ITERATIVE SOLUTION OF NETWORK FLOW PROBLEMS BY DYNAMIC PROGRAMMING

by

DEEPAK KESHAV PAI

B. E. M. E., University of Poona, India, 1963
M. Tech. M. E., Indian Institute of Technology,
Bombay, India, 1965

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

Approved by:

Frank A. Tillman
Major Professor

TABLE OF CONTENTS

1. INTRODUCTION 1
2. THE TRAFFIC ASSIGNMENT PROBLEM 3
2.1 Definitions. 4
2.2 Formulation of Traffic Assignment Problem. 7
2.3 Example Problems 9
3. GENERAL PIPELINE PROBLEM 10
3.1 Example Problems 11
4. LITERATURE SURVEY. 12
5. DYNAMIC PROGRAMMING. 14
6. THE SOLUTION PROCEDURE 16
6.1 The Solution of a N x M Traffic Assignment 16
Problem by Dynamic Programming 16
6.2 Example 1. 20
6.3 Example 2. 29
6.4 Example 3. 35
6.5 Example 4. 38
7. SUMMARY. 43
8. CONCLUSION 44
9. ACKNOWLEDGMENTS. 45
10. REFERENCES 46
11. APPENDIX I 48
12. APPENDIX II 49
13. APPENDIX III 58
14. APPENDIX IV 66

1. INTRODUCTION

Many problems of economic and physical origin form important classes of network problems. Traffic assignment and pipeline problems constitute such classes of problems.

In the traffic assignment problem the streets form the network and the vehicles may enter or leave at any of the intersections. In this situation various types of traffic flows can take place. For example all the vehicles may enter at one point on the network and travel on different streets to some common destination. This is called a one way flow problem. If the vehicles enter and leave at two different points, traveling in two different directions, this is called a two way flow problem. Thus there are as many types of problems as there are directions of traffic flows. In each of these problems vehicles may enter and leave the network at any of the intersections.

An optimal path through a network can be selected based on a number of different objectives. It has been observed that more time is required to travel a street as the traffic volume increases. Thus in making a trip, the driver will tend to select a route which requires the minimum time. Thus the traffic assignment problems arise where vehicles are to be assigned on each street so that the total travel time for all drivers is a minimum.

Pipeline problems arise in a similar fashion as the traffic assignment problems. These occur when oil, gas, water or any fluid is collected from various reserves and transported to a number of destinations through a network of pipelines. It is assumed in these problems that the cost of transportation of the fluid increases as the volume increases. This problem becomes one of assigning the volume of fluid to be transported over each link so that the total cost of transportation is minimum.

The solution of the traffic assignment problem can be used to determine the deficiencies of the existing transportation system and to assist in the development of future transportation system. The solution of the pipeline problem can be used for determining the optimum utilization of the existing system and to evaluate alternate system proposals for the development of future systems.

2. THE TRAFFIC ASSIGNMENT PROBLEM

In this section, the traffic assignment problem considered is one of assigning the vehicles to the streets of a network, where the vehicles enter or leave at one or more points on the network and travel in the same or different directions and minimize the total travel time for all drivers.

Figure 1. represents a travel time volume relationship. The form of the equation is:

$$t = k_1 + k_2 \cdot V + k_3 \cdot (V/c)^r \tag{2.1}$$

where

t = link travel time in hours per vehicle

k_1 = constant representing travel time at free flow conditions

k_2, k_3 = empirically derived constants

V = link volume in vehicles per link per hour

c = link capacity in vehicles per link per hour

r = empirically derived exponent

The first term of equation (2.1) represents the travel time at free flow conditions. The second term serves to increase travel time as the link volume increases. The increase in travel time due to a unit increase in volume depends on the magnitude of the constant k_2. The first two terms of equation (2.1) represents the linear portion of the time-volume curves between the points A and B as shown in Figure 1. The third term represents the effect of congestion on the travel time for the facility under consideration. As the link volume nears capacity, the value of this term increases rapidly and at volumes beyond capacity ($V > c$) the travel time becomes so great that in effect the link has been closed for additional traffic. In Figure 1., the curve between B and C represents conditions of congestion and thus is the undesirable region for operation. Total travel time through each link is obtained by multiplying both the sides of equation (2.1) by the traffic volume V.

$$T = K_1 \cdot V + K_2 \cdot V^2 + K_3 \cdot \left(\frac{V}{c}\right)^r \cdot V \qquad (2.2)$$

A traffic assignment problem is illustrated in Figure 2. The following definitions and terms are given here to simplify the latter discussion of the mathematical formulation of the traffic assignment problem.

2.1 Definitions

1. Objective function: The function which is to be optimized. In this discussion it is the time function and it is to be minimized.
2. Zone Centroid: The place of trip origin or destination.
3. Node: The point where the segments of the streets system connect.
4. Link: The connection between two nodes which represent the segments of a street system.

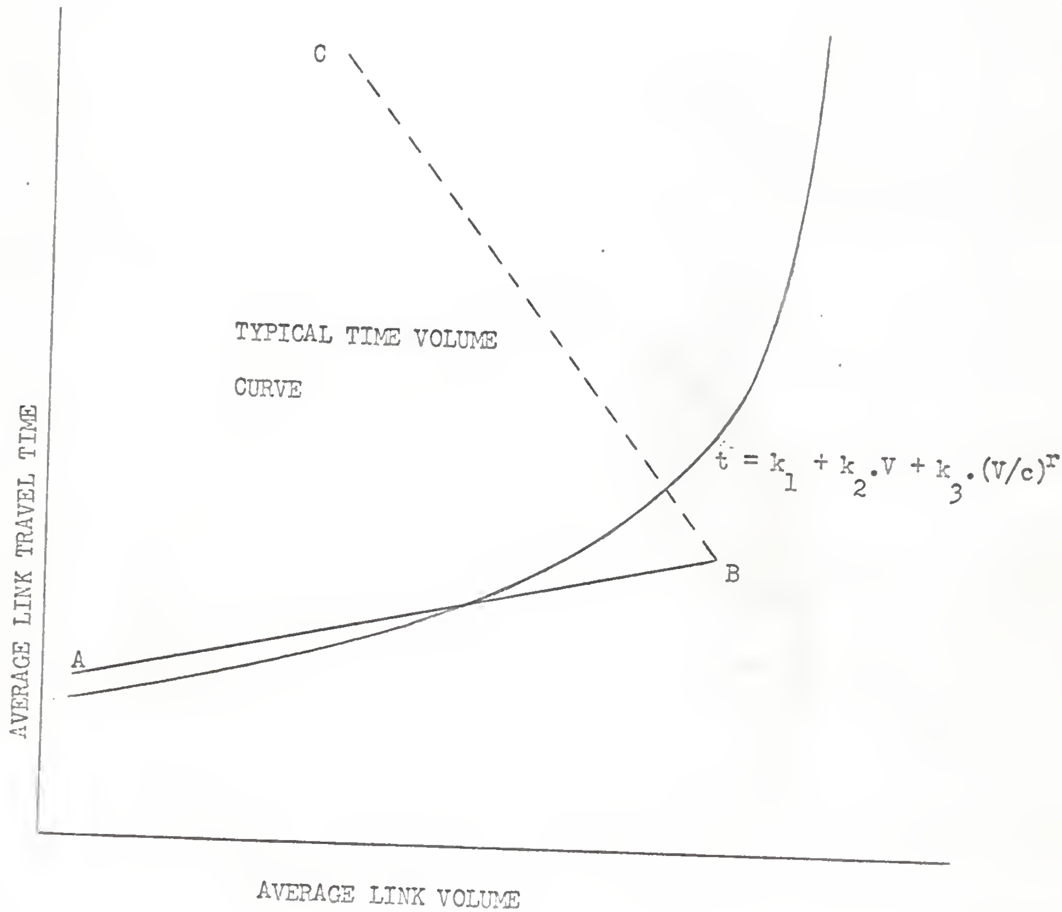


Figure 1. Typical Travel-Time Volume Relationship

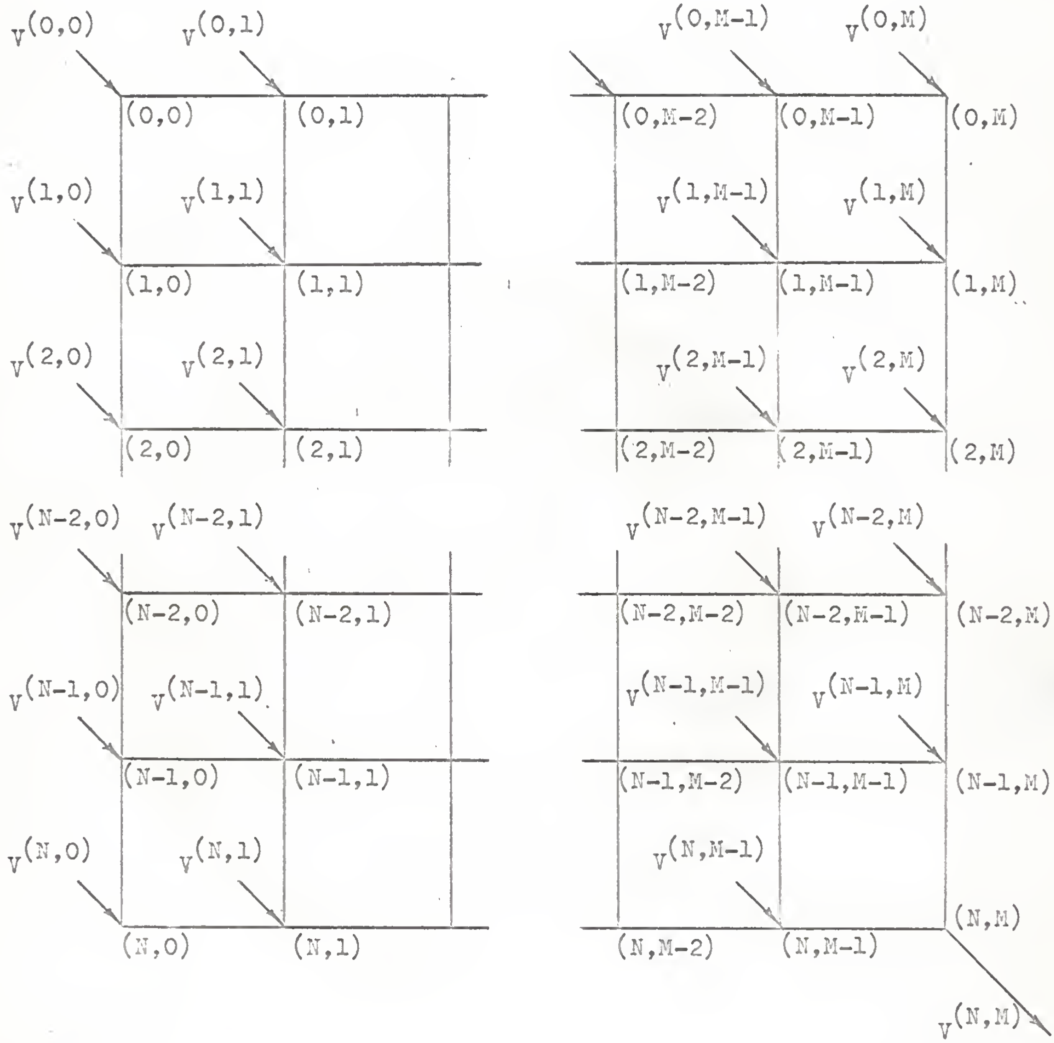


Figure 2. NxM network

5. Path: The series of connected links representing the trip route.

6. Network: The combination of all links and nodes.

2.2 Formulation of the Traffic Assignment Problem

Consider the network of streets as shown in Figure 2 where the following notations are used:

(n,m) = represents the nodes ($n = 0,1,2,\ldots N$; $m = 0,1,2,\ldots M$)

$V^{(n,m)}$ = the total number of vehicles entering at the node (n,m)

$Z^{(n,m)}$ = the total number of vehicles at the node (n,m)

$X_H^{(n,m)}$ = the total number of vehicles going in the horizontal direction from the node (n,m) towards node $(n,m+1)$

$X_V^{(n,m)}$ = the total number of vehicles going in the vertical direction from the node (n,m) towards node $(n+1,m)$

Using the above notations the percentage of the volume at the node (n,m) which travel in the horizontal direction can be expressed as:

$$P^{(n,m)} = X_H^{(n,m)} / Z^{(n,m)} \tag{2.3}$$

and consequently,

$$1 - P^{(n,m)} = X_V^{(n,m)} / Z^{(n,m)} \tag{2.4}$$

The last term of the equation (2.2), which is $K_3 \cdot \frac{V^{(r+1)}}{c^r}$, is insignificant at lower values of V and hence it can be neglected for small values of V. Thus the total time required to travel the network when this occurs is given by:

$$T = \sum_{m=0}^{M} \sum_{n=0}^{N} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (2.5)$$

with initial conditions that:

$$p^{(n,M)} = 0.0 \text{ and } X_H^{(n,M)} = 0.0, \ (n = 0,1,2..., N) \qquad (2.6)$$

$$p^{(N,m)} = 1.0 \text{ and } X_V^{(N,m)} = 0.0, \ (m = 0,1,2..., M) \qquad (2.7)$$

and where $K_{V1}^{(n,m)}$, $K_{V2}^{(n,m)}$ = the constants associated with the vertical streets from the node (n,m) to (n+1,m)

$K_{H1}^{(n,m)}$, $K_{H2}^{(n,m)}$ = the constants associated with the horizontal street from the node (n,m) to (n,m+1)

In summary then, the problem becomes one of minimizing T given by equation (2.5) by finding suitable values of $p^{(n,m)}$ ($n = 0,1,2,...,N$; $m = 0,1,2,...,M$) and satisfying the conditions given by equations (2.6) and (2.7).

