COMPUTER PROGRAM FOR AN OPTIMUM TRANSPORTATION SOLUTION BY THE DISCRETE MAXIMUM PRINCIPLE

by

PRADIT KONGKATONG

B.Sc. (Chemical Technology), Chulalongkorn University
Bangkok, Thailand, 1963

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

1968

Approved by:

Ching-Lai Hwang
Major Professor

TABLE OF CONTENTS

1. INTRODUCTION

In a large supermarket retail chain with several storing locations, one of the problems in supplying the individual stores is transportation, i.e., how much of a particular item should be sent each time and from what location should it be sent in order to satisfy the demands of all the stores, given the resources available at different storing locations.

Optimization of a transportation problem with linear cost function can be regarded as a generalization of the assignment problem, which can be carried out by means of the simplex method of linear programming (4). However, some special methods, such as the northwest corner method, the unit penalty method and Vogel's approximation method, have been developed which are easier to apply and are less tedious than the simplex method (1, 11). Optimization of transportation problems with non-linear cost functions cannot be solved by linear programming methods. The non-linear problem with two and three depots has been solved by dynamic programming (2). Recently, a discrete version of the maximum principle has been applied to the two-depot problem (5, 6) and the three-depot problem (7). This has resulted in a great simplification in numerical calculations.

The purpose of this report is to study and understand more thoroughly the application of the discrete version of the maximum principle in the transportation problem with non-linear cost function so as to enable us to modify the algorithm in Ref. (7) and to write computer program according to this algorithm.

In section 2 the transportation problem is defined and a table of transportation costs and requirements is presented. The modification of the discrete version of the maximum principle is developed to derive the Hamiltonian function for the transportation problem in section 3. Also in this section the numerical method in obtaining the optimal solution is illustrated. Section 4 deals with the explanation of the steps in the computer program.

2. STATEMENT OF THE PROBLEM

Suppose that there are a number of origins (depots) where the resource is located and N number of destinations (demand points) where the demand for this resource exists. Also suppose that there is only one type of resource and that its total supply is equal to the total demand. Let

θ_i^n = the quantity of the resource sent from the i-th depot (origin) to the n^{th} demand point, and

$F_i^n(\theta_i^n)$ = the cost incurred by this operation.

If there are s depots and N demand points, the problem is to determine the values of θ_i^n, i = 1, 2, ..., s; n = 1, 2, ..., N, so as to minimize the total cost of transporting the resources,

$$C_{sN} = \sum_{n=1}^{N} \sum_{i=1}^{s} F_i^n(\theta_i^n) \tag{1}$$

subject to the constraints.

(i) $\theta_i^n \geq 0$,

(ii) $\sum_{n=1}^{N} \theta_i^n = W_i$, number of units of the resource available at the i-th depot, i=1, 2, ..., s,

(iii) $\sum_{i=1}^{s} \theta_i^n = D^n$, number of units of the resource required by the n^{th} demand point, $n=1, 2, \ldots, N$.

A nonlinear cost function of the transportation problem is as follows:

$$F_i^n(\theta_i^n) = a_i^n\theta_i^n + b_i^n(\theta_i^n)^2 \tag{2}*$$

where a_i^n, b_i^n are constants. The values of a_i^n, b_i^n with D^n, the number of units of the resource required by the n^{th} demand point, and W_i, the number of units of the resource available at the i^{th} depot, are shown in Table 1.

*The superscript, n, indicates the stage number. The exponents are written with parentheses or brackets such as $(\theta^n)^2$ or $\{\phi(\theta^n)\}^2$.

Table 1. Transportation costs and requirements

Depots

	n \ i	1		2		3		D^n
		a_1^n	b_1^n	a_2^n	b_2^n	a_3^n	b_3^n	
Demand Points	1	2.5	0	1.0	-0.03	2.0	0	50
	2	2.5	-0.05	1.5	0	5.0	0	70
	3	3.5	0	3.0	0.02	1.0	0	100
	4	4.0	0.03	2.0	1.00	3.0	0	80
	W_i	130		90		80		300

3. SOLUTION BY THE DISCRETE MAXIMUM PRINCIPLE METHOD

Let the demand points be stages, and the total amount of resource which has been transported from the i^{th} depot to the first n stages (demand points) be state variables x_i^n, i = 1, 2, then

$$x_i^n = x_i^{n-1} + \theta_i^n, \quad x_i^o = 0, \quad x_i^4 = W_i, \tag{1}$$

$$i = 1, 2; \quad n = 1, 2, 3, 4.$$

It must be noted that, though there are 3 depots in the problem, there are only 2 state variables. This is because the demand of each stage is preassigned; hence, the number of the units supplied from the 3^{rd} depot to the n^{th} stage can be obtained by subtracting from the total number of units required by the n^{th} stage, the sum of the units supplied to the n^{th} stage from the first and second depot. This is equivalent to writing

$$\theta_3^n = D^n - \sum_{i=1}^{2} \theta_i^n, \quad n = 1, 2, 3, 4. \tag{2}$$

Since the objective is to minimize the total cost of transportation, we define this objective as the 3^{rd} state variable which satisfies the following performance equation:

$$x_3^n = x_3^{n-1} + \sum_{i=1}^{3} F_i^n(\theta_i^n), \quad x_3^o = 0, \tag{3}$$

$$n = 1, 2, 3, 4.$$

It can be shown that x_3^4 is equal to the total cost of transportation. Hence, the problem of minimizing the total cost of transportation becomes that of minimizing the final value of the 3rd state variable, x_3^4, by the proper choice of the sequence of θ_i^n, $i = 1, 2$; $n = 1, 2, 3, 4$ for the process described by equations (1) and (3).

The Hamiltonian function can be written as

$$H^n = \sum_{i=1}^{2} z_i^n(x_i^{n-1} + \theta_i^n) + z_3^n(x_3^{n-1} + \sum_{i=1}^{3} F_i^n(\theta_i^n)) ,$$

$$n = 1, 2, 3, 4. \tag{4}$$

The recursion relation for the components of the adjoint vector are found to be

$$z_i^{n-1} = \frac{\partial H^n}{\partial x_i^{n-1}} = z_i^n , \qquad i = 1, 2 \tag{5}$$

and

$$z_3^n = 1 , \qquad n = 1, 2, 3, 4 . \tag{6}$$

Since z_i^n and x_i^{n-1} are considered to be constants in searching for a stationary point or in finding a minimum point of the Hamiltonian function given by equation (4) with respect to θ_i^n, it is convenient to define the variable portion of the Hamiltonian function as

$$H_V^n = \sum_{i=1}^{2} z_i^n \theta_i^n + \sum_{i=1}^{3} F_i^n(\theta_i^n) . \tag{7}$$

Since the performance equations (1) and (3) are linear in the state variable x_i^{n-1} and the coefficients of the state variables are constants, the objective function is absolutely maximum (or minimum) if and only if H^n is absolutely maximum (or minimum). Therefore, the optimal sequence of the decision vector, θ^n, is obtained from equation (7) if and only if H^n is absolutely maximum (or minimum). By substituting equation (2-2) and (2) into equation (7) we have

$$H_V^n = (z_1^n + a_1^n - a_3^n - 2b_3^n D^n)\theta_1^n + (z_2^n + a_2^n - a_3^n - 2b_3^n D^n)\theta_2^n$$

$$+ (b_1^n + b_3^n)(\theta_1^n)^2 + (b_2^n + b_3^n)(\theta_2^n)^2 + a_3^n D^n$$

$$+ b_3^n (D^n)^2 + 2b_3^n \theta_1^n \theta_2^n .$$

But in our case $b_3^n = 0$ for $n = 1, 2, 3, 4$; therefore,

$$H_V^n = (z_1^n + a_1^n - a_3^n)\theta_1^n + (z_2^n + a_2^n - a_3^n)\theta_2^n$$

$$+ b_1^n(\theta_1^n)^2 + b_2^n(\theta_2^n)^2 + a_3^n D^n , \quad n = 1, 2, 3, 4 . \tag{8}$$

<u>Procedure for Optimal Allocation</u>:

Step 1. Calculate the breaking value z_i^n .

Stage 1:

The variable part of the Hamiltonian equation for the first demand point is

$$H_v^1 = (z_1^1 + 0.5)\theta_1^1 + (z_2^1 - 1)\theta_2^1 - .03(\theta_2^1)^2 + 100.$$

Therefore the breaking value of z_1^1 is -0.5. Taking the partial derivative of H_v^1 with respect to θ_2^1 and equating it to zero results in

$$\frac{\partial H_v^1}{\partial \theta_2^1} = 0 = z_2^1 - 1 - 0.06\ \theta_2^1\ .$$

