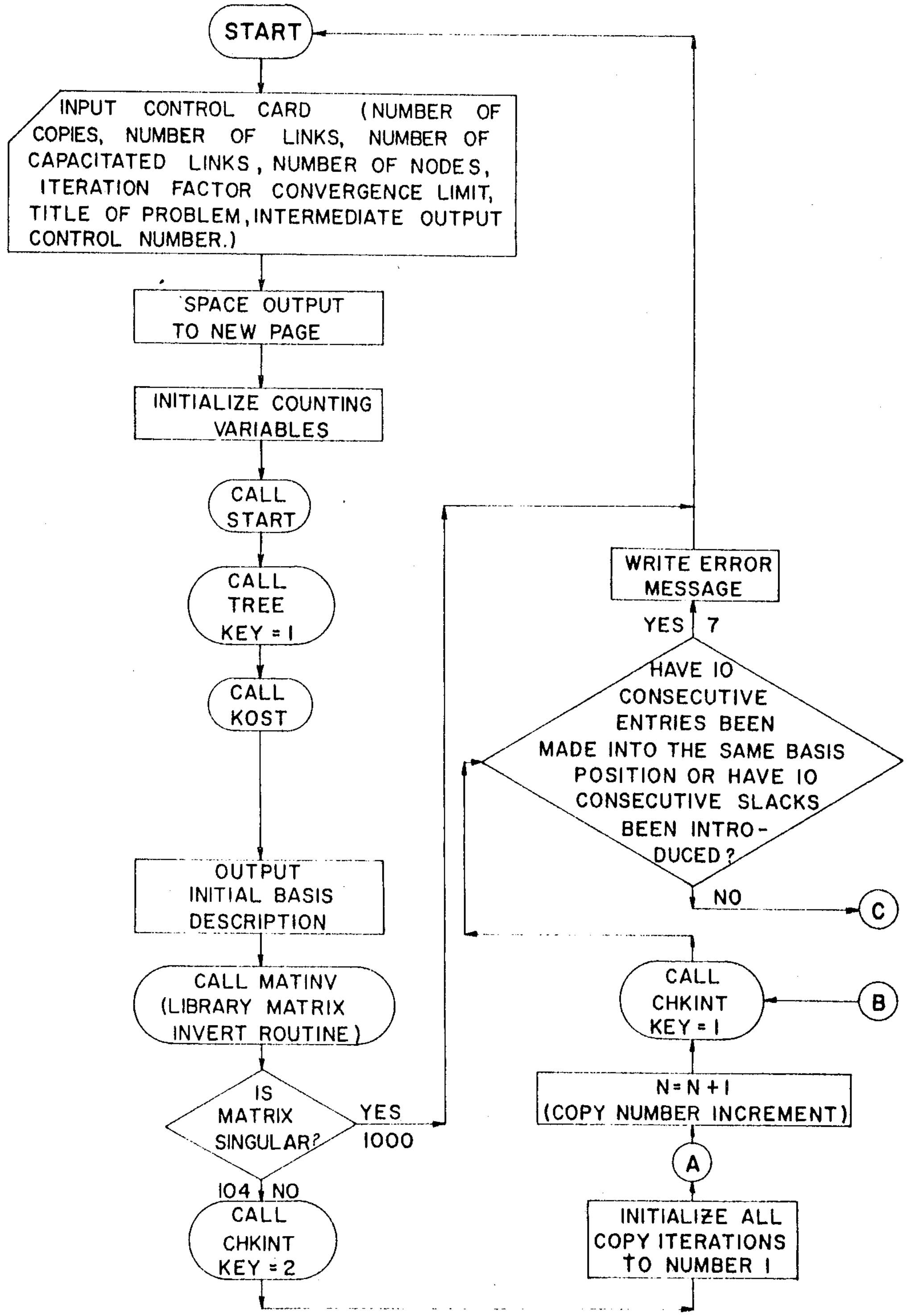
NOTE : ALL WRITE STATEMENTS REFER TO OUTPUT TAPE 6.