2.3 Example Problems

1. The 3 x 3 network shown in Figure 4 is solved where $V(0,0)$ vehicles enter at node (0,0) and leave at node (3,3). The problem is to determine $P(n,m)$ (n = 0,1,2,3; m = 0,1,2,3) for the network which will minimize the total traveling time.

2. The 2 x 2 network shown in Figure 5 is solved where $V_{NW}(0,0)$ vehicles enter the network at node (0,0) from the Northwest and leave at node (2,2) and $V_{SW}(2,0)$ vehicles enter the network at the node (2,0) from the Southwest and leave at node (2,0). The problem is to determine $P_{SW}(n,m)$ and $P_{SW}(n,m)$ (n = 0,1,2; m = 0,1,2) which will minimize the total traveling time.

3. The 2 x 2 network shown in Figure 7 is solved where $V_{NW}(0,0)$ vehicles enter at node (0,0) from the Northwest and $V_{NW}(2,1)$, $V_{NW}(2,2)$ and $V_{NW}(1,2)$ vehicles leave at node (2,1), (2,2) and (1,2) respectively. Similarly, $V_{SW}(2,0)$ vehicles enter the netowrk at (0,2) and $V_{SW}(0,1)$, $V_{SW}(0,2)$ and $V_{SW}(1,2)$ leave at node (0,1), (0,2) and (1,2) respectively. The problem is to determine $P_{NW}(n,m)$ and $P_{SW}(n,m)$ (n = 0,1,2; m = 0,1,2) which will minimize the total traveling time.

3. GENERAL PIPELINE PROBLEM

The pipeline problem is similar to the traffic assignment problem with the exception that the flow of fluid is considered as a continuous function. The fluid can be fed in or tapped at any node. The cost of transporting fluid through a section of pipe can be represented by the following equation:

$$C = K_1 \cdot v + K_2 \cdot v^2 \tag{3.1}$$

where K_1, K_2 = the constants for the pipeline to be experimentally determined

v = the quantity of the fluid flowing through the pipe under consideration.

Thus the total cost for N X M pipeline network is given by:

$$C = \sum_{m=0}^{M}\sum_{n=0}^{N} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \tag{3.2}$$

where $K_{V1}^{(n,m)}$, $K_{V2}^{(n,m)}$ = the constants for the vertical pipe from (n,m) to $(n + 1,m)$

$K_{H1}^{(n,m)}$, $K_{H2}^{(n,m)}$ = the constants for the horizontal pipe from (n,m) to $(n,m + 1)$

$X_H^{(n,m)}$ = the quantity of fluid flowing in the horizontal direction from the node (n,m) to $(n,m + 1)$

$X_V^{(n,m)}$ = the quantity of fluid flowing in the vertical direction from the node (n,m) to (n + 1,m)

Using the above notations the percentage of the volume at the node (n,m) which flows in the horizontal direction can be expressed by:

$$p^{(n,m)} = X_H^{(n,m)} / \left(X_H^{(n,m)} + X_V^{(n,m)} \right) \tag{3.3}$$

with initial conditions that

$$p^{(n,m)} = 0.0 \text{ and } X_H^{(n,m)} = 0.0 \ (n = 0,1,2) \tag{3.4}$$

$$p^{(N,m)} = 1.0 \text{ and } X_V^{(N,m)} = 0.0 \ (m = 0,1,2) \tag{3.5}$$

3.1 Example Problem

4. A 2 x 2 network shown in Figure 8 is solved where $v^{(0,0)}$ units of fluid enter at node (0,0) and $v^{(n,m)}$ ($n = 0,1,2$; $m = 0,1,2$; $(n,m) \neq (0,0)$) units of fluid enter and leave at other nodes which are not necessarily the same nodes. The problem is one of determining $p^{(n,m)}$ that will minimize the total cost of transportation of the fluid through the network.

4.0 LITERATURE SURVEY

Various optimization techniques and algorithms have been used as a basis for traffic assignment. Wilson Campbell (1) presented a procedure to assign traffic to expressways in 1956. Moore (2) and Dantzing (3) developed algorithms for selecting the shortest path through a network. Wattleworth and Shuldiner (4) illustrated a basic application of inear programming to traffic assignment problems. Since 1957 many other techniques have been developed to determine the shortest path through a network. However, these techniques have not been as widely adapted as the Moore algorithm which is currently the method used with most computer traffic assignment problems(5).

Current traffic assignment are of "all or nothing" type, that is, all of the trips between two zones are assigned to a single route regardless of traffic volume on that route. This method lacks realism in that it does not provide for a revision of the link travel time as traffic volume increases.

Tsung-chang Yang and R. R. Snell (6) presented an application of an optimal traffic assignment technique which has the ability to overcome the capacity restraint shortcoming of the present day assignment procedure. They used the discrete version of the "maximum principle" (7) with linear time functions. R. R. Snell, M. L. Funk and J. B. Blackburn (5) used the same method with constant, linear and nonlinear time volume relationship.

Cantrell (9) has made investigations of pressure and flow of fluid through pipeline network. He determines the pressure drop in an existing pipeline for a given Reynold's number. However, a suitable method of assigning the volume of fluid to be transported over each link has not been developed. By considering the pressure drop as a cost of transportation the proposed dynamic programming method will provide such a method.

No one has yet solved the problems 2,3, and 4 above and no one has utilized dynamic programming for solving these types of problems. Thus the purpose of this paper is to illustrate that dynamic programming can be used to solve the problems stated above.

5. DYNAMIC PROGRAMMING

Dynamic programming developed by Bellman (9) is a mathematical technique which is used to serve many types of multistage decision problems. This technique is based on the "principle of optimality" which is stated by Bellman as:

"An optimal policy has the property that whatever the initial decisions are, the remaining decisions must constitute an optimal policy with regard to the state resulting from first decisions."

Let the function to be maximized be denoted by:

$$R(X_1, X_2, \ldots\ldots, X_k) = g_1(X_1) + g_2(X_2) + \ldots\ldots + g_K(X_K) \tag{5.1}$$

over the region $X_k \geqslant 0.0$ and $\sum_{k=1}^{K} X_k = X$

where

X = the amount of resources available

X_1 = the amount of resources allocated at stage k.

Since the maximum of the function $R(X_1, X_2, \ldots, X_K)$ over the designated region depends upon X and K, the sequence of functions $f_k(X)$ are introduced and are defined for k = 1,2 ....X and $X_k \geqslant 0.0$ as follows:

$$f_k(X) = \text{maximum } R(X_1, X_2, \ldots X_K) \tag{5.2}$$

where

$$X_k \geqslant 0.0 \text{ and } \sum_{k=1}^{K} X_k = X$$

The optimal value of the function $f_k(X)$ is obtained by allocating the resources X to the K activities in an optimal fashion. The problem considered here satisfies the following relationship:

$$f_1(X) = g_1(X) , \qquad X \geqslant 0 \tag{5.3}$$

where

$$f_0(0) = 0.0$$

The recurrence relation connecting $f_K(X)$ and $f_{K-1}(X)$ for some arbitrary X is obtained from equation (5.1), thus

$$f_K(x) = \underset{0 \leqslant X_k \leqslant X}{\text{Maximum}} \left[g_K(x) + f_{K-1}(X - X_K) \right] \tag{5.4}$$

The process is repeated for $f_{K-1}(X - X_K)$ to obtain the resurrence relationship

$$f_k(X') = \underset{0 \leqslant X_k \leqslant X'}{\text{Maximum}} \left[g_k(X_k) + f_{k-1}(X' - X_k) \right] \tag{5.5}$$

where

$$X' = X - \sum X_i$$

Thus if $f_1(X')$ is known, the sequence $f_K(X')$ can be obtained from equation (5.5).

6. THE SOLUTION PROCEDURE

Before discussing the method of solution for a traffic and pipeline network, it is necessary to state the assumptions made for each problem.

I. For the traffic assignment problem the assumptions are:

1. There are no turn penalties, that is, no extra time is required in making a turn.
2. The zone centriod coincides with the nodes.
3. The traffic directions are known.
4. Travel time is the only factor that influences the traffic pattern.
5. The travel time on each link can be expressed by Equation (2.1) with the appropriate constants.

II. For the pipeline problem the assumptions are:

1. The flow directions are known.
2. Cost is the only factor that influences the flow pattern.
3. Fluid is tapped or fed into the network only at the nodes.
4. The cost of transportation for each link is given by Equation (3.1) with the appropriate constants.

6.1 The Solution of a N x M Traffic Assignment Problem by Dynamic Programming

The following procedure is outlined for solving a N x M network by dynamic programming.

STEP 1: Divide the network into K stages in the following way:

K^{th} stage: All the routes that form a rectangular whose diagonal nodes are $(N - 1, M - 1)$ and (N,M).

$(K-1)^{th}$ stage: All the routes that form a rectangle whose diagonal nodes are $(N - 1, M - 2)$ and (N,M).

$(K-2)^{th}$ stage: All the routes that form a rectangle whose diagonal nodes are $(N - 2, M - 1)$ and (N,M).

. . . .

. . . .

1^{st} stage: All the routes that form a rectangle whose diagonal nodes are $(0,0)$ and (N,M).

In short the network can be formed by moving a diagonal straight line, perpendicular to the line joining $(0,0)$ and (N,M). Whenever this line touches the node (n,m) ($n \neq N$ and $m \neq M$) a stage can be formed by taking all the routes that form a rectangle whose diagonals are (n,m) and (N,M). Figure 4 shows a step by step procedure of dividing a 3 x 3 network into 9 stages. It is also noted that the nodes covered with hatch marks should be excluded from the stages since they do not alter the value of $_p(n,m)$ in determining the minimum travel time.

STEP 2: Assume an initial value of 0.5 for all $_p(n,m)$, the fraction of vehicles at node (n,m) that travel in the horizontal direction towards the node $(n,m + 1)$ with the exceptions that:

$$_p(n,M) = 0.0 \;;\; n = 0,1,2....,N$$

and

$$_p(N,m) = 1.0 \;,\; m = 0,1,2....,M$$

STEP 3: With these values of $_p(n,m)$, start from the node (0,0) and determine the number of vehicles on all the routes. This number must be an integer. If it is a fraction, convert it to the nearest integer.

STEP 4: Now with the vehicles loads as determined in step 3, start at the k^{th} stage, which represents node (i,j) and by keeping the number of vehicles entering the k^{th} stage constant, determine the new value of $_p(i,j)$ that minimizes the total time given by the following equation:

$$T = \sum_{m=j}^{M} \sum_{n=i}^{N} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.1)$$

In this discussion a single search technique (see appendix I for details) has been used for all the problems. Now the previous value of $_p(i,j)$ is replaced with the new value and the number of vehicles on all the routes are adjusted according to the following relation:

$$X_H^{(n,m)} = {}_P(n,m) \cdot \left(X_H^{(n,m-1)} + X_V^{(n-1,m)} \right) \tag{6.2}$$

$$(n = i, i+1, \ldots, N;\ m = j, j+1, \ldots, M)$$

Procede to the next stage and repeat the process. The process is repeated for all stages until new values for all ${}_P(n,m)$ have been determined.

STEP 5: One iteration is complete when the new values for all the ${}_P(n,m)$ have been determined. The values of ${}_P(n,m)$ from this iteration are compared to the corresponding values from the previous iteration. When the values of ${}_P(n,m)$ do not differ significantly on two successive iterations the answer is considered optimal. If they do differ significantly, go to step 3 using these new values of the ${}_P(n,m)$ as the initial values and repeat the entire procedure until an optimalsolution is obtained.

The solution procedure is illustrated by solving Example 1.

6.2 Example 1.

Example 1 is based on the 3 x 3 network illustrated in Figure 3 where all nodes are denoted by (0,0),..., (3,3). The percentage of vehicles at each node which procede in the horizontal direction is denoted by $p^{(n,m)}$ (n = 0,1,2,3; m = 0,1,2,3). Similarly the percentage of the vehicles which procedes in the vertical direction is denoted by $1 - p^{(n,m)}$. The number of vehicles entering the network at node (0,0) is denoted by $v^{(0,0)}$. The total time required to travel from (0,0) to (3,3) is given by the following equation:

$$T = \sum_{m=0}^{3} \sum_{n=0}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 \qquad (6.3)$$
$$+ K_{V1}^{(n,m)} \cdot X_V^{(n,m)}$$
$$+ K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2$$

where $K_{H1}^{(n,m)}$, $K_{H2}^{(n,m)}$ = the constants for the horizontal street from the node (n,m) to (n,m + 1)

$K_{V1}^{(n,m)}$, $K_{V2}^{(n,m)}$ = the constants for the vertical street from the node (n,m) to (n + 1,m)

$X_H^{(n,m)}$, $X_V^{(n,m)}$ = number of vehicles passing in the horizontal and vertical direction from the node (n,m) to (n,m + 1) and (n + 1,m) respectively.