However, the second derivative of H_v^1 with respect to θ_2^1 yields,

$$\frac{\partial^2 H_v^1}{\partial(\theta_2^1)^2} = -\ 0.06 < 0\ .$$

Therefore, from the condition $\frac{\partial H_v^1}{\partial \theta_2^1} = 0$, we cannot obtain a value for θ_2^1 which yields the minimum of H_v^1. The minimum of H_v^1 occurs at the boundary of the constraint of θ_2^1, as shown in Fig. 1. Therefore, the conditions for H_v^1 to be minimum are

$$H_v^1 = \text{min. at:}\quad \theta_2^1 = 0 \quad \text{if } z_2^1 \geq 2.5,$$

$$\theta_2^1 = 50 \text{ if } z_2^1 \leq 2.5.$$

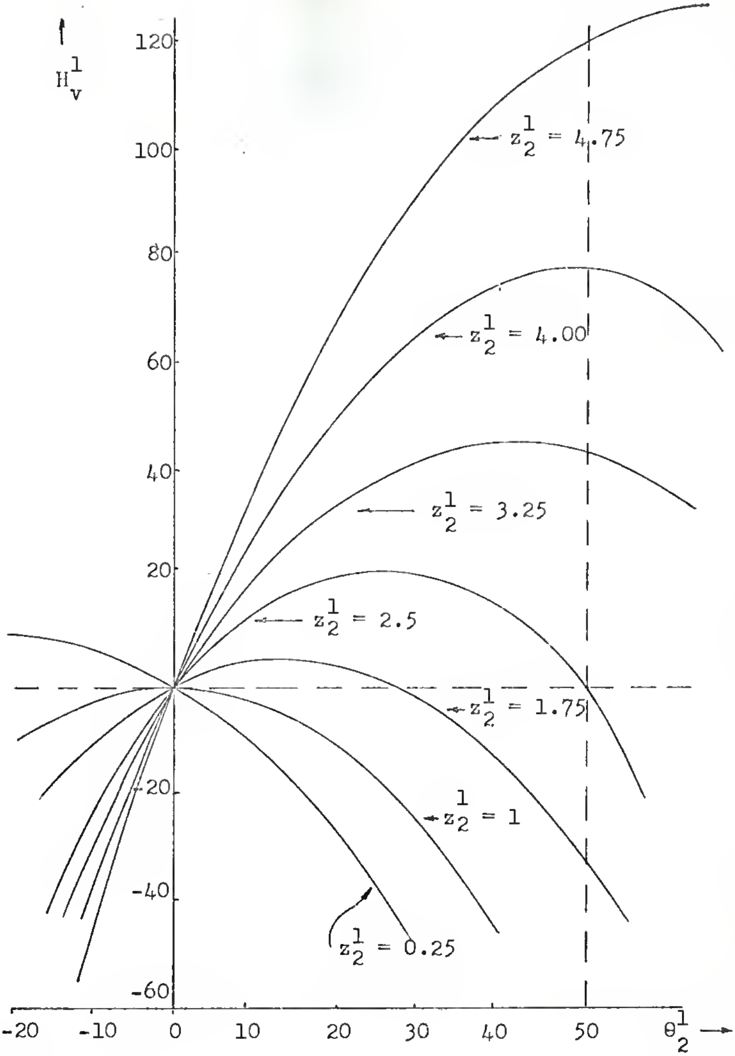


Fig. 1. H_v^1 v.s. θ_2^1

Thus, the so called breaking values of z_i^n are -0.5 and 2.5. The six conditions at which H_v^1 may be minimum are presented in Table 2.

Stage 2:

The variable part of the Hamiltonian equation for the second demand point (stage) is

$$H_v^2 = (z_1^2 - 2.5)\theta_1^2 + (z_2^2 - 3.5)\theta_2^2 - 0.05(\theta_1^2)^2 + 350 .$$

Therefore, the breaking value of z_2^2 is 3.5. Taking the partial derivative of H_v^2 with respect to θ_1^2 and equating it to zero results in

$$\frac{\partial H_v^2}{\partial \theta_1^2} = 0 = z_1^2 - 2.5 - 0.1\ \theta_1^2 .$$

However, the second derivative of H_v^2 with respect to θ_1^2 yields

$$\frac{\partial^2 H_v^2}{\partial (\theta_1^2)^2} = -0.1 < 0 .$$

Therefore, from the condition $\frac{\partial H_v^2}{\partial \theta_1^2} = 0$, we cannot obtain θ_1^2 which yields the minimum of H_v^2. The minimum of H_v^2 occurs at the boundary of the constraint of θ_1^2, with curve similar to Fig. 1. Therefore, the conditions of H_v^2 to be minimum are

Table 2. Conditions necessary for H_v to be minimum

n	Minimum of H_v^n occurs at			
	θ_1^n	θ_2^n	z_1^n	z_2^n
1	0	0	> -0.5	≥ 2.5
	$0 \leq \theta_1^1 \leq 50$	0	$= -0.5$	≥ 2.5
	50	0	< -0.5	≥ 2.5
	0	50	> -0.5	≤ 2.5
	0	50	$= -0.5$	≤ 2.5
	$\theta_1^1 = 0$ or 50	$\theta_2^1 = 0$ or 50	< -0.5	≤ 2.5
2	0	0	≥ 6	> 3.5
	70	0	≤ 6	> 3.5
	0	$0 \leq \theta_2^2 \leq 70$	≥ 6	$= 3.5$
	70	0	≤ 6	$= 3.5$
	0	70	≥ 6	< 3.5
	$\theta_1^2 = 0$ or 70	$\theta_2^2 = 0$ or 70	≤ 6	< 3.5
3	0	0	> -2.5	≥ -2
	$0 \leq \theta_1^3 \leq 100$	0	$= -2.5$	≥ -2
	100	0	< -2.5	≥ -2
	0	$-25\ z_2^3 - 50$	> -2.5	$-6 \leq z_2^3 < -2$
	$0 \leq \theta_1^3 \leq 100$	$-25\ z_2^3 - 50$	$= -2.5$	$-6 \leq z_2^3 < -2$
	$0 \leq \theta_1^3 \leq 100$	$-25\ z_2^3 - 50$	< -2.5	$-6 \leq z_2^3 < -2$
	0	100	> -2.5	< -6
	0	100	$= -2.5$	< -6
	$\theta_1^3 = 0$ or 100	$\theta_2^3 = 0$ or 100	< -2.5	< -6

Table 2. Conditions necessary for H_V to be minimum (cont.)

n	Minimum of H_V^n occurs at			
	θ_1^n	θ_2^n	z_1^n	z_2^n
4	0	0	≥ -1	> 1
	$-16.7\, z_1^4 - 16.7$	0	$-5.8 \leq z_1^4 < -1$	> 1
	80	0	< -5.8	> 1
	0	$-0.5\, z_2^4 + 1$	-1	$-159 \leq z_2^4 < 2$
	$-16.7\, z_1^4 - 16.7$	$-0.5\, z_2^4 + 1$	$-5.8 \leq z_1^4 < -1$	$-159 \leq z_1^4 < 2$
	$0 \leq \theta_1^4 \leq 80$	$-0.5\, z_2^4 + 1$	< -5.8	$-159 \leq z_2^4 < 2$
	0	80	≥ -1	< -159
	$-16.7\, z_1^4 - 16.7$	$0 \leq \theta_2^4 \leq 80$	$-5.8 \leq z_1^4 < -1$	< -159
	$\theta_1^4 = 0$ or 80	$\theta_2^4 = 0$ or 80	< -5.8	< -159

$$H_v^2 = \text{min. at: } \theta_1^2 = 0 \text{ if } z_1^2 \geq 6 ,$$

$$\theta_1^2 = 70 \text{ if } z_1^2 \leq 6 .$$

Thus, the breaking values of the z_i^n are 6 and 3.5. The six conditions at which H_v^2 may be minimum are presented in Table 2.

Stage 3:

The variable portion part of the Hamiltonian equation for the third demand point (stage) is

$$H_v^3 = (z_1^3 + 2.5)\theta_1^3 + (z_2^3 + 2)\theta_2^3 + .02(\theta_2^3)^2 + 100 .$$

The breaking value of z_1^3 is -2.5. Taking the partial derivative of H_v^3 with respect to θ_1^3 and equating it to zero results in

$$\frac{\partial H_v^3}{\partial \theta_2^3} = z_2^3 + 2 + .04\ \theta_2^3 = 0$$

or

$$\theta_2^3 = -25z_2^3 - 50 . \tag{9}$$

Therefore, the upper and lower breaking values of z_2^3 are

upper $z_2^3 = -2$, when $\theta_2^3 = 0$,

and lower $z_2^3 = -6$, when $\theta_2^3 = 100$.

Hence, H_v^3 is minimum at $\theta_2^3 = -25z_2^3 - 50$ for $- 6 \quad z_2^3 \quad -2$ as shown in Fig. 2. The nine conditions at which H_v^3 may be minimum are presented in Table 2.

Stage 4:

The variable part of the Hamiltonian equation for the third demand point (stage) is

$$H_v^4 = (z_1^4 + 1)\theta_1^4 + (z_2^4 - 1)\theta_2^4 + 0.03(\theta_1^4)^2 + (\theta_2^4)^2 + 240 .$$

Taking the partial derivative of H_v^4 with respect to θ_1^4 and equating it to zero results in

$$\frac{\partial H_v^4}{\partial \theta_1^4} = 0 = z_1^4 + 1 + 0.06\ \theta_1^4$$

or

$$\theta_1^4 = - 16.7\ z_1^4 - 16.7 . \tag{10}$$

Therefore, the upper and lower breaking values of z_1^4 are

upper $z_1^4 = - 1$, when $\theta_1^4 = 0$,

and lower $z_1^4 = - 5.8$, when $\theta_1^4 = 80$.