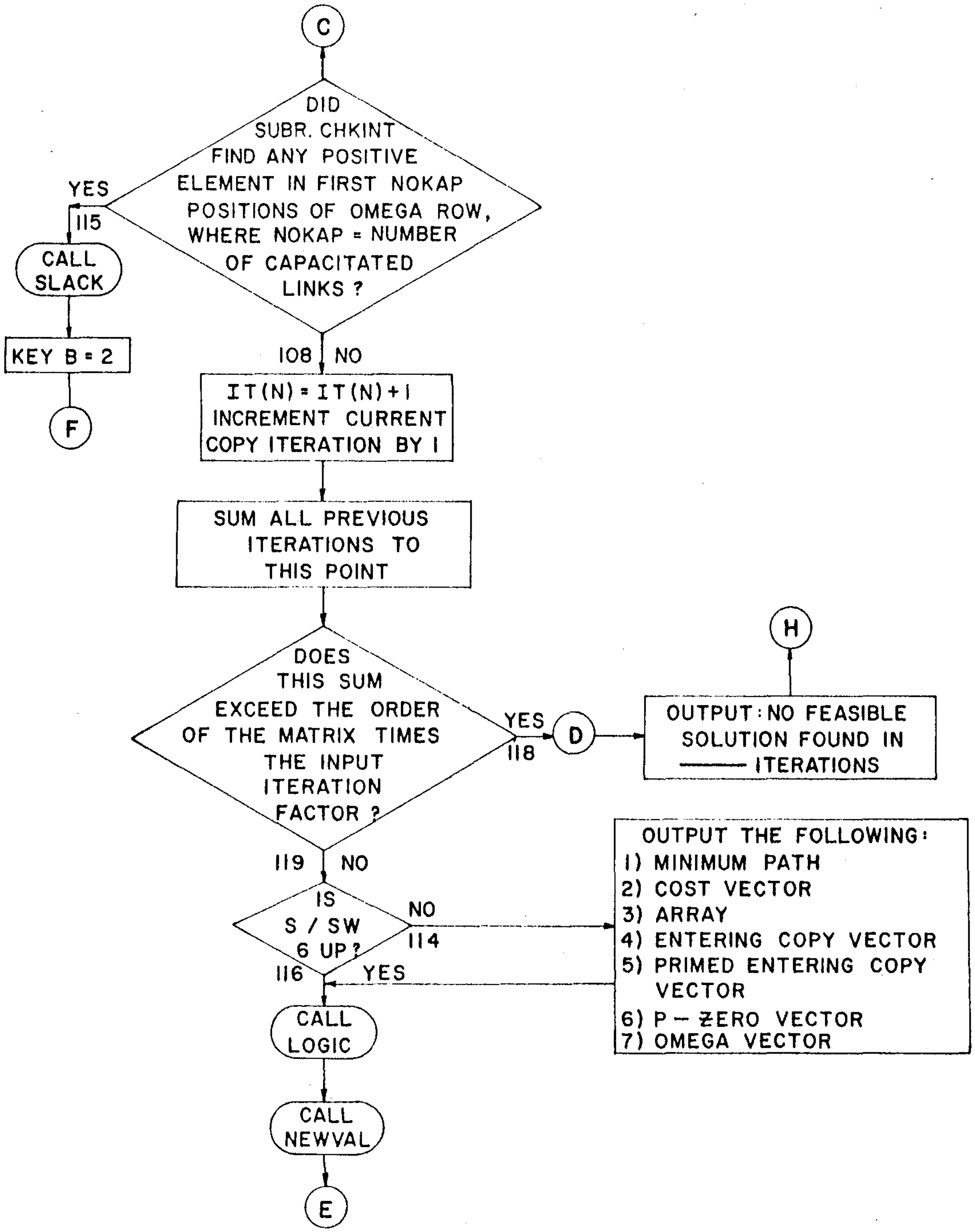
FLOW DIAGRAM
SUBROUTINE PRNT
FIGURE 28-A