In this example:

$$p^{(n,3)} = 0.0 \; ; \; n = 0,1,2,3 \qquad (6.4)$$

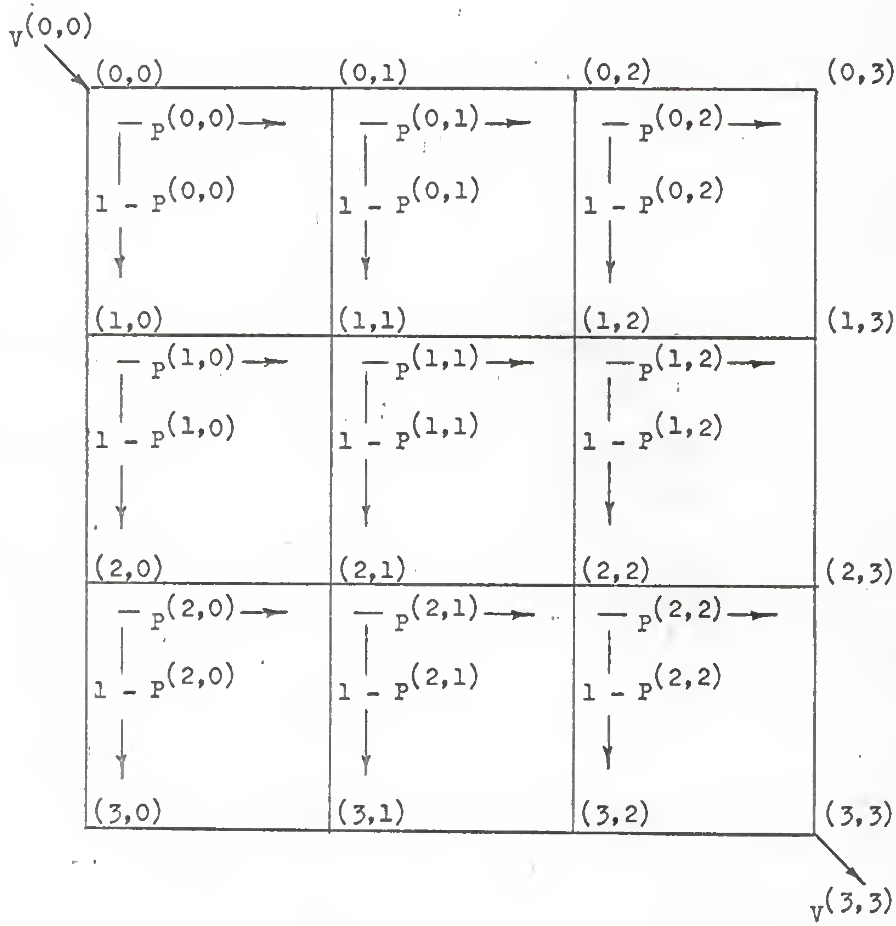
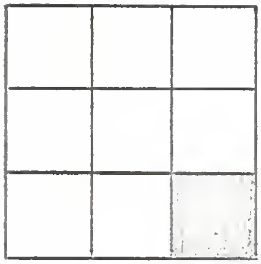
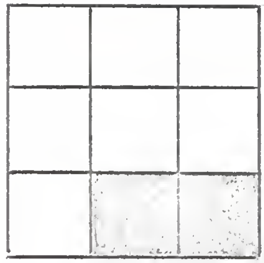


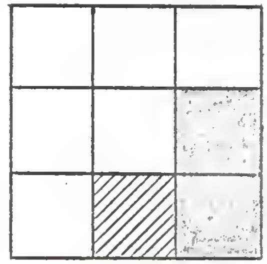
Figure 3. 3x3 Network



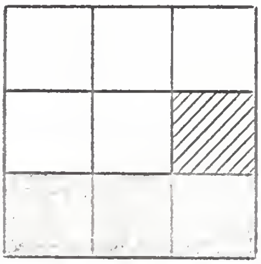
STAGE # 9



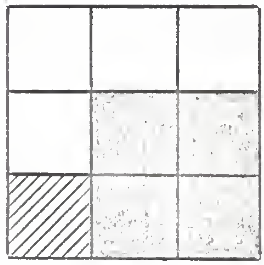
STAGE # 8



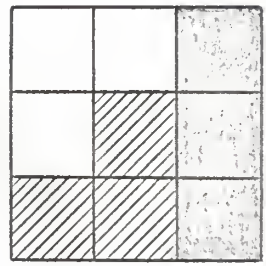
STAGE # 7



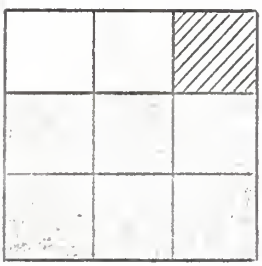
STAGE # 6



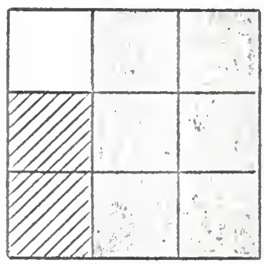
STAGE # 5



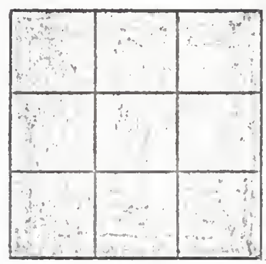
STAGE # 4



STAGE # 3



STAGE # 2



STAGE # 1

Note : The hatched portion is omitted from consideration since the vehicles in that area do not affect the value of $P^{(n,m)}$ at that stage.

Figure 4. 9 stages of a 3x3 network.

and ${}_p(3,m) = 1.0$; $m = 0,1,2,3$ (6.5)

Thus ${}_p(n,m) = X_H(n,m) \,/\, (X_H(n,m) + X_V(n,m))$ (6.6)

The objective is to determine the set of ${}_p(n,m)$ ($n = 0,1,2,3$; $m = 0,1,2,3$) which will minimize the time required to travel from node (0,0) to node (3,3). The procedure for solving this problem by dynamic programming is as follows:

STEP 1: Divide the network into 9 stages as shown in Figure 4.

Stage 9: (2,2), (2,3), (3,3) and (3,2)

Stage 8: (2,1), (2,2), (2,3), (3,3), (3,2) and (3.1)

.. ..

.. ..

Stage 1: (0,0) to (0,3) to (3,3) to (3,0)

STEP 2: Assume initial values of all ${}_p(n,m)$ to be 0.5. Note that:

$${}_p(n,3) = 0.0 \; ; \; (n = 0,1,2,3)$$

and

$${}_p(3,m) = 1.0 \; ; \; (m = 0,1,2,3)$$

Now, starting from node (0,0), determine the number of vehicles on all the routes of the network by equation (6.2).

STEP 3: Beginning at the k^{th} stage and keeping the number of vehicles entering this stage constant, ${}_p(i,j)$ is determined by changing $X_H(i,j)$ such that the total travel time for this stage is a minimum. The equation for determining the travel time at each stage are given below.

The equation for the 9th stage is:

$$T = \sum_{m=2}^{3}\sum_{n=2}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.7)$$

The equation for the 8th stage is:

$$T = \sum_{m=2}^{3}\sum_{n=1}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.8)$$

The equation for the 7th stage is:

$$T = \sum_{m=1}^{3}\sum_{n=2}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.9)$$

The equation for the 6th stage is:

$$T = \sum_{m=1}^{3}\sum_{n=0}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2 + K_{V1}^{(n,m)} \cdot X_V^{(n,m)} + K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.10)$$

The equation for the 5th stage is:

$$T = \sum_{m=1}^{3} \sum_{n=1}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2$$
$$+ K_{V1}^{(n,m)} \cdot X_V^{(n,m)}$$
$$+ K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.11)$$

The equation for the 4th stage is:

$$T = \sum_{m=0}^{3} \sum_{n=0}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2$$
$$+ K_{V1}^{(n,m)} \cdot X_V^{(n,m)}$$
$$+ K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.12)$$

The equation for the 3rd stage is:

$$T = \sum_{m=1}^{3} \sum_{n=0}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2$$
$$+ K_{V1}^{(n,m)} \cdot X_V^{(n,m)}$$
$$+ K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.13)$$

The equation for the 2nd stage is:

$$T = \sum_{m=0}^{3} \sum_{n=1}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2$$
$$+ K_{V1}^{(n,m)} \cdot X_V^{(n,m)}$$
$$+ K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.14)$$

The equation for the 1^{st} stage is:

$$T = \sum_{m=0}^{3}\sum_{n=0}^{3} K_{H1}^{(n,m)} \cdot X_H^{(n,m)} + K_{H2}^{(n,m)} \cdot (X_H^{(n,m)})^2$$

$$+ K_{V1}^{(n,m)} \cdot X_V^{(n,m)}$$

$$+ K_{V2}^{(n,m)} \cdot (X_V^{(n,m)})^2 \qquad (6.15)$$

Thus the new value of $p^{(2,2)}$ for the 9^{th} stage is the one which minimizes the time given by the equation for the 9^{th} stage and this is determined by varying the value of $X_H^{(2,2)}$. The previous value of $p^{(2,2)}$ is replaced by this new value and the number of vehicles on all the routes are adjusted according to equation (6.2).

$$X_H^{(n,m)} = p^{(n,m)} \cdot (X_H^{(n,m-1)} + X_V^{(n-1,m)})$$

$$(n = 0,1,2,3;\ m = 0,1,2,3) \qquad (6.16)$$

STEP 4: Preceding to the 8^{th} stage and repeat step 3 to determine the new value of $p^{(2,1)}$ that will minimize the total time for all vehicles at the 8^{th} stage given by equation (6.8). The previous value of $p^{(2,1)}$ is replaced by the new value and the number of vehicles on all the routes are adjusted. This procedure is repeated for all the stages to determine the new value of $p^{(n,m)}$.

STEP 5: One iteration is completed when all the values of $p^{(n,m)}$ have been determined. These values are then compared with the corresponding values of the previous iteration. If they are not significantly different the optimal answer has been obtained. If they are different the procedure continues at step 3 and the number of vehicles on all the routes are adjusted for this new set of $p^{(n,m)}$. The procedure is repeated until the optimal solution is obtained.

This procedure has been programmed for the IBM 1410 computer and is listed in appendix II. The solution to example 1 which follows was obtained using this program.

Example 1.

Data

$K_{H1}^{(0,0)} = 50.0$	$K_{H2}^{(0,0)} = 2.8$	$K_{H1}^{(0,1)} = 66.0$	$K_{H2}^{(0,1)} = 2.1$
$K_{H1}^{(0,2)} = 56.0$	$K_{H2}^{(0,2)} = 1.7$	$K_{H1}^{(1,0)} = 80.0$	$K_{H2}^{(1,0)} = 1.9$
$K_{H1}^{(1,1)} = 60.0$	$K_{H2}^{(1,1)} = 1.3$	$K_{H1}^{(1,2)} = 45.0$	$K_{H2}^{(1,2)} = 2.1$
$K_{H1}^{(2,0)} = 52.0$	$K_{H2}^{(2,0)} = 1.2$	$K_{H1}^{(2,1)} = 71.0$	$K_{H2}^{(2,1)} = 3.2$
$K_{H1}^{(2,2)} = 91.0$	$K_{H2}^{(2,2)} = 1.6$	$K_{H1}^{(3,0)} = 63.0$	$K_{H2}^{(3,0)} = 1.4$
$K_{H1}^{(3,1)} = 51.0$	$K_{H2}^{(3,1)} = 1.9$	$K_{H1}^{(3,2)} = 30.0$	$K_{H2}^{(3,2)} = 2.8$
$K_{V1}^{(0,0)} = 60.0$	$K_{V2}^{(0,0)} = 3.2$	$K_{V1}^{(0,1)} = 75.0$	$K_{V2}^{(0,1)} = 2.5$
$K_{V1}^{(0,3)} = 71.0$	$K_{V2}^{(0,3)} = 2.1$	$K_{V1}^{(1,0)} = 85.0$	$K_{V2}^{(1,0)} = 2.6$
$K_{V1}^{(1,1)} = 45.0$	$K_{V2}^{(1,1)} = 2.9$	$K_{V1}^{(1,2)} = 52.0$	$K_{V2}^{(1,2)} = 1.8$
$K_{V1}^{(1,3)} = 90.0$	$K_{V2}^{(1,3)} = 3.1$	$K_{V1}^{(2,0)} = 61.0$	$K_{V2}^{(2,0)} = 1.4$
$K_{V1}^{(2,1)} = 31.0$	$K_{V2}^{(2,1)} = 2.6$	$K_{V1}^{(0,2)} = 66.0$	$K_{V2}^{(0,2)} = 3.1$
$K_{V1}^{(2,2)} = 81.0$	$K_{V2}^{(2,2)} = 2.5$	$K_{V1}^{(2,3)} = 50.0$	$K_{V2}^{(2,3)} = 2.1$

$V^{(0,0)} = 100.0$

Results

$P^{(0,0)} = 51.000\%$	$P^{(0,1)} = 52.941\%$	$P^{(0,2)} = 58.851\%$
$P^{(1,0)} = 44.897\%$	$P^{(1,1)} = 63.043\%$	$P^{(1,2)} = 26.190\%$
$P^{(2,0)} = 44.444\%$	$P^{(2,1)} = 37.931\%$	$P^{(2,2)} = 66.666\%$

6.3 Example 2.

Example 2 is based on a simple 2 x 2 network illustrated in Figure 5 where the nodes are represented by (0,0), (0,1) ..., (2,2). In this example $V_{NW}^{(0,0)}$ denotes the vehicles which enter the network from the northwest at node (0,0) and leave at node (2,2). Similarly $V_{SW}^{(2,0)}$ denotes the vehicles which enter the network at node (2,0) from the Southwest and leave at node (0,2). $P_{NW}^{(0,0)}$, $P_{NW}^{(0,1)}$, $P_{NW}^{(1,0)}$ and $P_{NW}^{(1,1)}$ represent the percentage of the vehicles which enter from northwest and turn in the horizontal direction. Similarly $P_{SW}^{(2,0)}$, $P_{SW}^{(2,1)}$, $P_{SW}^{(1,0)}$ and $P_{SW}^{(1,1)}$ represents the percentage of vehicles which enter from the Southwest and travel in the horizontal direction. The problem is one of determining the values of $P_{NW}^{(n,m)}$ and $P_{SW}^{(n,m)}$ which will minimize the total travel time of the network.