Hence, H_v^4 is minimum at $\theta_1^4 = - 16.7\ z_1^4 - 16.7$ for $- 5.8 \leq z_1^4 \leq - 1$ with curve similar to Fig. 2. Taking the partial derivative of H_v^4 with respect to θ_2^4 and equating it to zero results in

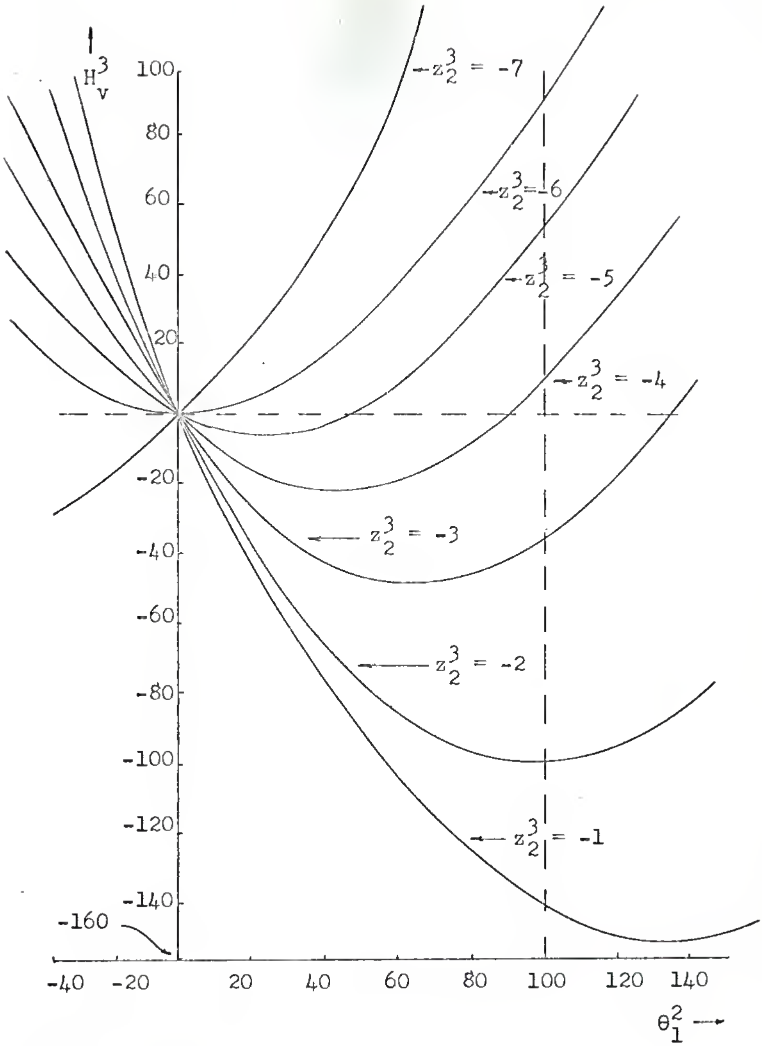


Fig. 2. H_v^3 v.s. θ_2^3.

$$\frac{\partial H_v^4}{\partial \theta_2^4} = 0 = z_2^4 - 1 + 2\,\theta_2^4$$

or

$$\theta_2^4 = -\,0.5\, z_2^4 + 0.5\ . \qquad (11)$$

Therefore, the upper and lower breaking values of z_2^4 are

upper $z_2^4 = 1$, when $\theta_2^4 = 0$,

and lower $z_2^4 = -159$, when $\theta_2^4 = 80$.

Hence H_v^4 is minimum at $\theta_2^4 = -\,0.5\, z_2^4$ for $-159 \leq z_2^4 \leq 1$ with curve similar to Fig. 2. The nine conditions at which H_v^4 may be minimum is given in Table 2. The conditions for all H_v^n to be minimized are summarized in Table 2. The breaking values of all the stages are summarized in Fig. 3a and Fig. 3b.

Step 2. Systematically searching

Each combination of the interior and/or the breaking values of z_1^n and z_2^n are systematically searched for a feasible solution; cases which do not satisfy the constraint $\sum_{n=1}^{4} \theta_1^n = W_i$ will be eliminated.

a) Region to be searched

In this searching procedure regions or points of the

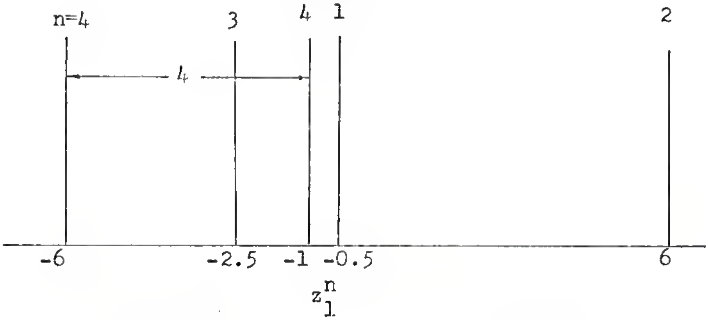


Fig. 3a. The breaking value of adjoint variable z_1^n

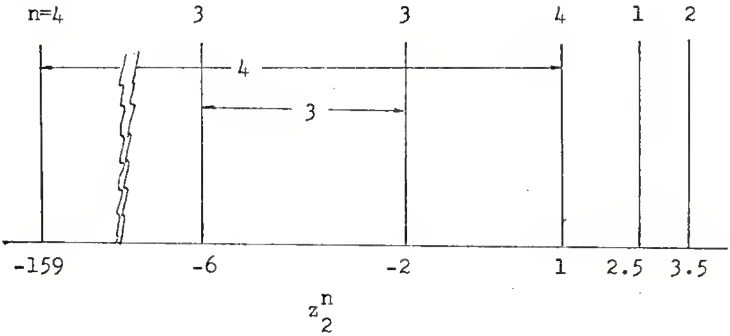


Fig. 3b. The breaking value of adjoint variable z_2^n.

breaking value of z_i^n are searched systematically by exhausting all possible combinations in the region of the breaking value.

b) Elimination of infeasible solution

Most of the infeasible solutions result from not satisfying the constraint we put into the system, i.e.,

$$\sum_{n=1}^{4} \theta_i^n = W_i, \quad i = 1, 2 .$$

Example 1. The combination of regions z_1 and of z_2 given by $-0.5 < z_1 \leq 6$; $2.5 \leq z_2 < 3.5$:

From Table 2, the tables of conditions necessary for H_v^n to be minimum, we have the conditions shown in Table 3. It is obvious that this is an infeasible case, since $\sum_{n=1}^{4} \theta_3^n =$

$50 + 0 + 100 + 80 = 230$ is larger than W_3, where $W_3 = 80$.

However, there are cases where the value z_i calculated from the equation of optimality (such as equations (9), (10), and (11) when the value of θ_i^n are known) does not fall in the region of search.

Example 2. The combination of region for z_1 and z_2 given by $-2.5 < z_1 < -1$; $-6 \leq z_2 < -2$:

From Table 2 we obtain the conditions necessary for H_v^1 to be minimum as shown in Table 4. The condition of $\theta_1^1 = 0$ or 50 and $\theta_2^1 = 0$ or 50 means that the minimum value of H_v^1 is either at $\theta_1^1 = 0$ and $\theta_2^1 = 50$ or $\theta_2^1 = 0$ and $\theta_1^1 = 50$.

Table 3. θ_i^n corresponding to the condition that z_1^n and z_2^n are in the region of $-0.5 < z_1^n \le 6$ and $2.5 \le z_2^n < 3.5$

n \ i	1	2	3	D^n
1	0	0	50	50
2	$\theta_1^2 = 0$ or 70	$\theta_2^2 = 0$ or 70	0	70
3	0	0	100	100
4	0	0	80	80
W_i	130	90	80	300

Table 4. θ_i^n corresponding to $-2.5 < z_1^n < -1$ and $-6 \le z_2^n < -2$

n \ i	1	2	3	D^n
1	$\theta_1^1 = 0$ or 50	$\theta_2^1 = 0$ or 50		50
2	$\theta_1^2 = 0$ or 70	$\theta_2^2 = 0$ or 70		70
3	0	$-25\ z_2^3 - 50$		100
4	$-16.7z_1^4 - 16.7$	$-0.5z_2^4 + 1$		80
W_i	130	90	80	300

By using equation (8) we can compare the value of H_v^1 for both cases as follows:

For $\theta_1^1 = 0,\ \theta_2^1 = 50$

upper $\quad H_v^1 = (z_2^1 + a_2^1 - a_3^1)\theta_2^1 + b_2^1(\theta_2^1)^2 + a_3^1\ D^1$

$$= (-2.1 + 1 - 2)\ 50 - .03\ (50)^2 + 2 \times 50$$

$$= 3.1 \times 50 - 75 + 100$$

$$= -\ 130$$

lower $\quad H_v^1 = (-5.9 + 1 - 2)50 - .03(50)^2 + 2 \times 50$

$$= -\ 6.0 \times 50 - 0.03 \times (50)^2 + 2 \times 50$$

$$= -\ 345 - 75 + 100$$

$$= -\ 370$$

For $\theta_1^1 = 50,\ \theta_2^1 = 0$

upper $\quad H_v^1 = (-\ 1.1 + 2.5 - 2)\ 50 + 2 \times 50$

$$= -0.6 \times 50 + 100$$

$$= -\ 30 + 100$$

$$= +\ 70$$

lower $\quad H_v^1 = (-\ 2.4 + 2.5 - 2)\ 50 + 2 \times 50$

$$= -\ 1.9 \times 50 + 2 \times 50$$

$$= +\ 10$$

It is obvious that $\theta_1^1 = 0$, $\theta_2^1 = 50$ gives a minimum, since the value H_v^1 is negative throughout the regions of z_2 while the value of H_v^1 when $\theta_1^1 = 50$, $\theta_2^1 = 0$ is positive throughout the region of z_1. Similarly, we assign $\theta_1^2 = 70$, $\theta_2^2 = 0$; therefore, $\theta_1^4 = W_1 - \sum_{n=1}^{3} \theta_1^n = 130 - 70 = 60$. By using the equation of optimality given by equation (10) we find that

$$-16.7\, z_1^4 - 16.7 = 60,$$

or

$$z_1^4 = \frac{60 + 16.7}{-16.7} = -\,4.6.$$

By equation (6) we have

$$z_1 = z_1^n = z_1^4 = -\,4.6 \quad \text{for} \quad n = 1,\ 2,\ 3,\ 4.$$

Since the region of z_1 we are now searching is $-\,2.5 < z_1 < -1$ and since the value of z_1 we just calculated is out of the region, this is an infeasible combination of the region of search.

c) Feasible solution

Feasible solutions are combinations of regions of z_i^n which satisfy all the constraints of the system and the value of z_i^n calculated from the equation of optimality is within the region.

Example 3. A feasible solution corresponding to the values of z_1^n at the point of $z_1^n = -\,2.5$ and z_2^n in the region of

$- 6 \leq z_2^n < - 2$ which satisfies the constraints is presented in Table 5.

By comparing H_v^1 as we did in example 2 we have $\theta_1^1 = 0$, $\theta_2^1 = 50$; similarly, we have $\theta_1^2 = 70$, $\theta_2^2 = 0$.