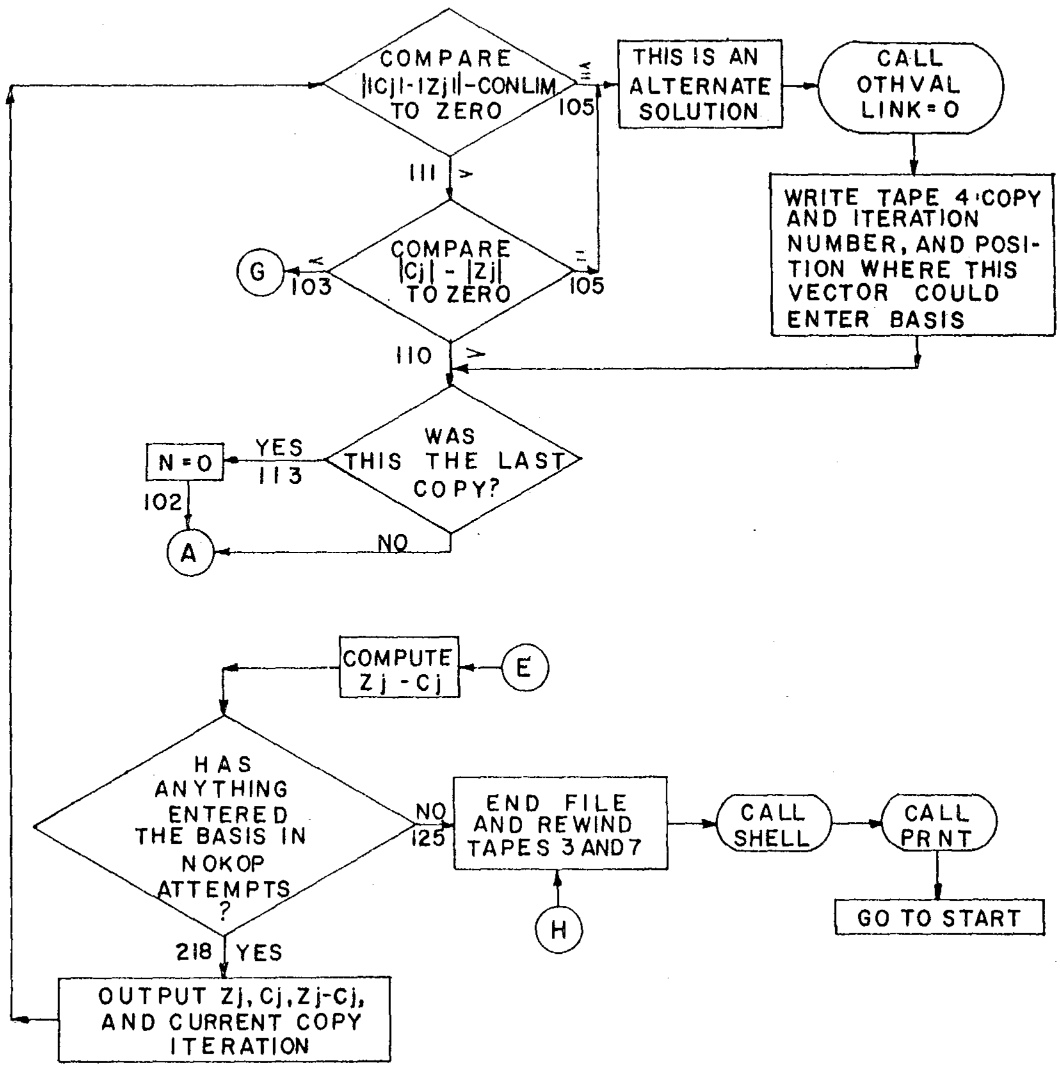
FLOW DIAGRAM
SUBROUTINE PRNT
FIGURE 28-B