Observe from Figure 5 that the directions of these two types of vehicles are not the same everywhere. When the directions of two types of vehicles are the same, their sum can be combined to equal $X_H^{(n,m)}$ in equation (2,4), but when they are not traveling in the same direction, their times are found separately and added.

This problem is treated as a combination of two problems which are solved simultaneously. Thus there are eight stages; four for the Northwest vehicles and four for the Southwest

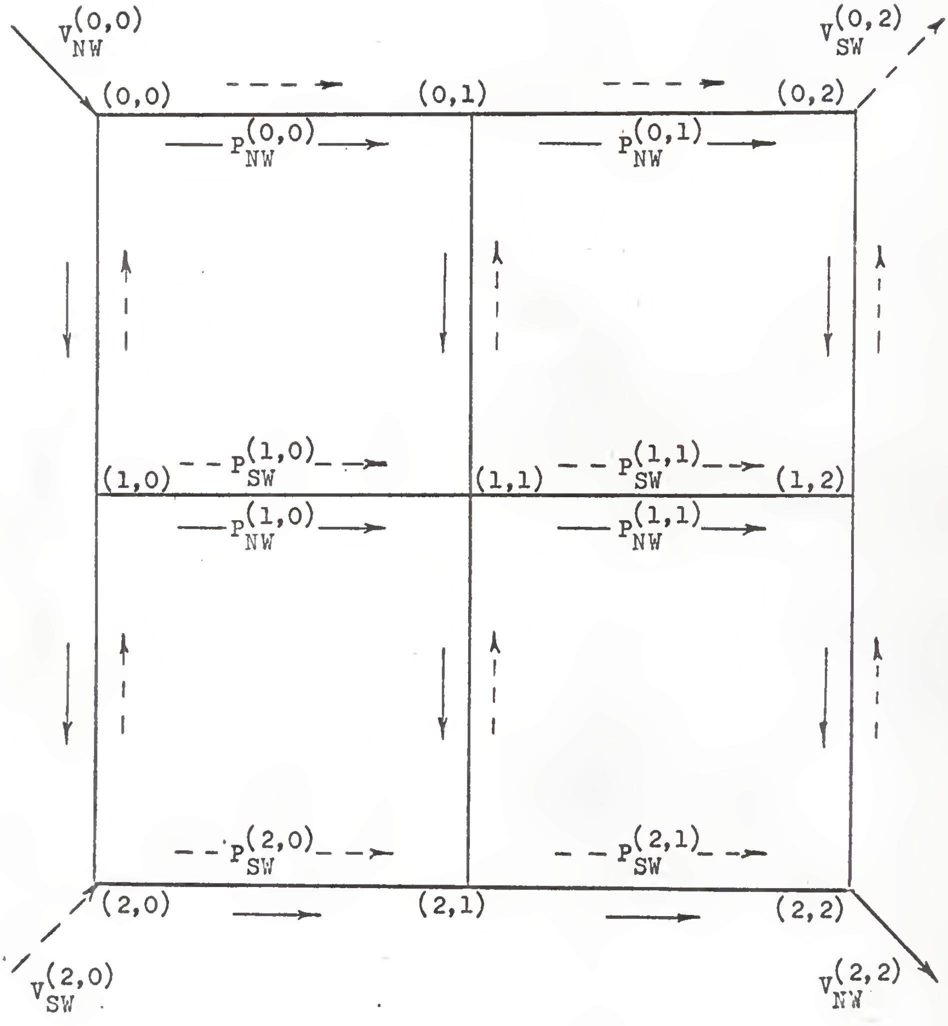
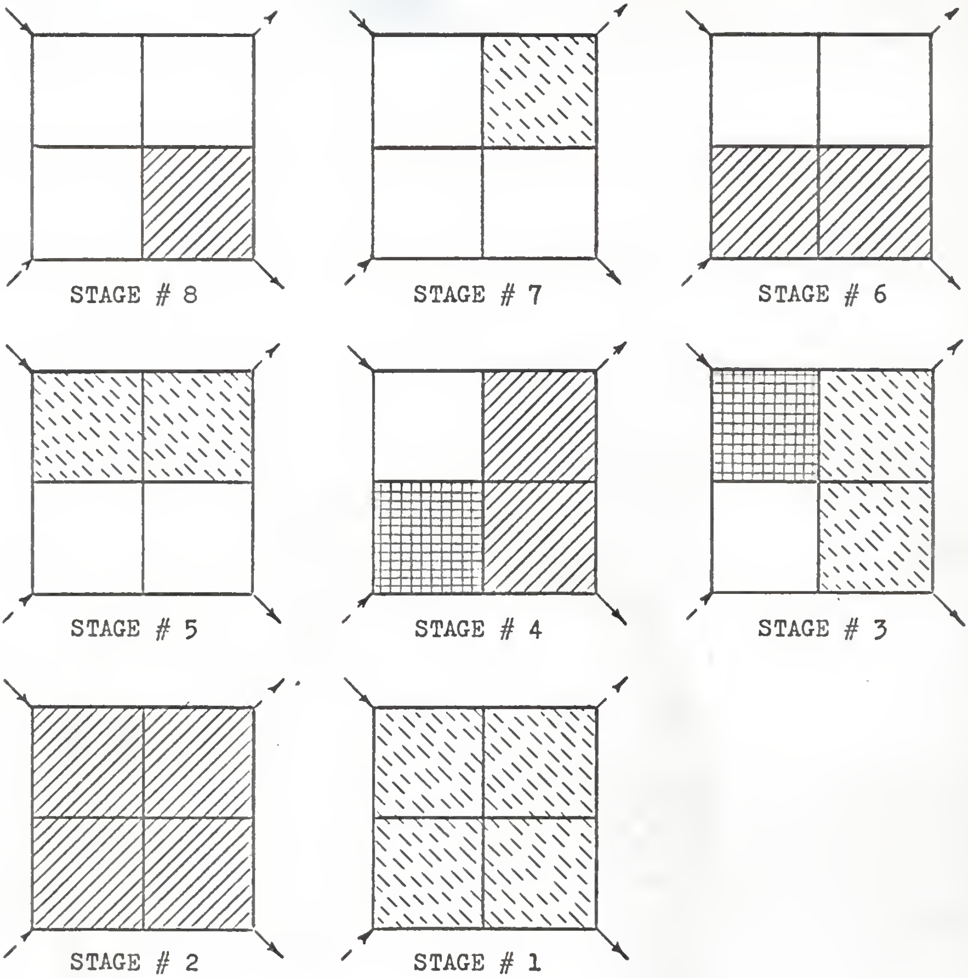


Figure 5. 2x2 Network



Note : A stage with solid hatched lines represents the part of the network where $P_{NW}^{(n,m)}$ is to be determined by varing the Northwest vehicles on the appropriate routes and by keeping the other vehicles which enter the stage constant. The alternate is true for the Southwest vehicles represented by dashed lines. The cross hatched portion is omitted from consideration since the vehicles in that area do not affect the value of $P_{NW}^{(n,m)}$ or $P_{SW}^{(n,m)}$ at that stage.

Figure 6. 8 stages of a 2x2 network.

vehicles. The procedure for solving this problem is as follows:

STEP 1: Divide the network into eight stages as shown in Figure 6. The Northwest and Southwest vehicles are considered alternately.

STEP 2: Assume initial values for $P_{NW}(0,0)$, $P_{NW}(0,1)$, $P_{NW}(1,1)$, $P_{NW}(1,0)$, $P_{SW}(2,0)$, $P_{SW}(2,1)$, $P_{SW}(1,0)$, $P_{SW}(1,1)$ to be equal to 0.5.

STEP 3: Starting from (0,0) and (2,0) respectively assign the Northwest and the Southwest vehicles to all the routes using the initial values of $P_{NW}(n,m)$ and $P_{SW}(n,m)$.

STEP 4: Starting from stage 8 and keeping all the vehicles entering the stage 8 constant, change $X_{NWH}(1,1)$ in such a way that the total travel time given by equation (2.4) for this stage is minimum. From this value of $X_{NWH}(1,1)$ determine the value of $P_{NW}(1,1)$ and replace the previous value with it. Make the new assignment of vehicles using this new value of $P_{NW}(1,1)$ and procede to step 5.

STEP 5: Repeat the procedure of step 4 for the Southwest vehicles at stage 7 and determine the value of $P_{SW}(1,1)$ which will minimize the traveling time at stage 7. Replace the previous value of $P_{SW}(1,1)$ by this new value and make the new assignment of vehicles using these new values of $p_{NW}(1,1)$ and $p_{SW}(1,1)$. Repeat the procedure for all the remaining stages to determine the corresponding values of $p_{NW}(n,m)$ and $p_{SW}(n,m)$. At each stage the new values of $p_{NW}(n,m)$ and $p_{SW}(n,m)$ are used to make the new assignment of vehicles on the links.

STEP 7: One iteration is complete when all the values of $p_{NW}(n,m)$ and $p_{SW}(n,m)$ have been determined. These values are compared with the values of the previous iteration. If there is no significant difference the optimal solution has been obtained. If they are different, return to step 3 with new set of values for $p_{NW}(n,m)$ and $p_{SW}(n,m)$.

This procedure has been programmed for the IBM 1410 computer and is listed in appendix III.

The solution to example 2 which follows was obtained using this program.

Example 2.

Data

$K_{H1}^{(0,0)} = 40.0$	$K_{H2}^{(0,0)} = 1.1$	$K_{H1}^{(0,1)} = 32.0$	$K_{H2}^{(0,1)} = 1.5$
$K_{H1}^{(1,0)} = 29.0$	$K_{H2}^{(1,0)} = 1.3$	$K_{H1}^{(1,1)} = 35.0$	$K_{H2}^{(1,1)} = 1.2$
$K_{H1}^{(2,0)} = 34.0$	$K_{H2}^{(2,0)} = 1.2$	$K_{H1}^{(2,0)} = 32.0$	$K_{H2}^{(2,0)} = 1.0$
$K_{V1}^{(0,0)} = 30.0$	$K_{V2}^{(0,0)} = 0.8$	$K_{V1}^{(0,1)} = 41.0$	$K_{V2}^{(0,1)} = 0.9$
$K_{V1}^{(0,2)} = 51.0$	$K_{V2}^{(0,2)} = 1.2$	$K_{V1}^{(1,0)} = 42.0$	$K_{V2}^{(1,0)} = 1.6$
$K_{V1}^{(1,1)} = 51.0$	$K_{V2}^{(1,1)} = 1.5$	$K_{V1}^{(1,2)} = 42.0$	$K_{V2}^{(1,2)} = 1.3$
$V_{NW}^{(0,0)} = 100.0$	$V_{SW}^{(2,0)} = 100.0$		

Results

$P_{NW}^{(0,0)} = 41.00\%$	$P_{NW}^{(0,1)} = 66.10\%$	$P_{NW}^{(1,0)} = 9.76\%$	$P_{NW}^{(1,1)} = 57.89\%$
$P_{SW}^{(2,0)} = 55.00\%$	$P_{SW}^{(1,0)} = 36.36\%$	$P_{SW}^{(2,1)} = 50.00\%$	$P_{SW}^{(1,1)} = 29.54\%$

6.4 Example 3.

Example 3 shown in Figure 7 is similar to problem 2 except that vehicles enter and leave at several nodes of the network. The problem is one of determining the values of $P_{NW}^{(n,m)}$ and $P_{SW}^{(n,m)}$ so that the total travel time is minimum.

The method of solving this example is same as example 2. When the vehicles are assigned to the routes, the arrival or departure of the respective vehicles at corresponding nodes are taken into account.

The solution to example 3 was obtained using the computer program in appendix III.

Example 3.

Data

$K_{H1}^{(n,m)}$, $K_{H2}^{(n,m)}$, $K_{V1}^{(n,m)}$ and $K_{V2}^{(n,m)}$ ($n = 0,1,2;\ m = 0,1,2$)

are same as example 2.