Since $z_1^n = z_1^4 = - 2.5$, from equation (10) we have

$$\theta_1^4 = - 16.7\ z_1^4 - 16.7$$
$$= - 16.7 \times (- 2.5) - 16.7$$
$$= 41.7 - 16.7 = 25$$

$$\theta_1^3 = W_1 - \theta_1^1 - \theta_1^2 - \theta_1^4$$
$$= 130 - 0 - 70 - 25$$
$$= 35.$$

From the constraint $W_2 = 90$ and $z_2 = z_2^3 = z_2^4$ we obtain

$$50 + 0 - 25\ z_2 - 50 - 0.5\ z_2 + 0.5 = 90$$

or

$$z_2 = - \frac{89.5}{25.5} = -3.51$$

which satisfies the condition, $- 6 \leq z_2^n < - 2$. The corresponding value of θ_2^3 obtained by substitution of $z_2^3 = - 3.51$ is

$$\theta_2^3 = - 25\ (-3.51) - 50$$
$$= 87.6 - 50 = 37.6$$

Table 5. θ_i^n corresponding to the condition that z_1^n and z_2^n are in the region of $z_1^n = -2.5$ and $-6 \leq z_2^n < -2$.

n \ i	1	2	3	D^n
1	$\theta_1^1 = 0$ or 50	$\theta_2^1 = 0$ or 50		50
2	$\theta_1^2 = 0$ or 70	$\theta_2^2 = 0$ or 70		70
3	$0 \leq \theta_1^3 \leq 100$	$-2.5\ z_2^3 - 50$		100
4	$-16.7\ z_1^4 - 16.7$	$-0.5\ z_2^4 + 0.5$		80
W_i	130	90	80	300

Table 6. θ_i^n corresponding to $z_1^n = -2.5$ and $-6 \leq z_2^n < -2$

n \ i	1	2	3	D^n
1	0	50	0	50
2	70	0	0	70
3	35	38	27	100
4	25	2	53	80
W_i	130	90	80	300

or

$$\theta_2^3 = 38$$

$$\theta_2^4 = -0.5\ (-3.51) + 0.5 = 2.25$$

or

$$\theta_2^4 = 2.$$

The value of θ_3^n can be found by using the constraint,

$$\sum_{i=1}^{s} \theta_i^n = D^n.$$

The itemized cost for the above solution is presented in Table 7, and the total cost is

$$\sum_{n=1}^{4} \sum_{i=1}^{3} F_i^n(\theta_i^n) = 171.25 + 125.8 + 186 = \$483.05.$$

Another feasible solution corresponding to the values of z_1^n in the region of $-5.8 \leq z_1^n < -2.5$ and z_2^n in the region $-6 \leq z_i^n < -2$ which satisfies the constraints is presented in Table 8.

The allocation of the value of θ_i^n and the calculation of transportation cost, similar to example 3, are shown in Tables 9 and 10.

The total transporation cost for this combination of regions is

$$\sum_{i=1}^{3} \sum_{n=1}^{4} F_i^n(\theta_i^n) = \$173.30 + 125.80 + 184.00 = \$483.10.$$

Table 7. Cost of transportation corresponding to $z_1^n = -2.5$

$-6 \le z_2^n < -2$

n \ i	1		2		3	
	$a_1^n\theta_1^n$	$b_1^n(\theta_1^n)^2$	$a_2^n\theta_2^n$	$b_2^n(\theta_2^n)^2$	$a_3^n\theta_3^n$	$b_3^n(\theta_3^n)^2$
1	0	0	50	-75	0	0
2	175	-245	0	0	0	0
3	122.5	0	114	28.8	27	0
4	100	18.75	4	4	159	0
	397.5	-226.25	168	-42.2	186	0
$\sum_{n=1}^{4} F_i^n(\theta_i^n)$	171.25		125.8		186	
$\sum_{n=1}^{4}\sum_{i=1}^{3} F_i^n(\theta_i^n)$	483.05					

Table 8. θ_i^n corresponding to the condition that z_1^n is in the region of $-5.8 \le z_1^n < -2.5$ and z_2^n is in the region of $-6 \le z_2^n < -2$

n \ i	1	2	3	D^n
1	$\theta_1^1 = 0$ or 50	$\theta_2^1 = 0$ or 50		50
2	$\theta_1^2 = 0$ or 70	$\theta_2^2 = 0$ or 70		70
3	$0 \le \theta_1^3 \le 100$	$-25\, z_2^3 - 50$		100
4	$-16.7\, z_1^4 - 16.7$	$-0.5\, z_2^4 + 0.5$		80
W_i	130	90	80	300

Table 9. θ_i^n corresponding to $-5.8 \leq z_1^n < -2.5$ and $-6 \leq z_2^n < -2$.

n \ i	1	2	3	D^n
1	0	50	0	50
2	70	0	0	70
3	34	38	28	100
4	26	2	52	80
W_i	130	90	80	300

Table 10. Cost of transportation corresponding to $-5.8 \leq z_1^n < -2.5$ and $-6 \leq z_2^n < -2$

n \ i	1		2		3	
	$a_1^n\theta_1^n$	$b_1^n(\theta_1^n)^2$	$a_2^n\theta_2^n$	$b_2^n(\theta_2^n)^2$	$a_3^n\theta_3^n$	$b_3^n(\theta_3^n)^2$
1	0	0	50	-75	0	0
2	175	-245	0	0	0	0
3	119	0	114	28.8	28	0
4	104	20.3	4	4	156	0
	398	-224.7	168	-42.2	184	
$\sum_{n=1}^{4} F_i^n(\theta_i^n)$	173.3		125.8		184	
$\sum_{n=1}^{4}\sum_{i=1}^{3} F_i^n(\theta_i^n)$	483.10					

Step 3. The optimal solution is obtained by comparing the total transportation cost of all the feasible regions. In our case there are only two feasible regions. Since the total transportation cost corresponding to $z_1^n = -2.5$ and $-6 \leq z_2^n < -2$ is less than that of $-5.8\, z_1^n < -2.5$ and $-6 \leq z_2^n < -2$, the optimal solution is θ_i^n corresponding to $z_1^n = -2.5$ and $-6 \leq z_2^n < -2$ which is shown in Table 6, and the minimum total transportation cost - \$483.05.

4. COMPUTING PROCEDURE

Step 1. Read in data: the values of a_i^n, b_i^n, W_i, and D^n.

Step 2. Calculate the breaking value of z_i^n.

The equation used for the calculation depends on the value of b_i^n as follows:

a) For $b_i^n < 0$, the secondary derivative of H_v^n with respect to θ_i^n will be negative. Therefore, from the condition $\frac{\partial H_v^n}{\partial \theta_i^n} = 0$, we cannot obtain a θ_i^n which yields the minimum of H_v^n. The minimum of H_v^n occurs at the boundary of the constraint of θ_i^n as shown in Fig. 1; the equation to be used in this case is as follows:

$$z_i^n = - a_i^n + a_s^n - 2\, b_i^n \frac{D^n}{2} \tag{12}$$

b) For $b_i^n = 0$, the cost function is linear, and the minimum will occur at the boundary of the constraint of θ_i^n. The breaking value is given by

$$z_i^n = - a_i^n + a_s^n \,. \tag{13}$$

c) For $b_i^n > 0$, the cost of transportation is non-linear and the second derivative of H_v^n with respect to θ_i^n is larger than zero. The shape of the curve of H_v^n verses θ_i^n is as shown in Fig. 2. The minimum of H_v^n can be obtained by equating the derivative of H_V^n with respect to θ_i^n to zero. The lower breaking value (Lower z_i^n) and the upper breaking value (Upper z_i^n) are at the boundary of the constraint, that is, $\theta_i^n = D^n$ and $\theta_i^n = 0$ respectively, or

$$\text{Lower } z_i^n = - a_i^n + a_s^n - 2b_i^n D^n , \tag{14}$$

$$\text{Upper } z_i^n = - a_i^n + a_s^n . \tag{15}$$

Step 3. Define the regions of search according to the breaking values.

This step consists of three substeps.

a) Rearrange the upper breaking value, z_i^n, in an order of largeness and represent it by ZBU_i^j, where $j = 1, 2, \ldots,$ indicate the order of largeness. The lower breaking value of the n^{th} stage will use the same j value as the upper breaking value of that stage and represent it by ZBL_i^j.

b) Assign the upper limiting value of regions to be searched. The upper limits of regions to be searched are assigned according to the value of bb_i^j (the b_i^n value in the new index) as follows:

$$(1) \quad bb_i^j < 0, \quad ZOP_i^k = ZBU_i^j - 0.005 \tag{16}$$

$$(2) \quad bb_i^j = 0, \quad ZOP_i^k = ZBU_i^j \tag{17}$$

$$ZOP_i^{k+1} = ZBU_i^j - 0.005 \tag{18}$$

(3) $bb_i^j > 0 \quad ZOP_i^k = ZBU_i^j - 0.005$ (19)

$$ZOP_i^{k+1} = ZBL_i^j - 0.005 \tag{20}$$

The value 0.005 is a small increment between the lower limit of the previous region and the upper limit of the current region.

In the search for a feasible solution, the upper limit of each region will serve temporarily as the z_i value in the first trial.

k, the new index for the regions, follows the order of the j index. Therefore, ZOP_i^k, k = 1, 2, ..., 7, is not in order according to the largeness of the value of ZOP_i^k.

c) Rearrange ZOP_i^k, k = 1, 2, ..., 7, in an order of largeness of the value of ZOP_i^k.

The results of step 3 are shown in Figs. 4a and 4b.

Step 4. The first allocation of θ_i^n.