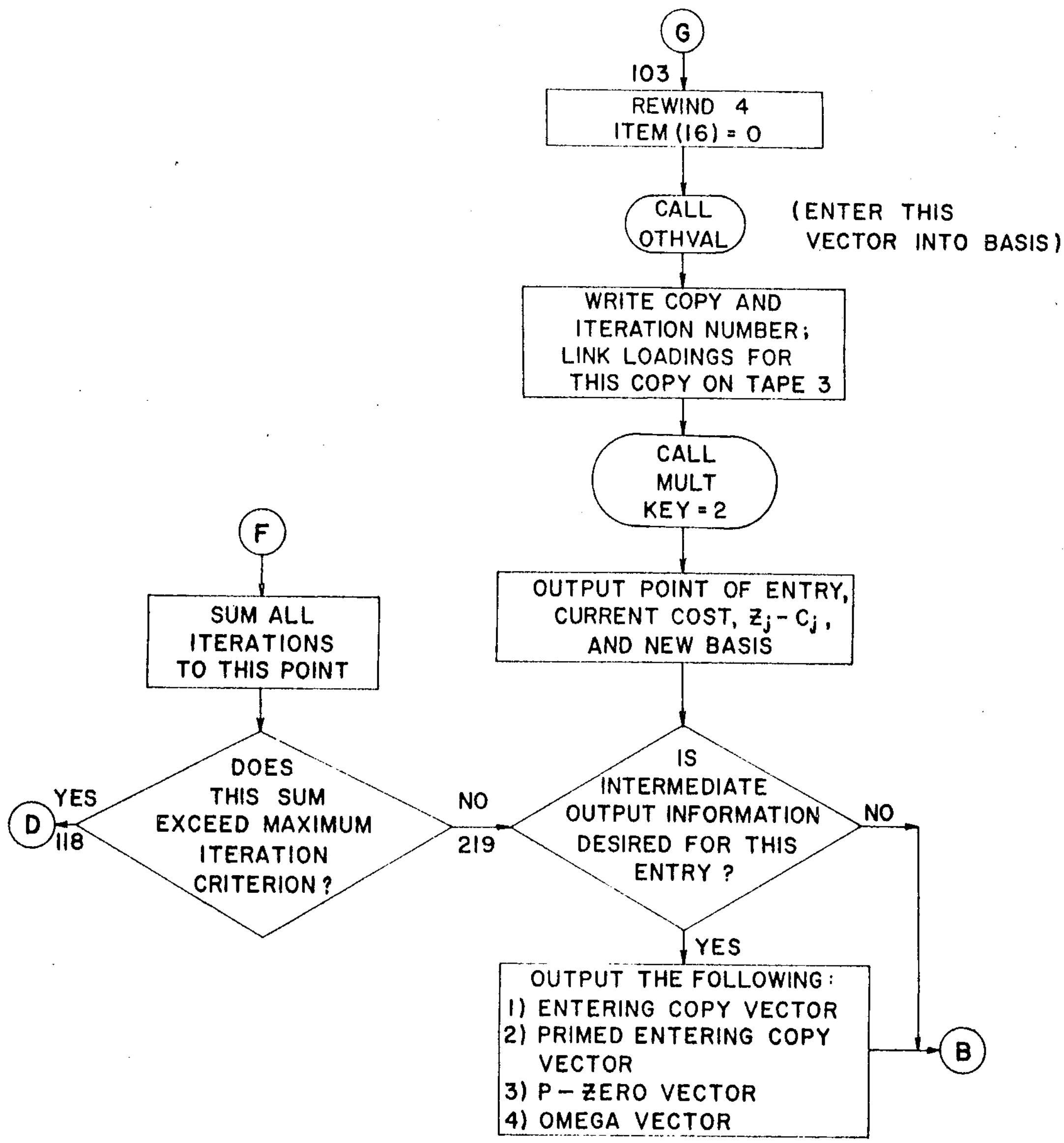
FLOW DIAGRAM
MAIN CONTROL PROGRAM
FIGURE 29-A



FLOW DIAGRAM
MAIN CONTROL PROGRAM
FIGURE 29-B



FLOW DIAGRAM
MAIN CONTROL PROGRAM
FIGURE 29-C



FLOW DIAGRAM
MAIN CONTROL PROGRAM
FIGURE 29-D

PUBLICATIONS

Project 2-8-61-24
Freeway Surveillance and Control

1. Research Report 24-1, "Theoretical Approaches to the Study and Control of Freeway Congestion" by Donald R. Drew.

2. Research Report 24-2, "Optimum Distribution of Traffic Over a Capacitated Street Network" by Charles Pinnell.