$V_{NW}^{(0,0)} = 100.0$ $V_{SW}^{(1,0)} = 100.0$ $V_{NW}^{(1,2)} = -20.0$ $V_{NW}^{(2,1)} = -30.0$

$V_{SW}^{(0,1)} = -30.0$ $V_{SW}^{(1,2)} = -10.0$

Results

$P_{NW}^{(0,0)} = 43.00\%$ $P_{NW}^{(1,0)} = 71.92\%$ $P_{NW}^{(0,1)} = 27.90\%$ $P_{NW}^{(1,1)} = 58.33\%$

$P_{SW}^{(2,0)} = 59.00\%$ $P_{SW}^{(1,0)} = 31.70\%$ $P_{SW}^{(1,1)} = 59.32\%$ $P_{SW}^{(1,1)} = 10.81\%$

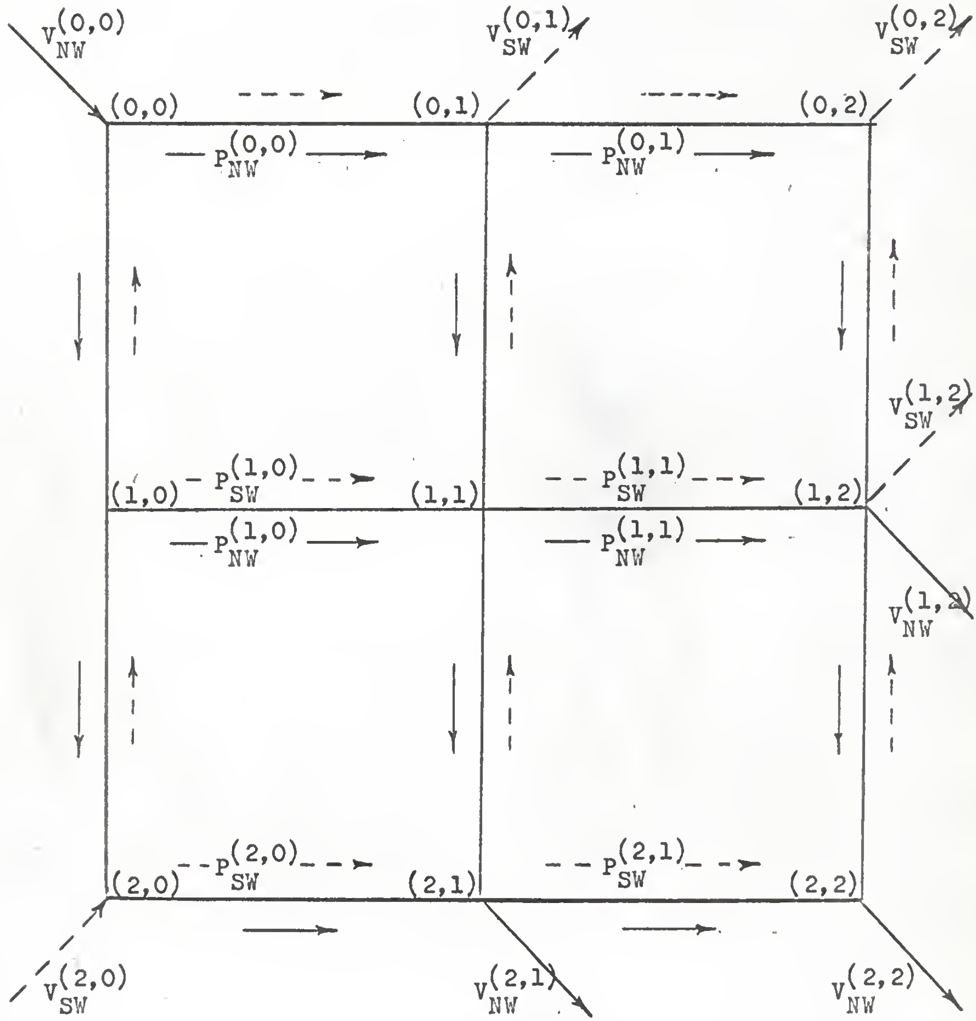


Figure 7. 2x2 Network

6.5 Example 4.

Pipeline problems are similar to traffic assignment problems except that fluid is treated as a continuous variable rather than a discrete or integer valued variable. The problem is one of determining the values of $p(n,m)$ which minimize the cost of transporting fluid. The procedure of solving a 2 x 2 pipeline network, shown in Figure 7, is as follows.

STEP 1: Divide the network into four stages in the same way as the traffic problems. The stages for this example are shown in Figure 8.

STEP 2: Assume initial values for $p(n,m)$ to be 0.5 and assign the corresponding quantity of fluid for all the pipes, taking into consideration the amount of fluid entering and leaving each node.

STEP 3: Starting from the 4^{th} stage and keeping all the fluid entering into this stage constant, search for the suitable value of $p(1,1)$ which will minimize the cost of transportation at the 4^{th} stage which is given by equation (4.2). Replace the previous value of $p(1,1)$ with this new value and make new assignment of fluid into network using this new value.

STEP 4: This procedure is repeated until all the new values of $p(1,0)$, $p(0,1)$ and $p(0,0)$ have been determined and the volume of fluid in the links of the network have been adjusted.

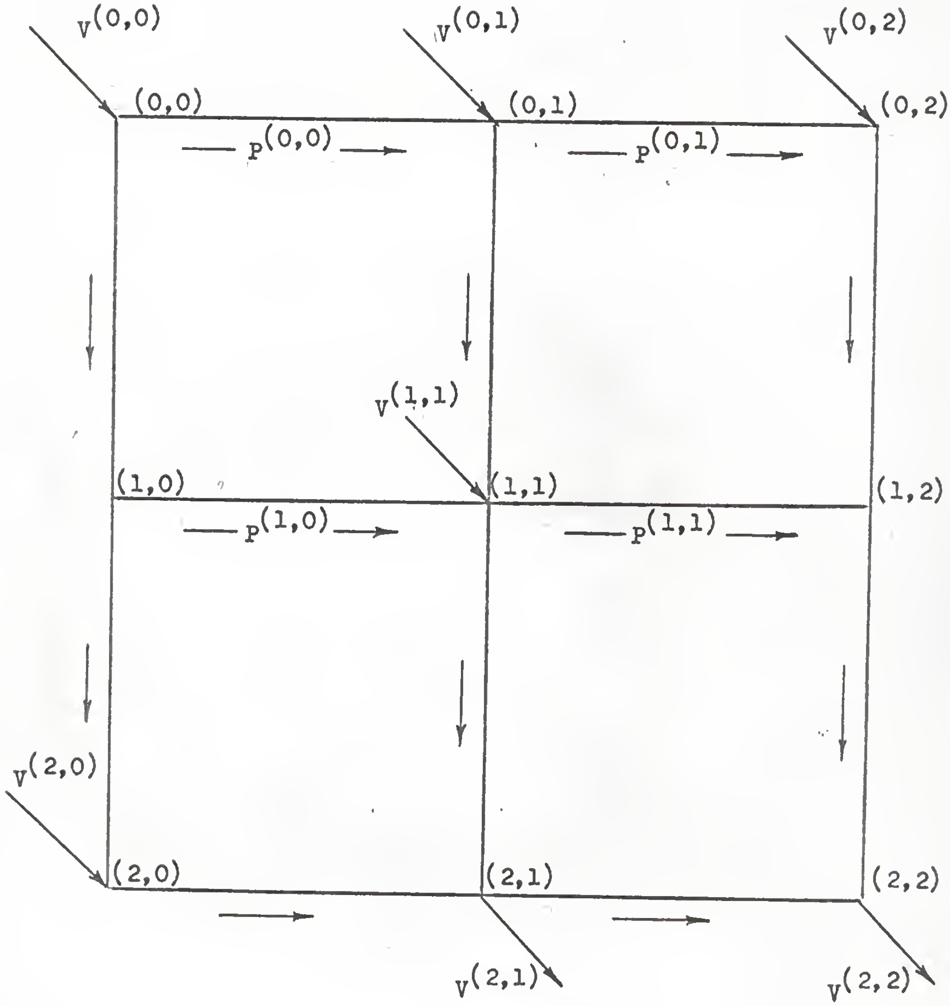
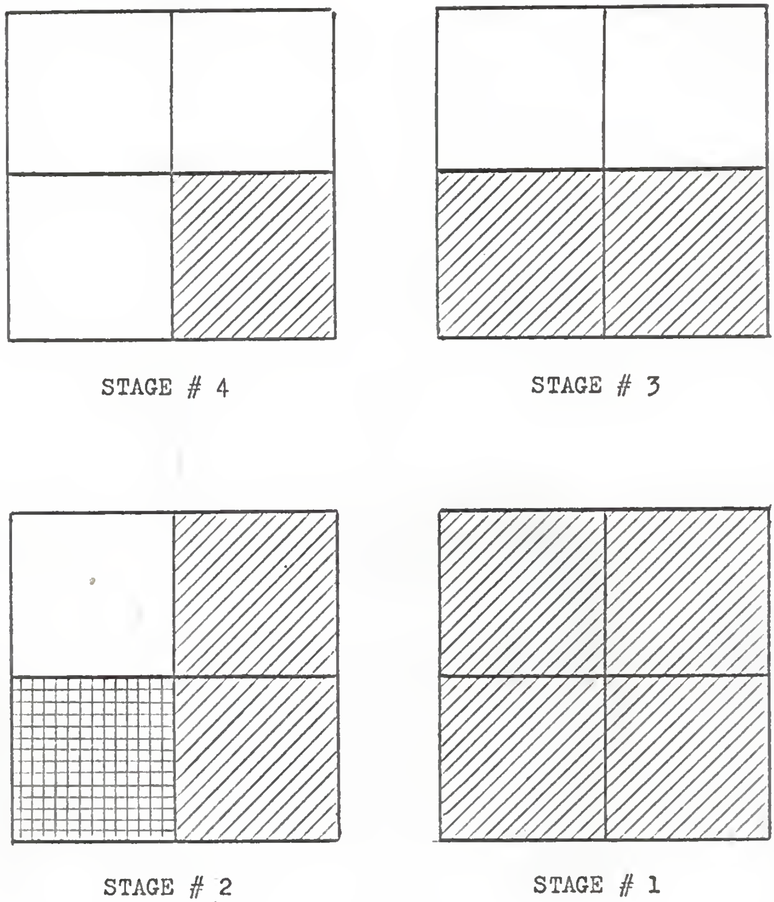


Figure 8. 2x2 Network



Note : The cross hatched portion is omitted from consideration since the fluid in that area does not affect the value of $P^{(n,m)}$ at that stage.

Figure 9.

STEP 5: Compare these values of ${}_p(n,m)$ with the corresponding values of the previous iteration. If there is no significant difference the optimal solution has been obtained. If there is a significant difference make the new load assignment with the new values of ${}_p(n,m)$ and return to the step 3. Repeat the whole procedure until an optimal solution is obtained. The solution to example 4 was obtained using the computer program in appendix IV.

Example 4.

Data

$K_{H1}^{(0,0)} = 51.0$	$K_{H2}^{(0,0)} = 1.8$	$K_{H1}^{(0,1)} = 67.0$	$K_{H2}^{(0,1)} = 1.9$
$K_{H1}^{(1,0)} = 70.0$	$K_{H2}^{(1,0)} = 1.6$	$K_{H1}^{(1,1)} = 63.0$	$K_{H2}^{(1,1)} = 1.4$
$K_{H1}^{(2,0)} = 65.0$	$K_{H2}^{(2,0)} = 1.2$	$K_{H1}^{(2,1)} = 65.0$	$K_{H2}^{(2,1)} = 1.8$
$K_{V1}^{(0,0)} = 55.0$	$K_{V2}^{(0,0)} = 1.2$	$K_{V1}^{(0,1)} = 61.0$	$K_{V2}^{(0,1)} = 1.1$
$K_{V1}^{(0,2)} = 58.0$	$K_{V2}^{(0,2)} = 1.5$	$K_{V1}^{(1,0)} = 65.0$	$K_{V2}^{(1,0)} = 1.3$
$K_{V1}^{(1,1)} = 58.0$	$K_{V2}^{(1,1)} = 1.4$	$K_{V1}^{(1,2)} = 45.0$	$K_{V2}^{(1,2)} = 1.2$
$V^{(0,0)} = 100.0$	$V^{(0,1)} = 100.0$	$V^{(0,2)} = 50.0$	$V^{(1,1)} = 100.0$
$V^{(2,0)} = 50.0$	$V^{(2,1)} = -50.0$	$V^{(2,2)} = -350.0$	

Results

$P^{(0,0)} = 27.00\%$	$P^{(1,0)} = 36.00\%$	$P^{(0,1)} = 38.00\%$	$P^{(1,1)} = 53.00\%$

7. SUMMARY

A number of one way traffic, two way traffic and pipeline problems have been solved by this method on IBM 1410 and 1620 computer. This method is suitable for small network problems. As the size of the network increases, the size of program reaches the capacity of the existing computer.

This method is not well suited for the complicated cases of example 3 with vehicles entering or leaving the network at more than two nodes. One reason is that the entrance or exit of the Northwest or the Southwest vehicles at any node other than the last stage, as in example 3, are dependent on the directions of each other. However, this method is applicable to pipeline problems since the question of multidirectional flow does not arise.

The success of this method lies in its simplicity. An important aspect of this method is that it is selfcorrecting and thus converges to the optimal solution if errors occur from roundoff, by performing additional iterations.

The percentage of vehicles at the node (n,m) which travels in the horizontal direction, that is $_p(n,m)$ $(n = 0, 1, \dots N;\ m = 0, 1, \dots M)$, is a function of number of vehicles and it will not remain the same as the number of vehicles are doubled.

8. CONCLUSION

One way or two way traffic problems can be successfully solved by this method. The program size is the limiting factor. This method is applicable for simple cases of two way traffic problems like example 2 and 3. However, this method is not applicable for two way traffic problems with vehicles entering or leaving the network at more than two nodes. The success of this method lies in its simplicity and computational efficiency. Even if the mistakes are made at a previous step, it is possible to obtain correct answer at the cost of few more iterations.

ACKNOWLEDGMENTS

The writer is deeply indebted to Dr. Frank A. Tillman, Head, Industrial Engineering Department, for his able guidance and suggestions given throughout this work.

He wishes to thank the staff members and friends of Industrial Engineering Department for the valuable aid in successful completion of this report.

REFERENCES

(1) Campbell, E. Wilson

A Mechanical Method for Assigning Traffic to Expressways., Bulletin 130, Highway Research Bd., Washington D. D., 1956.

(2) Moore, Edward G.

The Shortest Path Through a Maze., Proceeding of an International Symposium on the Theory of Switching, April, 1957.

(3) Dantzing, George B.

The Shortest Route Problem. Operation Research, 5; 270-3, 1957.

(4) Wattleworth, Joseph A. and Shuldiner, Paul W.