The optimal value of θ_i^n which corresponds to the value of z_i, given (ZOP_i^k), i = 1, 2, can be classified into the following 5 cases (see Table 2)

a) $\theta_i^n = 0, \qquad ZOP_i^k > z_i^n$.

b) $\theta_i^n = -\ (ZOP_i^k + a_i^n - a_s^n)/(2\ b_i^n)$,

$$\text{Lower } z_i^n \leq ZOP_i^k < \text{Upper } z_i^n .$$

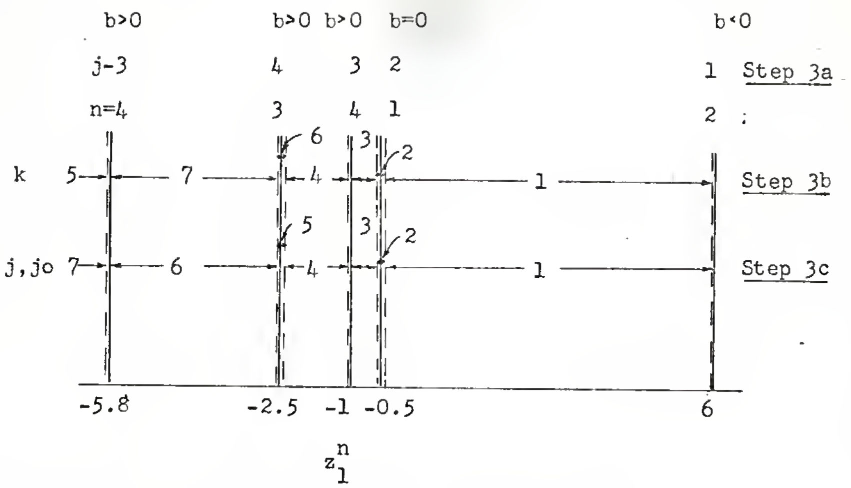


Fig. 4a. The region or point of search of z_1.

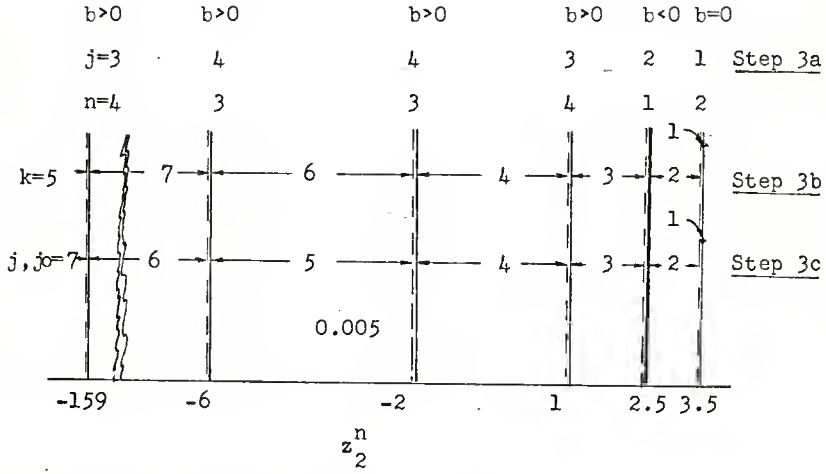


Fig. 4b. The region or point of search of z_2.

c) $\theta_i^n = D^n$, $\quad ZOP_i^k < z_i^n$.

d) $0 \leq \theta_i^n \leq D^n$, $\quad ZOP_i^k = z_i^n$. In computer programming the optimal value θ_i^n is represented by $D^n + 1$, that is, $\theta_i^n = D^n + 1$ at this step.

e) $\theta_i^n = D^n - [-(ZOP_{ia}^k + a_{ia}^n - a_s^n)/2b_{ia}^n]$,

$ZOP_i^k < z_i^n$ and

Lower $z_{ia}^n \leq ZOP_{ia}^k <$ Upper z_{ia}^n . In computer programming the optimal value θ_i^n is represented by $D^n + 2$, that is $\theta_i^n = D^n + 2$ at this step.

Step 5. Preliminary elimination of infeasible combinations of regions of search.

Most of the combinations of the region of search are elinimimated in this step. This is done by using the optimal values of θ_i^n which have been allocated in Step 4.1a and 4.1c ($\theta_i^n = 0$, $\theta_i^n = D^n$), and the constraint ($W_i = \sum_{n=1}^{N} \theta_i^n$). The two conditions that give the infeasible regions are

a) $\sum_{n=1}^{N} \theta_i^n > W_i$; this may occur when we have $\theta_i^n = D^n$ for several n's in the i^{th} origin, and

b) $\sum_{n=1}^{N} \theta_i^n < W_i$; this may occur when we have $\theta_i^n = 0$ for many n's in the i^{th} origin, even though we assign the maximum value (D^n) for the rest of n's.

Step 6. The second allocation of θ_i^n: $D^n + 1$, $D^n + 2$, $\theta_i^n = - (ZOP_i^j + a_i^n - a_s^n)/2b_i^n$.

This step is divided into 7 substeps.

1) The identification of the different conditions of optimality.

In each origin the number of occurences of each type of optimal condition classified in Step 4 is counted. The type of optimal condition at each stage is also identified. The counters and indexes for different types of conditions of optimality are shown in Table 11.

2) Find the available resource for the unassigned stages.

The difference between the total resources available at each origin or depot (W_i) and the sum of all $\theta_i^n = D^n$ assigned (SUM D_i) is the available resource for the unassigned stages (θ SUM). It can be put into computer equations as follows:

$$\text{SUM } D_i = \text{SUM } D_i + D^n \tag{21}$$

$$\theta \text{ SUM} = W_i - \text{SUM } D_i \tag{22}$$

Table 11. Indicator in computer program

Type of Condition	Kind of Condition	Temporary Value	No. of Types at each Origin	Sum of $\sum_{i=1}^{2}$ ke etc.	Index to identify stage
4.1b	Equation of optimality	$0 \leq \theta_i^n \leq D^n$	ke	ckk	me_{ke}
4.1e	Severely Constrained	$\theta_i^n = D^n+2$	kc,kk	p	mk_{kc}
4.1d	Free varied of Theta	$\theta_i^n = D^n+1$	k	q	m_k

When the assigned stages include the temporary value of θ_i^n = equation of optimality and $\theta_i^n = D^n$, the sum of the resource of the assigned stages (SUM_i) will be the sum of all $\theta_i^n = D^n$ and $\theta_i^n = -(z_i^n + a_i^n - a_s^n)/(2b_i^n)$. The available resource for the unassigned stages (TH_i) will be the difference between the total resource available at that origin (W_i) and the sum of the resource of the assigned states (SUM_i), which can be written in computer equations as follows:

$$SUM_i = SUM_i + D^n \tag{23}$$

$$SUM_i = SUM_i + \theta_i^n \tag{24}$$

$$TH_i = W_i - SUM_i \tag{25}$$

When the assigned stages include the temporary θ_i^n value ($\theta\theta_i^n$) of the severely constrained condition and $\theta_i^n = D^n$, the available resource for the unassigned stages (θS_i) will be the difference between the total resource available at that origin (W_i) and the sum of the resource of the assigned stages ($SUM\ \theta\theta_i$). It can be written in computor equations as follows:

$$SUM\ \theta\theta_i = SUM\ \theta\theta_i + D^n \tag{26}$$

$$SUM\ \theta\theta_i = SUM\ \theta\theta_i + \theta\theta_i^n \tag{27}$$

$$\theta S_i = W_i - SUM\ \theta\theta_i \tag{28}$$

3) Allocation for θ_i^n which is given by the equation of optimality

$$\theta_i^n = - (z_i^n + a_i^n - a_s^n)/2b_i^n \; . \tag{29}$$

The z_i^n value used in finding the optimal θ_i^n may be divided into 5 catagories.

a) At the point of search. When there is a condition of free variation of θ_i^n in the origin, the ZUP_i^n should be used immediately and is fixed.

b) In the region of search. In this case we have to search a whole region or a particular interval of z_i^n value to find the optimal value of z_i^n. The equations for finding the optimal z_i^n depend upon the number of equations of optimality in the same origin as follows:

(1) One equation of optimality condition (ke = 1).

$$\theta_i^n = \frac{z_i^{im} + (a_i^{im} - a_3^{im})}{2\, b_i^{im}} = - (W_i - \sum_{\substack{n=1 \\ n \neq im}}^{N} D^n) = - \theta SUM \tag{30}$$

or

$$z_i^{im} = - 2\, b_i^{im}\, \theta SUM - (a_i^{im} - a_s^{im}) \; .$$

(2) Two equations of optimality conditions (ke = 2).

$$\frac{z_i^{im} + (a_i^{im} - a_s^{im})}{2\, b_i^{im}} + \frac{z_i^{jm} + (a_i^{jm} - a_s^{jm})}{2\, b_i^{jm}} = - (W_i - \sum_{\substack{n=1 \\ m \neq im \\ n \neq jm}}^{N} D^n)$$

$$= - \theta SUM \; ,$$

or

$$z_i = z_i^{im} = z_i^{jm} = \frac{2b_i^{im} b_i^{jm} \theta SUM - b_i^{jm}(a_i^{im} - a_s^{im}) - b_i^{im}(a_i^{jm} - a_s^{jm})}{(b_i^{im} + b_i^{jm})} . \tag{31}$$

(3) Three equations of optimality conditions (ke = 3).

$$\frac{z_i^{im} + (a_i^{im} - a_s^{im})}{2\, b_i^{im}} + \frac{z_i^{jm} + (a_i^{jm} - a_s^{jm})}{2\, b_i^{jm}} + \frac{z_i^{km} + (a_i^{km} - a_s^{km}}{2\, b_i^{km}}$$

$$= - (W_i - \sum_{\substack{n=1 \\ n \neq im,jm,km}}^{N} D^n) = -\theta SUM \quad ,$$

or

$$z_i = z_i^{im} = z_i^{jm} = z_i^{km} = \frac{-2b_i^{im} b_i^{jm} b_i^{km} \theta SUM - b_i^{jm} b_i^{km}(a_i^{im} - a_s^{im})}{b_i^{im} b_i^{jm} + b_i^{jm} b_i^{km}}$$

$$\frac{-b_i^{im} b_i^{km}(a_i^{jm} - a_s^{jm}) - b^{im} b^{jm}(a_i^{km} - a_s^{km})}{+b_i^{im} b_i^{km}} . \tag{32}$$