Analitical Method in Transportation: Left Turn Penalties in Traffic Assignment Models., Journal of the Engineering Mechanics Division, ASCE, Vol. 89, No. EM6, December, 1963.

(5) Snell, Funk and Blackburn

Traffic Assignment by Discrete Maximum Principle Using Constant Linear and Nonlinear Time-Volume Relationship. For Presentation at the Annual Meeting of highway Research Board, Washington D. C., January 16-20, 1967.

(6) Tsung-chang Yang and Snell, R. R.

Traffic Assignment by Maximum Principle., Journal of Highway Division, October 1966.

(7) Pontryagin, L. S., Boltyanskii, V. C., Gamkrelidze, R. V.
The Mathematical Theory of Optimal Process., Interscience, Inc., New York, N. Y., 1962.

(8) Bellman, R.
Dynamic Programming., Princeton University Press, Princeton, N. J., 1961.

(9) Cantrell, H. N.
Incompressible Flow Network Calculations., Comm. Acm. 6, June 1963.

APPENDIX I

SINGLE SEARCH METHOD

In order to find the value of $p^{(i,j)}$ of the k^{th} stage which will minimize the time for this stage, the following procedure is used.

STEP 1: Keep all the vehicles entering the X^{th} stage constant and assume an initial value of $X_H^{(i,j)}$ to equal 0.0. Adjust vehicles on all the routes according to the equation (6.2) and find the total time for k^{th} stage according to the equation (6.1). Denote this time by T_0.

STEP 2: Increase the value of $X_H^{(i,j)}$ by 5.0, adjust vehicles and find total time in the same way as in step 1. Denote this time by T_1.

STEP 3: Compare T_0 and T_1. If $T_1 < T_0$, replace the value of T_0 by the value of T_1, go to step 2 and repeat the procedure. If $T_1 \geqslant T_0$, go to step 4.

STEP 4: Decrease the value of $X_H^{(i,j)}$ by 1.0, adjust vehicles and find the total time in the same way as in step 1. Denote this time by T_2.

STEP 5: Compare T_2 and T_1. If $T_2 < T_1$, replace the value of T_1 by the value of T_2, go to step 4 and repeat the procedure. If $T_2 \geqslant T_1$ the value of $p^{(i,j)}$ which minimizes the time for the k^{th} stage is given by: $p^{(i,j)} = (X_H^{(i,j)} +_1) \,/\, (X_H^{(i-1,j)} + X_V^{(i,j-1)})$

APPENDIX II

PR 115 PROGRAM FOR EXAMPLE 1.

```
MON$$      JOB
MON$$      COMT 15 MINUTES,10 PAGES
MON$$      ASGN MJB,12
MON$$      ASGN MGO,16
MON$$      MODE GO,TEST
MON$$      EXEQ FORTRAN,,,,,,,NO01
      DIMENSION P1(20),P2(20),P3(20),P4(20),P5(20),P6(20)
      DIMENSION P7(20),P8(20),P9(20)
  160 FORMAT(I2,9F9.5,F15.2)
      BK1=50.
      BK2=60.
      BK3=66.
      BK4=75.
      BK5=56.
      BK6=66.
      BK7=71.
      BK8=80.
      BK9=85.
      BK10=50.
      BK11=45.
      BK12=45.
      BK13=52.
      BK14=90.
      BK15=52.
      BK16=61.
      BK17=71.
      BK18=31.
      BK19=91.
      BK20=81.
      BK21=50.
      BK22=63.
      BK23=51.
      BK24=30.
      CK1=2.8
      CK2=3.2
      CK3=2.1
      CK4=2.5
      CK5=1.7
      CK6=3.1
      CK7=2.1
      CK8=1.9
      CK9=2.6
      CK10=1.3
      CK11=2.9
      CK12=2.1
      CK13=1.8
      CK14=3.1
      CK15=1.2
      CK16=1.4
      CK17=3.2
      CK18=2.6
      CK19=1.6
```

```
      CK20=2.5
      CK21=2.1
      CK22=1.4
      CK23=1.9
      CK24=2.8
      M=1
      SMA=100.
      P1(1)=.5
      P2(1)=.5
      P3(1)=.5
      P4(1)=.5
      P5(1)=.5
      P6(1)=.5
      P7(1)=.5
      P8(1)=.5
      P9(1)=.5
      IA=1
      IB=1
      IC=1
      ID=1
      IE=1
      IF=1
      IG=1
      IH=1
      II=1
    1 DA=0.
      DB=0.
      DC=0.
      DD=0.
      DE=0.
      DF=0.
      DG=0.
      DH=0.
      DI=0.
      M1=P1(IA)*SMA+.5
      SM1=M1
    2 SM2=SMA-SM1
      M3=P2(IB)*SM1+.5
      SM3=M3
    3 SM4=SM1-SM3
      M8=P4(ID)*SM2+.5
      SM8=M8
    4 SM9=SM2-SM8
      M5=P3(IC)*SM3+.5
      SM5=M5
    5 SM6=SM3-SM5
      SM7=SM5
      M10=P5(IE)*(SM4+SM8)+.5
      SM10=M10
    6 SM11=(SM4+SM8)-SM10
      M15=P7(IG)*SM9+.5
      SM15=M15
```

```
  7 SM16=SM9-SM15
    SM22=SM16
    M12=P6(IF)*(SM6+SM10)+.5
    SM12=M12
  8 SM13=(SM6+SM10)-SM12
    SM14=SM12+SM7
    M17=P8(IH)*(SM11+SM15)+.5
    SM17=M17
  9 SM18=(SM11+SM15)-SM17
    SM23=SM18+SM22
    M19=P9(II)*(SM13+SM17)+.5
    SM19=M19
 10 SM20=(SM13+SM17)-SM19
    SM21=SM14+SM19
    SM24=SM23+SM20
 11 IF (II-IH) 12,12,30
 12 IF (IH-IF) 13,13,33
 13 IF (IF-IG) 14,14,36
 14 IF (IG-IE) 15,15,39
 15 IF (IE-IC) 16,16,42
 16 IF (IC-ID) 17,17,45
 17 IF (ID-IB) 18,18,48
 18 IF (IB-IA) 19,19,51
 19 IF (IA-M) 60,60,54
 30 CI1=BK19*SM19+CK19*SM19**2+BK20*SM20+CK20*SM20**2
    CI2=BK21*SM21+CK21*SM21**2+BK24*SM24+CK24*SM24**2
    CI=CI1+CI2
    IF (SM19-DI) 66,61,63
 33 CH1=BK17*SM17+CK17*SM17**2+BK18*SM18+CK18*SM18**2
    CH2=BK23*SM23+CK23*SM23**2+BK20*SM20+CK20*SM20**2
    CH3=BK19*SM19+CK19*SM19**2+BK24*SM24+CK24*SM24**2
    CH4=BK21*SM21+CK21*SM21**2
    CH=CH1+CH2+CH3+CH4
    IF (SM17-DH) 76,71,73
 36 CF1=BK12*SM12+CK12*SM12**2+BK13*SM13+CK13*SM13**2
    CF2=BK14*SM14+CK14*SM14**2+BK19*SM19+CK19*SM19**2
    CF3=BK20*SM20+CK20*SM20**2+BK21*SM21+CK21*SM21**2
    CF4=BK24*SM24+CK24*SM24**2
    CF=CF1+CF2+CF3+CF4
    IF (SM12-DF) 86,81,83
 39 CG1=BK15*SM15+CK15*SM15**2+BK16*SM16+CK16*SM16**2
    CG2=BK22*SM22+CK22*SM22**2+BK18*SM18+CK18*SM18**2
    CG3=BK17*SM17+CK17*SM17**2+BK20*SM20+CK20*SM20**2
    CG4=BK23*SM23+CK23*SM23**2+BK19*SM19+CK19*SM19**2
    CG5=BK24*SM24+CK24*SM24**2+BK21*SM21+CK21*SM21**2
    CG=CG1+CG2+CG3+CG4+CG5
    IF (SM15-DG) 96,91,93
 42 CE1=BK10*SM10+CK10*SM10**2+BK11*SM11+CK11*SM11**2
    CE2=BK12*SM12+CK12*SM12**2+BK18*SM18+CK18*SM18**2
    CE3=BK13*SM13+CK13*SM13**2+BK17*SM17+CK17*SM17**2
    CE4=BK14*SM14+CK14*SM14**2+BK23*SM23+CK23*SM23**2
    CE5=BK20*SM20+CK20*SM20**2+BK19*SM19+CK19*SM19**2
```

```
   CE6=BK21*SM21+CK21*SM21**2+BK24*SM24+CK24*SM24**2
   CE=CE1+CE2+CE3+CE4+CE5+CE6
   IF (SM10-DE) 106,101,103
45 CC1=BK5*SM5+CK5*SM5**2+BK6*SM6+CK6*SM6**2
   CC2=BK7*SM7+CK7*SM7**2+BK12*SM12+CK12*SM12**2
   CC3=BK13*SM13+CK13*SM13**2+BK19*SM19+CK19*SM19**2
   CC4=BK14*SM14+CK14*SM14**2+BK20*SM20+CK20*SM20**2
   CC5=BK21*SM21+CK21*SM21**2+BK24*SM24+CK24*SM24**2
   CC=CC1+CC2+CC3+CC4+CC5
   IF (SM5-DC) 116,111,113
48 CD1=BK8*SM8+CK8*SM8**2+BK9*SM9+CK9*SM9**2
   CD2=BK15*SM15+CK15*SM15**2+BK11*SM11+CK11*SM11**2
   CD3=BK10*SM10+CK10*SM10**2+BK13*SM13+CK13*SM13**2
   CD4=BK17*SM17+CK17*SM17**2+BK12*SM12+CK12*SM12**2
   CD5=BK14*SM14+CK14*SM14**2+BK19*SM19+CK19*SM19**2
   CD6=BK16*SM16+CK16*SM16**2+BK18*SM18+CK18*SM18**2
   CD7=BK22*SM22+CK22*SM22**2+BK23*SM23+CK23*SM23**2
   CD8=BK20*SM20+CK20*SM20**2+BK21*SM21+CK21*SM21**2
   CD9=BK24*SM24+CK24*SM24**2
   CD=CD1+CD2+CD3+CD4+CD5+CD6+CD7+CD8+CD9
   IF (SM8-DD) 126,121,123
51 CB1=BK3*SM3+CK3*SM3**2+BK4*SM4+CK4*SM4**2
   CB2=BK5*SM5+CK5*SM5**2+BK6*SM6+CK6*SM6**2
   CB3=BK7*SM7+CK7*SM7**2+BK10*SM10+CK10*SM10**2
   CB4=BK12*SM12+CK12*SM12**2+BK11*SM11+CK11*SM11**2
   CB5=BK13*SM13+CK13*SM13**2+BK14*SM14+CK14*SM14**2
   CB6=BK17*SM17+CK17*SM17**2+BK19*SM19+CK19*SM19**2
   CB7=BK18*SM18+CK18*SM18**2+BK20*SM20+CK20*SM20**2
   CB8=BK21*SM21+CK21*SM21**2+BK23*SM23+CK23*SM23**2
   CB9=BK24*SM24+CK24*SM24**2
   CB=CB1+CB2+CB3+CB4+CB5+CB6+CB7+CB8+CB9
   IF (SM3-DB) 136,131,133
54 CA1=BK1*SM1+CK1*SM1**2+BK2*SM2+CK2*SM2**2
   CA2=BK3*SM3+CK3*SM3**2+BK4*SM4+CK4*SM4**2
   CA3=BK5*SM5+CK5*SM5**2+BK6*SM6+CK6*SM6**2
   CA4=BK7*SM7+CK7*SM7**2+BK8*SM8+CK8*SM8**2
   CA5=BK9*SM9+CK9*SM9**2+BK10*SM10+CK10*SM10**2
   CA6=BK11*SM11+CK11*SM11**2+BK12*SM12+CK12*SM12**2
   CA7=BK13*SM13+CK13*SM13**2+BK14*SM14+CK14*SM14**2
   CA8=BK15*SM15+CK15*SM15**2+BK16*SM16+CK16*SM16**2
   CA9=BK17*SM17+CK17*SM17**2+BK18*SM18+CK18*SM18**2
   CA10=BK19*SM19+CK19*SM19**2+BK20*SM20+CK20*SM20**2
   CA11=BK21*SM21+CK21*SM21**2+BK22*SM22+CK22*SM22**2
   CA12=BK23*SM23+CK23*SM23**2+BK24*SM24+CK24*SM24**2
   CA=CA1+CA2+CA3+CA4+CA5+CA6+CA7+CA8+CA9+CA10+CA11+CA12
   IF (SM1-DA) 146,141,143
60 II=II+1
   SM19=0.
   GO TO 10
61 CCI1=CI
62 SM19=SM19+5.
   GO TO 10
```

```
63 CCI2=CI
   IF (CCI1-CCI2) 65,65,64
64 CCI1=CCI2
   GO TO 62
65 SM19=SM19-1.
   DI=SM19+5.
   GO TO 10
66 CCI3=CI
   IF (CCI2-CCI3) 69,69,67
67 CCI2=CCI3
   GO TO 65
69 P9(II)=(SM19+1.)/(SM13+SM17)
   GO TO 70
70 IH=IH+1
   SM17=0.
   GO TO 9
71 CCH1=CH
72 SM17=SM17+5.
   GO TO 9
73 CCH2=CH
   IF (CCH1-CCH2) 75,75,74
74 CCH1=CCH2
   GO TO 72
75 SM17=SM17-1.
   DH=SM17+5.
   GO TO 9
76 CCH3=CH
   IF (CCH2-CCH3) 79,79,77
77 CCH2=CCH3
   GO TO 75
79 P8(IH)=(SM17+1.)/(SM11+SM15)
   GO TO 80
80 IF=IF+1
   SM12=0.
   GO TO 8
81 CCF1=CF
82 SM12=SM12+5.
   GO TO 8
83 CCF2=CF
   IF (CCF1-CCF2) 85,85,84
84 CCF1=CCF2
   GO TO 82
85 SM12=SM12-1.
   DF=SM12+5.
   GO TO 8
86 CCF3=CF
   IF (CCF2-CCF3) 89,89,87
87 CCF2=CCF3
   GO TO 85
89 P6(IF)=(SM12+1.)/(SM6+SM10)
   GO TO 90
90 IG=IG+1
   SM15=0.
   GO TO 7
```

```
 91 CCG1=CG
 92 SM15=SM15+5.
    GO TO 7
 93 CCG2=CG
    IF (CCG1-CCG2) 95,95,94
 94 CCG1=CCG2
    GO TO 92
 95 SM15=SM15-1.
    DG=SM15+5.
    GO TO 7
 96 CCG3=CG
    IF (CCG2-CCG3) 99,99,97
 97 CCG2=CCG3
    GO TO 95
 99 P7(JG)=(SM15+1.)/SM9
    GO TO 100
100 IF=IF+1
    SM10=0.
    GO TO 6
101 CCE1=CF
102 SM10=SM10+5.
    GO TO 6
103 CCE2=CF
    IF (CCE1-CCE2) 105,105,104
104 CCE1=CCE2
    GO TO 102
105 SM10=SM10-1.
    DF=SM10+5.
    GO TO 6
106 CCE3=CF
    IF (CCE2-CCE3)109,109,107
107 CCE2=CCE3
    GO TO 105
109 P5(IF)=(SM10+1.)/(SM8+SM4)
    GO TO 110
110 IC=IC+1
    SM5=0.
    GO TO 5
111 CCC1=CC
112 SM5=SM5+5.
    GO TO 5
113 CCC2=CC
    IF (CCC1-CCC2) 115,115,114
114 CCC1=CCC2
    GO TO 112
115 SM5=SM5-1.
    DC=SM5+5.
    GO TO 5
116 CCC3=CC
    IF (CCC2-CCC3) 119,119,117
117 CCC2=CCC3
    GO TO 115
```

```
119 P3(IC)=(SM5+1.)/SM3
    GO TO 120
120 ID=ID+1
    SM8=0.
    GO TO 4
121 CCD1=CD
122 SM8=SM8+5.
    GO TO 4
123 CCD2=CD
    IF (CCD1-CCD2) 125,125,124
124 CCD1=CCD2
    GO TO 122
125 SM8=SM8-1.
    DD=SM8+5.
    GO TO 4
126 CCD3=CD
    IF (CCD2-CCD3) 129,129,127
127 CCD2=CCD3
    GO TO 125
129 P4(ID)=(SM8+1.)/SM2
    GO TO 130
130 IB=IB+1
    SM3=0.
    GO TO 3
131 CCB1=CB
132 SM3=SM3+5.
    GO TO 3
133 CCB2=CB
    IF (CCB1-CCB2) 135,135,134
134 CCB1=CCB2
    GO TO 132
135 SM3=SM3-1.
    DB=SM3+5.
    GO TO 3
136 CCB3=CB
    IF (CCB2-CCB3) 139,139,137
137 CCB2=CCB3
    GO TO 135
139 P2(IB)=(SM3+1.)/SM1
    GO TO 140
140 IA=IA+1
    SM1=0.
    GO TO 2
141 CCA1=CA
142 SM1=SM1+5.
    GO TO 2
143 CCA2=CA
    IF (CCA1-CCA2) 145,145,144
144 CCA1=CCA2
    GO TO 142
145 SM1=SM1-1.
    DA=SM1+5.
```

```
      GO TO 2
  146 CCA3=CA
      IF (CCA2-CCA3) 149,149,147
  147 CCA2=CCA3
      GO TO 145
  149 P1(IA)=(SM1+1.)/SMA
      GO TO 150
  150 WRITE(3,160)M,P1(IA),P2(IB),P3(IC),P4(ID),P5(IE),P6(IF),P7(IG),
     1P8(IH),P9(II),CCA2
      M=M+1
  171 IF (P1(IA)-P1(IA-1)) 1,172,1
  172 IF (P2(IB)-P2(IB-1)) 1,173,1
  173 IF (P3(IC)-P3(IC-1)) 1,174,1
  174 IF (P4(ID)-P4(ID-1)) 1,175,1
  175 IF (P5(IE)-P5(IE-1)) 1,176,1
  176 IF (P6(IF)-P6(IF-1)) 1,177,1
  177 IF (P7(IG)-P7(IG-1)) 1,178,1
  178 IF (P8(IH)-P8(IH-1)) 1,179,1
  179 IF (P9(II)-P9(II-1)) 1,180,1
  180 STOP
      END
     MON$$       EXEQ LINKLOAD
                 CALL NC01
     MON$$       EXEQ NC01,MJB
     MON$$       JOB  ACT$$ D. K. PAI             IE          0313C40409
```