(4) Four equations of optimality conditions (ke = 4).

$$\frac{z_i^{im} + (a_i^{im} - a_s^{im})}{2\, b_i^{im}} + \frac{z_i^{jm}(a_i^{jm} - a_s^{jm})}{2\, b_i^{jm}} + \frac{z_i^{km} + (a_i^{km} - a_s^{km})}{2\, b_i^{km}}$$

$$+ \frac{z_i^{lm} + (a_i^{lm} - a_s^{lm})}{2\, b_i^{lm}} = -\left(W_i - \sum_{\substack{n=1 \\ n \neq im, jm, km, lm}}^{N} D^n\right) = -\theta SUM \quad ,$$

$$z_i = z_i^{im} = z_i^{jm} = z_i^{km} = z_2^{lm}$$

$$= -(2 b_i^{im} b_i^{jm} b_i^{km} b_i^{lm} \theta SUM + b_i^{jm} b_i^{km} b_i^{lm} (a_i^{im} - a_s^{im})$$

$$+ b_i^{im} b_i^{km} b_i^{lm} (a_i^{jm} - a_s^{jm}) + b_i^{im} b_i^{jm} b_i^{lm} (a_i^{km} - a_s^{km})$$

$$+ b_i^{im} b_i^{jm} b_i^{km} (a_i^{lm} - a_s^{lm}))/ \; b_i^{im} b_i^{jm} b_i^{km} + b_i^{im} b_i^{jm} b_i^{lm}$$

$$+ b_i^{im} b_i^{km} b_i^{lm} + b_i^{jm} b_i^{km} b_i^{lm} \quad . \qquad (33)$$

Similarly any number of equations of optimality condition in the same origin can be derived.

c) Upper limiting value in the region of search. When there is a severely constrained condition (4.e) in the origin and $\theta S_i \leq 0$, the maximum value of z_i in the interval of the region that we are searching is used. This value corresponds to the minimum value of θ_i^n that can be allocated to stage n in this region of search. This occurs when θS_i from equation (28) is less than or equal to zero.

d) The region of search with severe constraint. When there is a severely constrained condition in the origin and the value ΘS_i from equations (28) is larger than zero, z_i^n of the equation of optimality (29) can be calculated from (6.3b) instead of (6.3c) and the value z_i^n can be anywhere within the region.

e) Iterated value of z_i^n. When there is more than one origin with the property of (6.3d) we have a different series of values of Θ_i^n each time we repeat the DO-loop of step 6. This is because the value z_i is changing each time we complete the DO-loop of step 6. Therefore the DO-loop of step 6 has to be repeated until we get a set of constants, z_i^n, which give us the optimal Θ_i^n value for this combination of regions of search. When this is reached the triggering counter (ckk) will decrease to zero and the next substep will be executed.

4) Allocation for Θ_i^n with severely constrained condition case.

The reason this case has a severely constrained condition is that the condition for optimal Θ_i^n is $\Theta_i^n = D^n$, but it is constrained by the priority we give to the equation of optimality in the preceding or the next origin. If this condition is nowhere present in the total system at the current combination of regions of search (p=0), this substep will automatically skip to the next substep.

The maximum allowable, available resource is allocated to Θ_i^n in this case so that the value of Θ_i^n is as close to the optimal condition as possible. This is done by checking both constraints

$$\theta_i^{imk} = D^n - \theta_{ia}^n \tag{35}$$

and

$$\theta_i^{imk} = W_i - \sum_{\substack{n=1 \\ n \neq imk}}^{N} \theta_i^n \tag{36}$$

and choosing the smaller of these two values. This value is the maximum allowable, available amount of resource.

5) Allocation of θ_i^n with free variation of θ_i^n condition.

This is the last type of θ_i^n in the active origin to be allocated. Since the optimal value can be anything from 0 to D^n, it also serves as an allocation to satisfy the constraint

$$\sum_{n=1}^{N} \theta_i^n = W_i \ .$$

When this property is not present at the region of search (q=0), this substep will automatically skip to the next substep.

6) Control of Allocation. The control of the allocation serves two functions:

(1) It servcs to check if the optimal z_i is within the interval of the region of search, and

(2) as a trigger mechanism for substeps (6.3), (6.4), and (6.5) which can be described as follows: For substep (6.3) when the counter is triggered, (ckk=0) the search for optimal θ_i^n is complete for this substep and it will go back to step 6 again. This time the substep (6.4) is executed if $p > 0$ but if $p=0$ the next substep (6.5) will be executed if $q > 0$ and if $q=0$, the computer will go to substep (6.7)

7) Allocation of constrained origin (s^{th}-depot)

The reason we called this a constrained origin is that the number of the units supplied from the constrained origin (s^{th}-depot) to the n^{th} stage can be obtained by substracting from the total number of units required by the n^{th} stage the sum of the units supplied to the n^{th} stage from the first through $(s-1)^{th}$ depot (or active state). This is equivalent to writing

$$\theta_s^n = D^n - \sum_{i=1}^{s-1} \theta_i^n = D^n - \theta_1^n - \theta_2^n .$$

Step 7. Calculate total cost of transportation

The cost of all feasible solutions is calculated by substituting the optimal θ_i^n into the cost function equation, and summing all the costs of transportation of each feasible solution.

Step 8. Compare the total cost with that of the previous minimum.

The total cost of transportation of the new feasible solution is compared with that of the previous one. The new minimum is then stored for future comparison.

Step 9. Punch out output on cards

All the values of the feasible solution are punched on cards, including θ_i^n, $i = 1, 2, 3$ and $n = 1, 2, 3, 4$; the region of search of the new minimum; the up-dated minimum cost; and the current total cost.

ACKNOWLEDGEMENTS

The author is indebted to Dr. C. L. Hwang, his major professor, for his valuable guidance, tremendous devotion, assistance and constructive criticism in preparing this report. The support and encouragement provided by Dr. F. A. Tillman, Head of the Department, and Dr. L. E. Grosh is sincerely acknowledged. Finally, the author wishes to thank all the staff of the computer center for their assistance in debugging the computer program.

REFERENCES

1. Bowman, E. H., and Fetter, R. B., "Analysis for Production Management," Richard D. Irwin, Inc., 1961.

2. Bellman, R. E., and Dreyfus, S. E., "Applied Dynamic Programming," Princeton University Press, Princeton, New Jersey, 1962.

3. Chang, S. S. L., "Digitized Maximum Principle," Proceedings of I.R.E., 2030-2031, Dec., 1960.

4. Dantzig, C. B., "Linear Programming and Extensions," Princeton University Press, Princeton, New Jersey, 1963.

5. Fan, L. T., and Wang, C. S., "The Discrete Maximum Principle--A Study of Multistage System Optimization," John Wiley & Son, Inc., New York, 1964.

6. Fan, L. T., and Wang, C. S., "The Application of the Discrete Maximum Principle to Transportation Problem," J. Math. & Physics, 43, 255 (1964).

7. Hwang, C. L., Schrader, G. F., Panchal, J. M., Fan, L. T., and Chen, S. K., "The Application of the Discrete Maximum Principle to Transportation Problems with Linear and Nonlinear cost Functions," Special Report No. 75, Kansas Engineering Experiment Station, Nov. 1966.

8. Katz, S., "Best Operating Points for Staged Systems" I.&E.C. Fundamentals, Vol. 1, No. 4, Nov., 1962.

9. Pontriagin, L. S., "Some Mathematical Problem Arising in Connection with the Theory of Optimum Automatic Control System," (in Russian), Session of The Academy of Sciences of The USSR on Scientific Problems of Automating Industry, October 15-20, 1956.

10. Rozonoer, L. I., "The Maximum Principle of L. S. Pontryagin in Optimal-System Theory," Automat, Telemech., Moscow, 20, 1320, 1441, 1561, 1960.

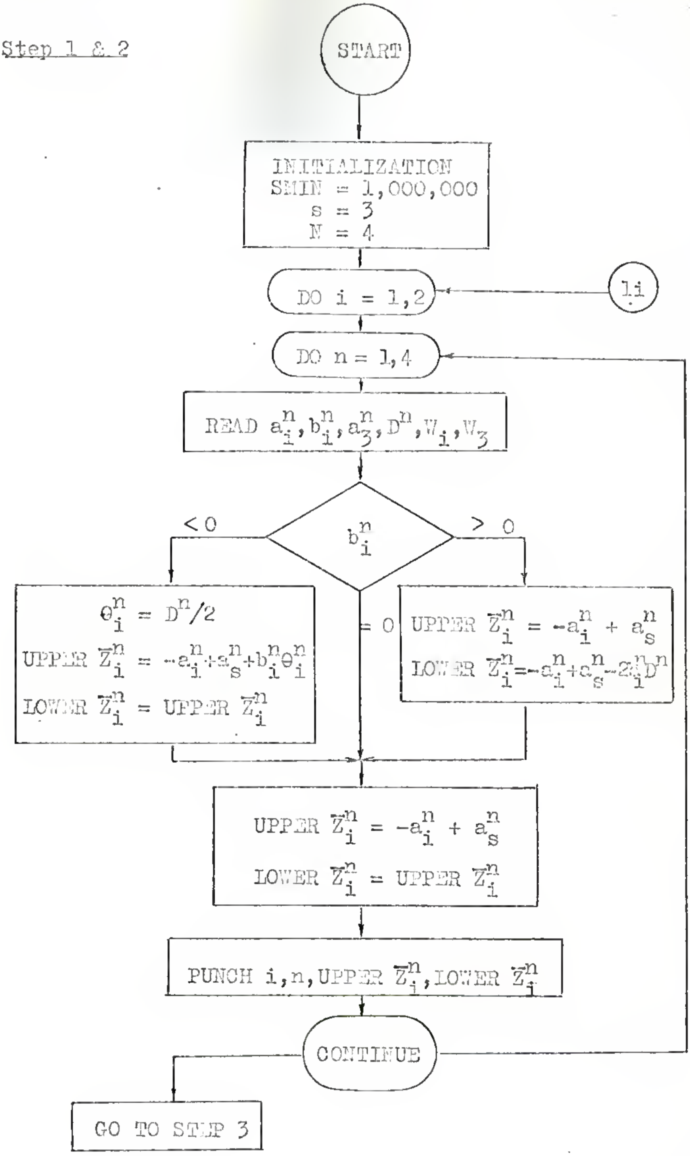
11. Sasieni, M., Yaspan, A., and Friedman, L., "Operation Research Methods," John Wiley & Son, Inc., New York, 1961.

APPENDIX A COMPUTER FLOW CHART

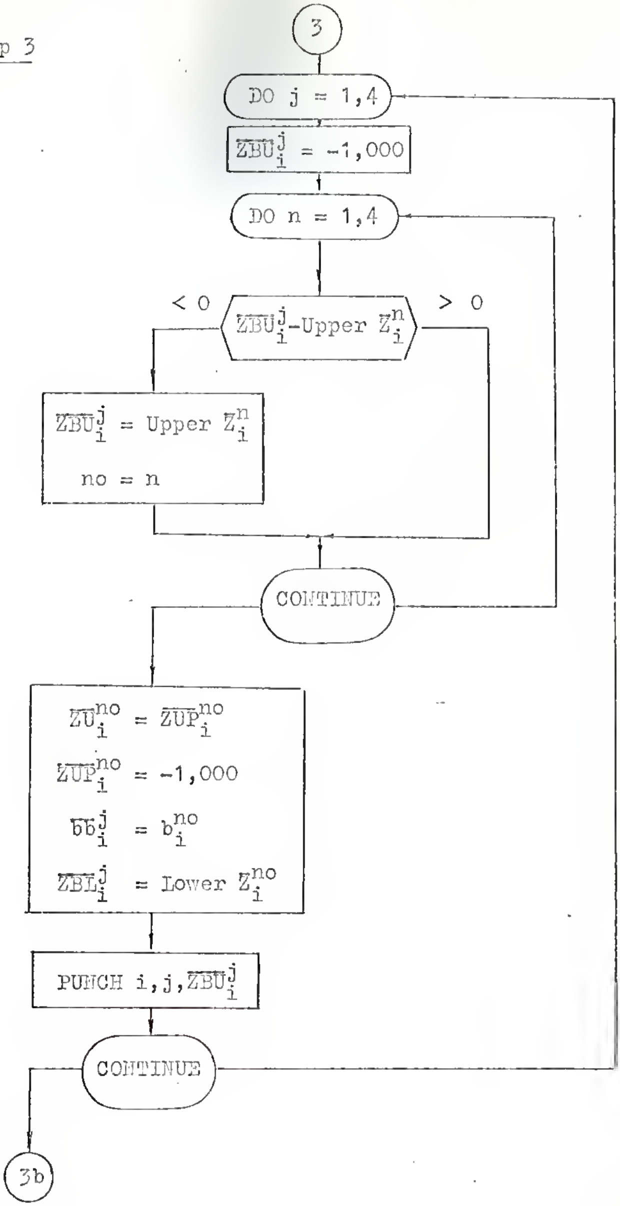
The steps of the computer program are:

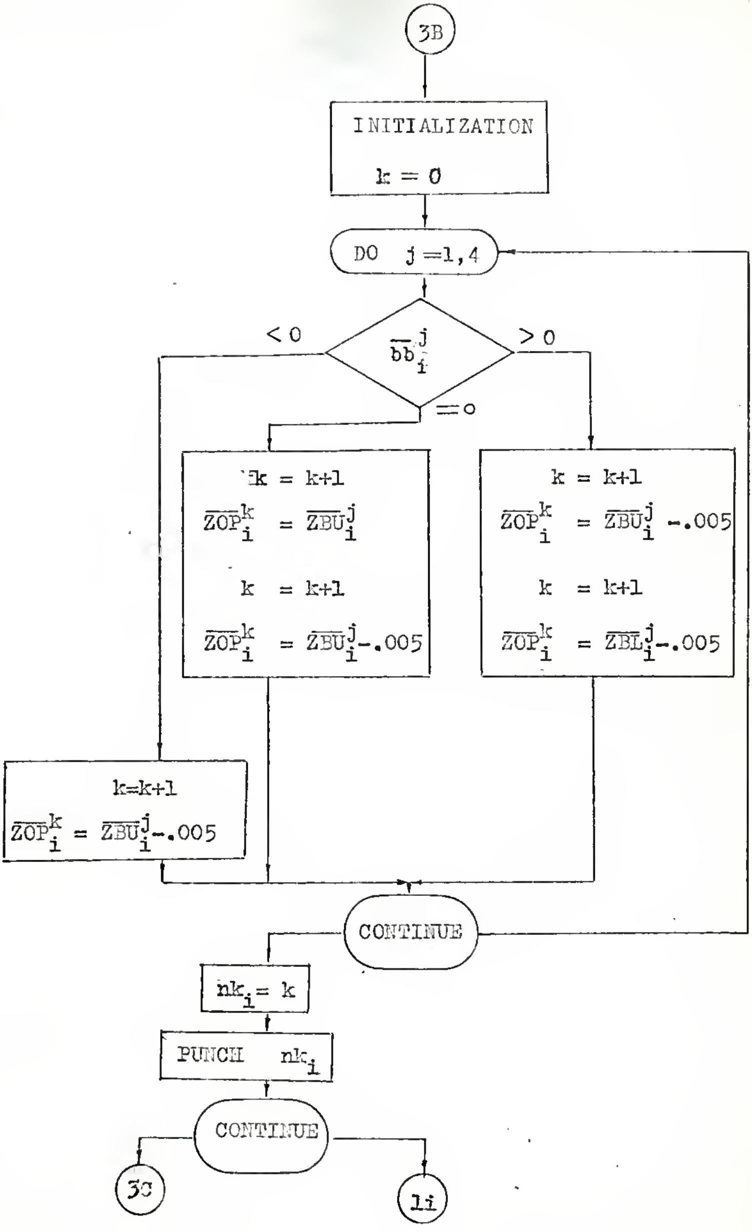
1. Read in a_i^n, b_i^n, a_3^n, D^n, W_i, W_3.
2. Calculate the breaking value, $\bar{Z}_i^n$.
3. Order the breaking values from the largest to the smallest; and assign the region of search.
4. Choose a pair of the region of search, and make the first allocation of θ_i^n; 0, $D^n + 1$, $\theta_i^n = \dfrac{-(\overline{ZOP}_i^j + a_i^n - a_s^n)}{2b_i^n}$, D^n and $D^n + 2$.
5. Check the feasibility of the region.
6. Make the second allocation for θ_i^n: $\theta_1^n = D^n + 1$; $\theta_i^n = \dfrac{-(\overline{ZOP}_i^j + a_i^n - a_s^n)}{2b_i^n}$; and $\theta_i^n = D^n + 2$.
7. Calculate total cost of transportation.
8. Compare the total cost with that of the previous minimum cost.
9. Punch out the θ_i^n which gives the current minimum, where $i = 1, 2, 3$ and $n = 1, 2, 3, 4$; the region of the new minimum; the registered minimum; and the current minimum.

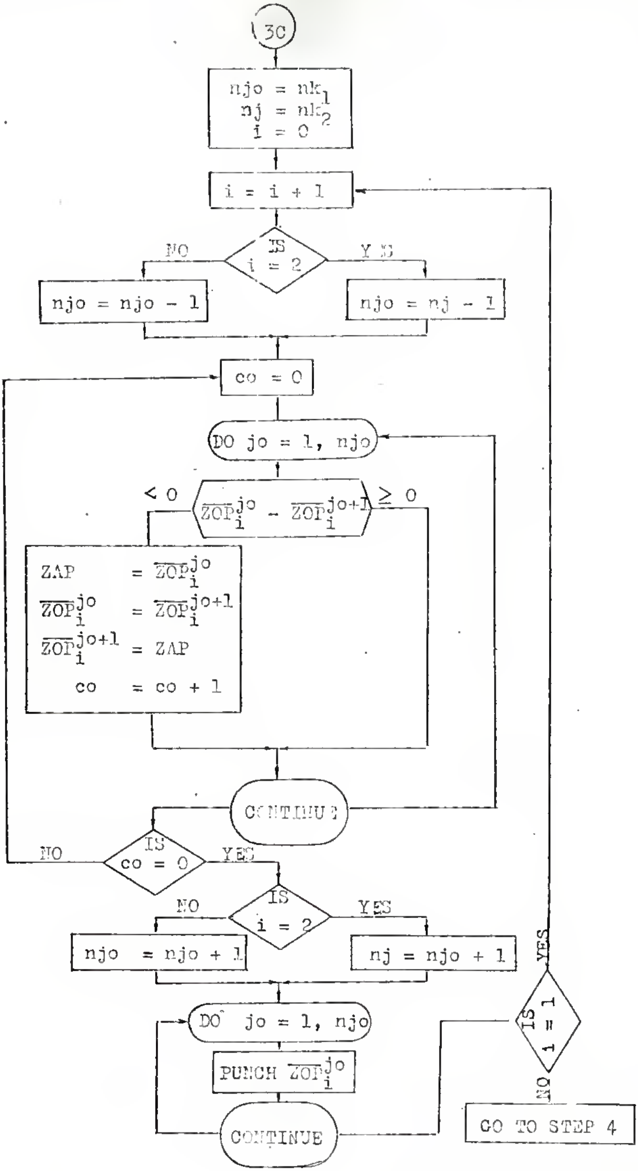
Step 1 & 2


START
INITIALIZATION
SMIN = 1,000,000
s = 3
N = 4
DO i = 1,2
1i
DO n = 1,4
READ $a_i^n, b_i^n, a_3^n, D^n, W_i, W_3$
b_i^n
< 0
> 0
= 0
$\theta_i^n = D^n/2$
UPPER $\bar{Z}_i^n = -a_i^n + a_s^n + b_i^n \theta_i^n$
LOWER Z_i^n = UPPER Z_i^n
UPPER $\bar{Z}_i^n = -a_i^n + a_s^n$
LOWER $Z_i^n = -a_i^n + a_s^n - 2b_i^n D^n$
UPPER $\bar{Z}_i^n = -a_i^n + a_s^n$
LOWER Z_i^n = UPPER Z_i^n
PUNCH i, n, UPPER $\bar{Z}_i^n$, LOWER Z_i^n
CONTINUE
GO TO STEP 3