APPENDIX III

PR 115 PROGRAM FOR EXAMPLE 2 AND 3.

```
MON$$      JOB
MON$$      COMT 15 MINUTES,10 PAGES
MON$$      ASGN MJB,12
MON$$      ASGN MGO,16
MON$$      MODE GO,TEST
MON$$      EXEQ FORTRAN,,,,,,,NC01
      DIMENSIONPR1(40),PR2(40),PR3(40),PR4(40),PB1(40),PB2(40),PB3(40),
     1PB4(40)
  200 FORMAT(I2,8F9.5,F15.3)
  420 FORMAT (7F8.4)
      READ 420,RB11,RB12,RB2,RB8,RB69,RB10,RB1
      READ 420,RR1,RR2,RR3,RR4,RR56,RR8,RR11
      CK1=1.1
      CK2=.8
      CK3=1.5
      CK4=1.2
      CK5=.9
      CK6=1.3
      CK7=1.2
      CK8=1.6
      CK9=1.5
      CK10=1.3
      CK11=1.2
      CK12=1.0
      BK1=40.
      BK2=30.
      BK3=32.
      BK4=51.
      BK5=41.
      BK6=29.
      BK7=35.
      BK8=42.
      BK9=51.
      BK10=42.
      BK11=34.
      BK12=32.
      I=1
      J=1
      K=1
      L=1
      II=1
      JJ=1
      KK=1
      LL=1
      M=1
      SMB=100.
      SMR=100.
      PB1(1)=.5
      PB2(1)=.5
      PB3(1)=.5
      PB4(1)=.5
      PR1(1)=.5
```

```
      PR2(1)=.5
      PR3(1)=.5
      PR4(1)=.5
  300 DR7=0.
      DR6=0.
      DB11=0.
      DB12=0.
      DR7=0.
      DR6=0.
      DR3=0.
      DR1=0.
      MR1=PR1(I)*SMR+.5
      SR1=MR1
    1 SR2=SMR-SR1
      MB11=PB1(II)*SMB+.5
      SB11=MB11
    2 SB8=SMB-SB11
      MR3=PR3(K)*(SR1+RR1)+.5
      SR3=MR3
    3 SR5=SR1+RR1-SR3
      SR4=SR3+RR3
      MB12=PB3(KK)*(SB11+RB11)+.5
      SB12=MB12
    4 SB9=SB11+RB11-SB12
      SB10=SB12+RB12
      MR6=PR2(J)*(SR2+RR2)+.5
      SR6=MR6
    5 SR8=SR2+RR2-SR6
      SR11=SR8+RR8
      MB6=PB2(JJ)*(SB8+RB8)+.5
      SB6=MB6
    6 SB2=SB8+RB8-SB6
      SB1=SB2+RB2
      MR7=PR4(L)*(SR5+SR6+RR56)+.5
      SR7=MR7
    7 SR9=SR5+SR6+RR56-SR7
      SR10=SR7+SR4+RR4
      SR12=SR9+SR11+RR11
      MB7=PB4(LL)*(SB9+SB6+RB69)+.5
      SB7=MB7
    8 SB5=SB6+SB9+RB69-SB7
      SB4=SB7+SB10+RB10
      SB3=SB1+SB5+RB1
   10 SM1=SB1+SR1
      SM3=SB3+SR3
      SM6=SB6+SR6
      SM7=SB7+SR7
      SM11=SB11+SR11
      SM12=SB12+SR12
   11 IF (LL-L) 12,12,20
   12 IF (I-JJ) 13,13,23
   13 IF (JJ-J) 14,14,26
   14 IF (J-KK) 15,15,29
   15 IF (KK-K) 16,16,32
```

```
16 IF (K-II) 17,17,35
17 IF (II-I) 18,18,38
18 IF (I-M) 50,50,41
20 CBA1=BK7*SM7+CK7*SM7**2+BK3*SM3+CK3*SM3**2
   CBA2=BK4*(SR4+SB4)+CK4*(SR4**2+SB4**2)
   CBA3=BK5*(SR5+SB5)+CK5*(SR5**2+SB5**2)
   CBA=CBA1+CBA2+CBA3
   IF (SR7-DR7) 56,51,53
23 CBA1=BK7*SM7+CK7*SM7**2+BK12*SM12+CK12*SM12**2
   CBA2=BK9*(SR9+SB9)+CK9*(SR9**2+SB9**2)
   CBA3=BK10*(SR10+SB10)+CK10*(SR10**2+SB10**2)
   CBA=CBA1+CBA2+CBA3
   IF (SR7-DR7) 66,61,63
26 CBB1=BK6*SM6+CK6*SM6**2+BK7*SM7+CK7*SM7**2
   CBB2=BK1*SM1+CK1*SM1**2+BK3*SM3+CK3*SM3**2
   CBB3=BK2*(SR2+SB2)+CK2*(SR2**2+SB2**2)
   CBB4=BK5*(SR5+SB5)+CK5*(SR5**2+SB5**2)
   CBB5=BK4*(SR4+SB4)+CK4*(SR4**2+SB4**2)
   CBB=CBB1+CBB2+CBB3+CBB4+CBB5
   IF (SR6-DR6) 76,71,73
29 CBB1=BK6*SM6+CK6*SM6**2+BK7*SM7+CK7*SM7**2
   CBB2=BK11*SM11+CK11*SM11**2+BK12*SM12+CK12*SM12**2
   CBB3=BK8*(SR8+SB8)+CK8*(SR8**2+SB8**2)
   CBB4=BK9*(SR9+SB9)+CK9*(SR9**2+SB9**2)
   CBB5=BK10*(SR10+SB10)+CK10*(SR10**2+SB10**2)
   CBB=CBB1+CBB2+CBB3+CBB4+CBB5
   IF (SR6-DR6) 86,81,83
32 CBC1=BK12*SM12+CK12*SM12**2+BK7*SM7+CK7*SM7**2
   CBC2=BK3*SM3+CK3*SM3**2+BK9*(SR9+SB9)+CK9*(SR9**2+SB9**2)
   CBC3=BK10*(SR10+SB10)+CK10*(SR10**2+SB10**2)
   CBC4=BK4*(SR4+SB4)+CK4*(SR4**2+SB4**2)
   CBC5=BK5*(SR5+SB5)+CK5*(SR5**2+SB5**2)
   CBC=CBC1+CBC2+CBC3+CBC4+CBC5
   IF (SR12-DR12) 96,91,93
35 CBC1=BK12*SM12+CK12*SM12**2+BK7*SM7+CK7*SM7**2
   CBC2=BK3*SM3+CK3*SM3**2+BK9*(SR9+SB9)+CK9*(SR9**2+SB9**2)
   CBC3=BK10*(SR10+SB10)+CK10*(SR10**2+SB10**2)
   CBC4=BK4*(SR4+SB4)+CK4*(SR4**2+SB4**2)
   CBC5=BK5*(SR5+SB5)+CK5*(SR5**2+SB5**2)
   CBC=CBC1+CBC2+CBC3+CBC4+CBC5
   IF (SR3-DR3) 106,101,103
38 CBD1=BK1*SM1+CK1*SM1**2+BK3*SM3+CK3*SM3**2
   CBD2=BK6*SM6+CK6*SM6**2+BK7*SM7+CK7*SM7**2
   CBD3=BK11*SM11+CK11*SM11**2+BK12*SM12+CK12*SM12**2
   CBD4=BK2*(SR2+SB2)+CK2*(SR2**2+SB2**2)+BK5*(SR5+SB5)
   CBD5=CK5*(SR5**2+SB5**2)+BK4*(SR4+SB4)+CK4*(SR4**2+SB4**2)
   CBD6=BK8*(SR8+SB8)+CK8*(SR8**2+SB8**2)+BK9*(SR9+SB9)
   CBD7=CK9*(SR9**2+SB9**2)+BK10*(SR10+SB10)+CK10*(SR10**2+SB10**2)
   CBD=CBD1+CBD2+CBD3+CBD4+CBD5+CBD6+CBD7
   IF (SR11-DR11) 116,111,113
41 CBD1=BK1*SM1+CK1*SM1**2+BK3*SM3+CK3*SM3**2
   CBD2=BK6*SM6+CK6*SM6**2+BK7*SM7+CK7*SM7**2
```

```
      CRD3=BK11*SM11+CK11*SM11**2+BK12*SM12+CK12*SM12**2
      CRD4=BK2*(SR2+SB2)+CK2*(SR2**2+SB2**2)+BK5*(SR5+SB5)
      CRD5=CK5*(SR5**2+SB5**2)+BK4*(SR4+SB4)+CK4*(SR4**2+SB4**2)
      CRD6=BK8*(SR8+SB8)+CK8*(SR8**2+SB8**2)+BK9*(SR9+SB9)
      CRD7=CK9*(SR9**2+SB9**2)+BK10*(SR10+SB10)+CK10*(SR10**2+SB10**2)
      CRD=CRD1+CRD2+CRD3+CRD4+CRD5+CRD6+CRD7
      IF (SR1-DR1) 126,121,123
   50 LL=LL+1
      SB7=0.
      GO TO 8
   51 CCBA1=CBA
   52 SB7=SB7+5.
      GO TO 8
   53 CCBA2=CBA
      IF (CCBA1-CCBA2) 55,55,54
   54 CCBA1=CCBA2
      GO TO 52
   55 SB7=SB7-1.
      DB7=SB7+5.
      GO TO 8
   56 CCBA3=CBA
      IF (CCBA2-CCBA3) 59,59,57
   57 CCBA2=CCBA3
      GO TO 55
   59 PB4(LL)=(SB7+1.)/(SB6+SB9)
      GO TO 60
   60 L=L+1
      SR7=0.
      GO TO 7
   61 CCRA1=CRA
   62 SR7=SR7+5.
      GO TO 7
   63 CCRA2=CRA
      IF (CCRA1-CCRA2) 65,65,64
   64 CCRA1=CCRA2
      GO TO 62
   65 SR7=SR7-1.
      DR7=SR7+5.
      GO TO 7
   66 CCRA3=CRA
      IF (CCRA2-CCRA3) 69,69,67
   67 CCRA2=CCRA3
      GO TO 65
   69 PR4(L)=(SR7+1.)/(SR5+SR6)
      GO TO 70
   70 JJ=JJ+1
      SR6=0.
      GO TO 6
   71 CCRB1=CRB
   72 SR6=SR6+5.
      GO TO 6
   73 CCRB2=CRB
```

```
      IF (CCBB1-CCBB2) 75,75,74
   74 CCBB1=CCBB2
      GO TO 72
   75 SB6=SB6-1.
      DB6=SB6+5.
      GO TO 6
   76 CCBB3=CBB
      IF (CCBB2-CCBB3) 79,79,77
   77 CCBB2=CCBB3
      GO TO 75
   79 PB2(JJ)=(SB6+1.)/SB8
      GO TO 80
   80 J=J+1
      SB6=0.
      GO TO 5
   81 CCRB1=CRB
   82 SR6=SR6+5.
      GO TO 5
   83 CCRB2=CRB
      IF (CCRB1-CCRB2) 85,85,84
   84 CCRB1=CCRB2
      GO TO 82
   85 SR6=SR6-1.
      DR6=SR6+5.
      GO TO 5
   86 CCRB3=CRB
      IF (CCRB2-CCRB3) 89,89,87
   87 CCRB2=CCRB3
      GO TO 85
   89 PR2(J)=(SR6+1.)/SR2
      GO TO 90
   90 KK=KK+1
      SB12=0.
      GO TO 4
   91 CCBC1=CBC
   92 SB12=SB12+5.
      GO TO 4
   93 CCBC2=CBC
      IF (CCBC1-CCBC2) 95,95,94
   94 CCBC1=CCBC2
      GO TO 92
   95 SB12=SB12-1.
      DB12=SB12+5.
      GO TO 4
   96 CCBC3=CBC
      IF (CCBC2-CCBC3) 99,99,97
   97 CCBC2=CCBC3
      GO TO 95
   99 PB3(KK)=(SB12+1.)/SB11
      GO TO 100
  100 K=K+1
      SR3=0.
```

```
      GO TO 3
  101 CCRC1=CRC
  102 SR3=SR3+5.
      GO TO 3
  103 CCRC2=CRC
      IF (CCRC1-CCRC2) 105,105,104
  104 CCRC1=CCRC2
      GO TO 102
  105 SR3=SR3-1.
      DR3=SR3+5.
      GO TO 3
  106 CCRC3=CRC
      IF (CCRC2-CCRC3) 109,109,107
  107 CCRC2=CCRC3
      GO TO 105
  109 PR3(K)=(SR3+1.)/SR1
      GO TO 110
  110 II=II+1
      SB11=0.
      GO TO 2
  111 CCBD1=CBD
  112 SB11=SB11+5.
      GO TO 2
  113 CCBD2=CBD
      IF (CCBD1-CCBD2) 115,115,114
  114 CCBD1=CCBD2
      GO TO 112
  115 SB11=SB11-1.
      DB11=SB11+5.
      GO TO 2
  116 CCBD3=CBD
      IF (CCBD2-CCBD3) 119,119,117
  117 CCBD2=CCBD3
      GO TO 115
  119 PB1(II)=(SB11+1.)/SMB
      GO TO 120
  120 I=I+1
      SR1=0.
      GO TO 1
  121 CCRD1=CRD
  122 SR1=SR1+5.
      GO TO 1
  123 CCRD2=CRD
      IF (CCRD1-CCRD2) 125,125,124
  124 CCRD1=CCRD2
      GO TO 122
  125 SR1=SR1-1.
      DR1=SR1+5.
      GO TO 1
  126 CCRD3=CRD
      IF (CCRD2-CCRD3) 129,129,127
  127 CCRD2=CCRD3
```

```
      GO TO 125
  129 PR1(I)=(SR1+1.)/SMR
      GO TO 130
  130 WRITE(3,200)M,PR1(I),PR2(J),PR3(K),PR4(L),PB1(II),PB2(JJ),PB3(KK)
     1.PB4(LL),CCRD2
      M=M+1
      GO TO 140
  140 IF (PR1(I)-PR1(I-1)) 300,141,300
  141 IF (PR2(J)-PR2(J-1)) 300,142,300
  142 IF (PR3(K)-PR3(K-1)) 300,143,300
  143 IF (PR4(L)-PR4(L-1)) 300,144,300
  144 IF (PB1(II)-PB1(II-1)) 300,145,300
  145 IF (PB2(JJ)-PB2(JJ-1)) 300,146,300
  146 IF (PB3(KK)-PB3(KK-1)) 300,147,300
  147 IF (PB4(LL)-PB4(LL-1)) 300,150,300
  150 STOP
      END
    MON$$       EXEQ LINKLOAD
                CALL NC01
    MON$$       EXEQ NC01,MJB
    MON$$       JOB  ACT$$ D. K. PAI            IE          0313C40409
```

APPENDIX IV

FORGO PROGRAM FOR EXAMPLE 4.

```
C  C  PIPELINE PROBLEM
      DIMENSION P1(20),P2(20),P3(20),P4(20)
      R1=100.
      R2=0.
      R3=50.
      R4=0.
      R56=100.
      R8=50.
      R11=-50.
      CK1=1.8
      CK2=1.2
      CK3=1.9
      CK4=1.5
      CK5=1.1
      CK6=1.6
      CK7=1.4
      CK8=1.3
      CK9=1.4
      CK10=1.2
      CK11=1.2
      CK12=1.8
      BK1=51.
      BK2=55.
      BK3=67.
      BK4=58.
      BK5=61.
      BK6=70.
      BK7=63.
      BK8=65.
      BK9=58.
      BK10=45.
      BK11=65.
      BK12=65.
      II=1
      I=1
      J=1
      K=1
      L=1
      P1(1)=0.5
      P2(1)=0.5
      P3(1)=0.5
      P4(1)=0.5
   10 D1=.05
      D2=.05
      D3=.05
      D4=.05
    1 SMA=100.
      SM1=P1(I)*100.
      SM2=SMA-SM1
    2 SM3=P3(K)*(SM1+R1)
      SM5=SM1+R1-SM3
      SM4=SM3+R3
    3 SM6=P2(J)*(SM2+R2)
```

```
      SM8=SM2+R2-SM6
      SM11=SM8+R8
    4 SM7=P4(L)*(SM5+SM6+R56)
      SM9=(SM5+SM6+R56)-SM7
      SM10=SM7+SM4+R4
      SM12=SM9+SM11+R11
    5 IF (L-J) 1,6,11
    6 IF (J-K) 1,7,13
    7 IF (K-I) 1,8,15
    8 IF (I-II) 1,20,17
   11 CO1=BK7*SM7+CK7*SM7**2+BK9*SM9+CK9*SM9**2
      CO2=BK10*SM10+CK10*SM10**2+BK12*SM12+CK12*SM12**2
      CA=CO1+CO2
      IF (P4(L)-D1) 26,21,23
   13 CP1=BK6*SM6+CK6*SM6**2+BK8*SM8+CK8*SM8**2
      CP2=BK11*SM11+CK11*SM11**2+BK9*SM9+CK9*SM9**2
      CP3=BK7*SM7+CK7*SM7**2+BK10*SM10+CK10*SM10**2
      CP4=BK12*SM12+CK12*SM12**2
      CB=CP1+CP2+CP3+CP4
      IF (P2(J)-D2) 46,41,43
   15 CQ1=BK3*SM3+CK3*SM3**2+BK4*SM4+CK4*SM4**2
      CQ2=BK5*SM5+CK5*SM5**2+BK7*SM7+CK7*SM7**2
      CQ3=BK9*SM9+CK9*SM9**2+BK10*SM10+CK10*SM10**2
      CQ4=BK12*SM12+CK12*SM12**2
      CC=CQ1+CQ2+CQ3+CQ4
      IF (P3(K)-D3) 66,61,63
   17 CR1=BK1*SM1+CK1*SM1**2+BK2*SM2+CK2*SM2**2
      CR2=BK3*SM3+CK3*SM3**2+BK4*SM4+CK4*SM4**2
      CR3=BK5*SM5+CK5*SM5**2+BK6*SM6+CK6*SM6**2
      CR4=BK7*SM7+CK7*SM7**2+BK8*SM8+CK8*SM8**2
      CR5=BK9*SM9+CK9*SM9**2+BK10*SM10+CK10*SM10**2
      CR6=BK11*SM11+CK11*SM11**2+BK12*SM12+CK12*SM12**2
      CD=CR1+CR2+CR3+CR4+CR5+CR6
      IF (P1(I)-D4) 86,81,83
   20 L=L+1
      P4(L)=.05
      GO TO 4
   21 CA1=CA
   22 P4(L)=P4(L)+.05
      GO TO 4
   23 CA2=CA
      IF (CA1-CA2) 25,25,24
   24 CA1=CA2
      GO TO 22
   25 P4(L)=P4(L)-.01
      D1=D1+5.
      GO TO 4
   26 CA3=CA
      IF (CA2-CA3) 28,28,27
   27 CA2=CA3
      GO TO 25
   28 P4(L)=P4(L)+.01
```

```
   GO TO 40
40 J=J+1
   P2(J)=.05
   GO TO 3
41 CB1=CB
42 P2(J)=P2(J)+.05
   GO TO 3
43 CB2=CB
   IF(CB1-CB2) 45,45,44
44 CB1=CB2
   GO TO 42
45 P2(J)=P2(J)-.01
   D2=D2+5.
   GO TO 3
46 CB3=CB
   IF (CB2-CB3) 48,48.47
47 CB2=CB3
   GO TO 45
48 P2(J)=P2(J)+.01
   GO TO 60
60 K=K+1
   P3(K)=.05
   GO TO 2
61 CC1=CC
62 P3(K)=P3(K)+.05
   GO TO 2
63 CC2=CC
   IF (CC1-CC2) 65.65,64
64 CC1=CC2
   GO TO 62
65 P3(K)=P3(K)-.01
   D3=D3+5.
   GO TO 2
66 CC3=CC
   IF (CC2-CC3) 68.68,67
67 CC2=CC3
   GO TO 65
68 P3(K)=P3(K)+.01
   GO TO 80
80 I=I+1
   P1(I)=.05
   GO TO 1
81 CD1=CD
82 P1(I)=P1(I)+.05
   GO TO 1
83 CD2=CD
   IF (CD1-CD2) 85.85.84
84 CD1=CD2
   GO TO 82
85 P1(I)=P1(I)-.01
   D4=D4+5.
```

```
      GO TO 1
   86 CD3=CD
      IF (CD2-CD3) 88,88,87
   87 CD2=CD3
      GO TO 85
   88 P1(I)=P1(I)+.01
      PUNCH 100,II,P1(I),P2(J),P3(K),P4(L),CD3
      PRINT 100,II,P1(I),P2(J),P3(K),P4(L),CD3
  100 FORMAT (I2,2XF6.4,2XF6.4,2XF6.4,2XF6.4,2XF8.0)
      II=II+1
      GO TO 110
  110 IF (P4(L)-P4(L-1)) 10,111,10
  111 IF (P3(K)-P3(K-1)) 10,112,10
  112 IF (P2(J)-P2(J-1)) 10,113,10
  113 IF (P1(I)-P1(I-1)) 10,115,10
  115 STOP
      END
```

SOLUTION OF NETWORK FLOW PROBLEM BY DYNAMIC PROGRAMMING

by

DEEPAK KESHAV PAI

B. E. M. E., University of Poona, India, 1963
M. Tech. M. E., Indian Institute of Technology, Bombay, India, 1965

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Industrial Engineering

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1967

The purpose of this report is to demonstrate the application of dynamic programming to the network type traffic assignment and pipeline problems. This technique allows the use of nonlinear time-volume and cost-volume relationships.

A number of one way and two way traffic assignment and pipeline problems have been solved by this technique. The success of this technique lies in its simplicity, computational efficiency and selfcorrecting characteristics.