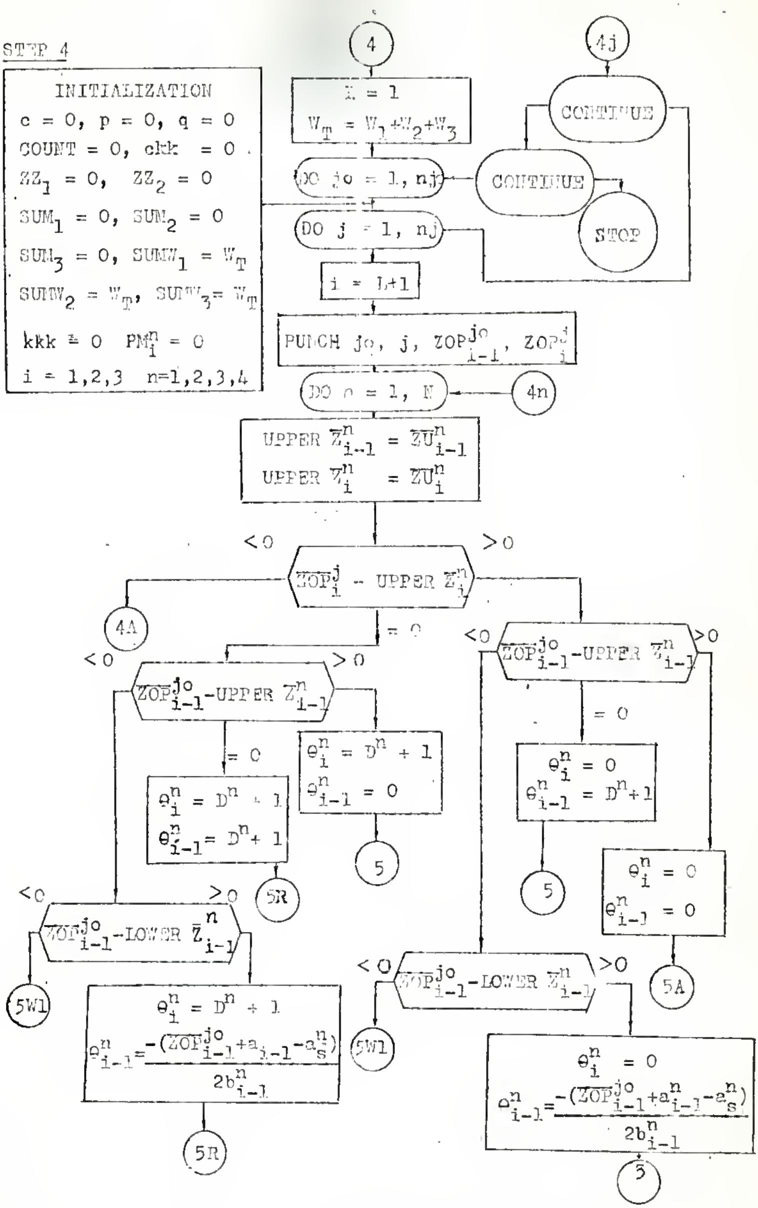
Step 3

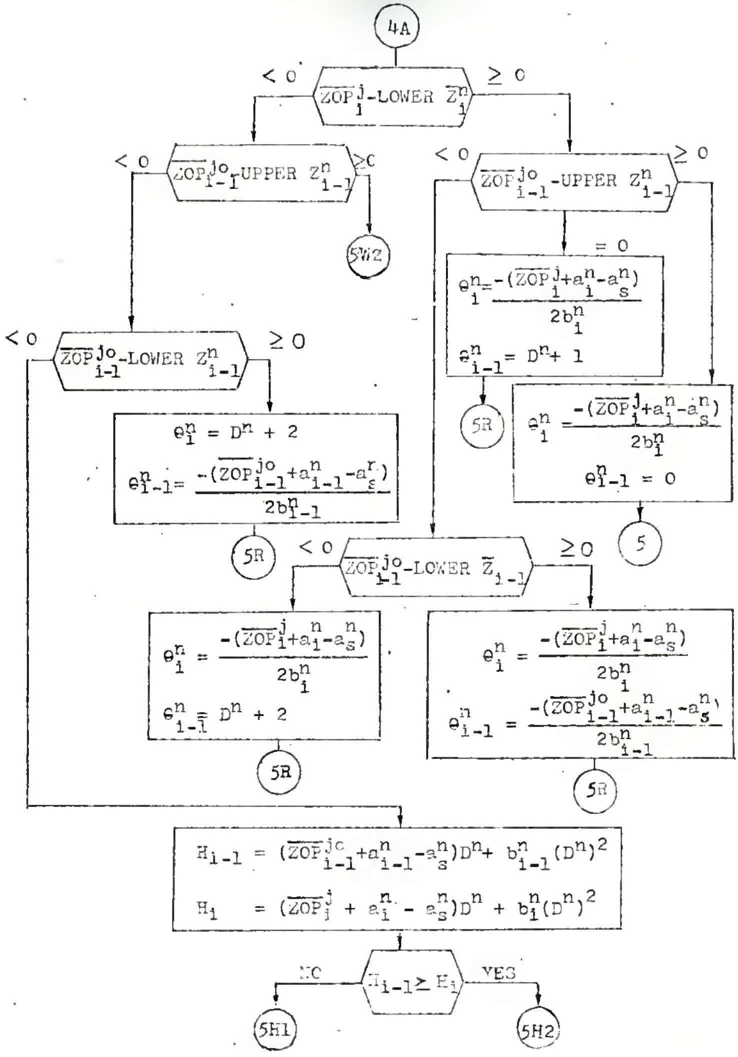

3
DO j = 1,4
$\overline{ZBU}_i^j$ = -1,000
DO n = 1,4
< 0
$\overline{ZBU}_i^j$-Upper Z_i^n
> 0
$\overline{ZBU}_i^j$ = Upper Z_i^n
no = n
CONTINUE
$\overline{ZU}_i^{no}$ = $\overline{ZUP}_i^{no}$
$\overline{ZUP}_i^{no}$ = -1,000
$\overline{bb}_i^j$ = b_i^{no}
$\overline{ZBL}_i^j$ = Lower Z_i^{no}
PUNCH i, j, $\overline{ZBU}_i^j$
CONTINUE
3b

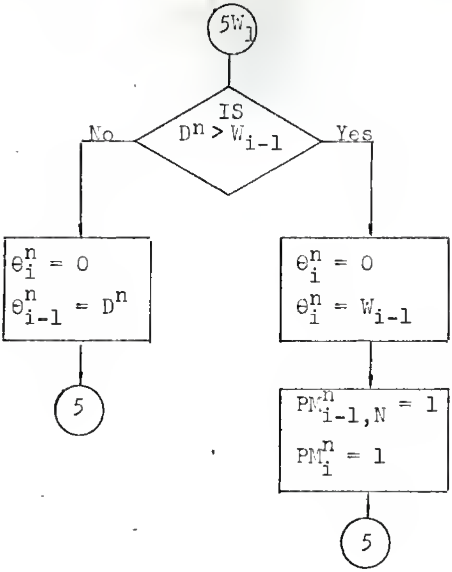

3B
INITIALIZATION
k = 0
DO j=1,4
$\overline{bb}_i^j$
< 0
> 0
= 0
k = k+1
$\overline{ZOP}_i^k = \overline{ZBU}_i^j$
k = k+1
$\overline{ZOP}_i^k = \overline{ZBU}_i^j - .005$
k = k+1
$\overline{ZOP}_i^k = \overline{ZBU}_i^j - .005$
k = k+1
$\overline{ZOP}_i^k = \overline{ZBL}_i^j - .005$
k=k+1
$\overline{ZOP}_i^k = \overline{ZBU}_i^j - .005$
CONTINUE
$nk_i = k$
PUNCH nk_i
CONTINUE
3C
1i

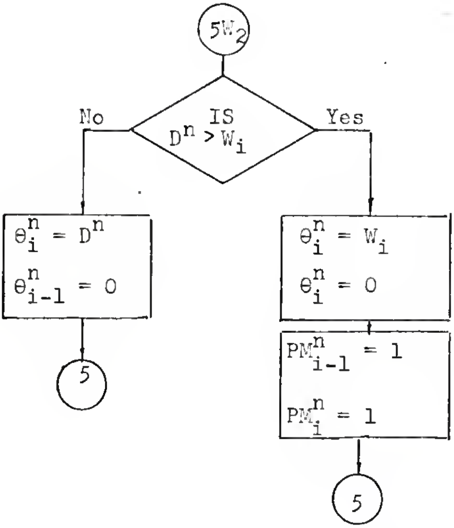

30
njo = nk1
nj = nk2
i = 0
i = i + 1
IS i = 2
NO
YES
njo = njo - 1
njo = nj - 1
co = 0
DO jo = 1, njo
< 0
≥ 0
ZOP i jo - ZOP i jo+1
ZAP = ZOP i jo
ZOP i jo = ZOP i jo+1
ZOP i jo+1 = ZAP
co = co + 1
CONTINUE
IS co = 0
NO
YES
IS i = 2
NO
YES
njo = njo + 1
nj = njo + 1
DO jo = 1, njo
PUNCH ZOP i jo
CONTINUE
IS i = 1
YES
NO
GO TO STEP 4

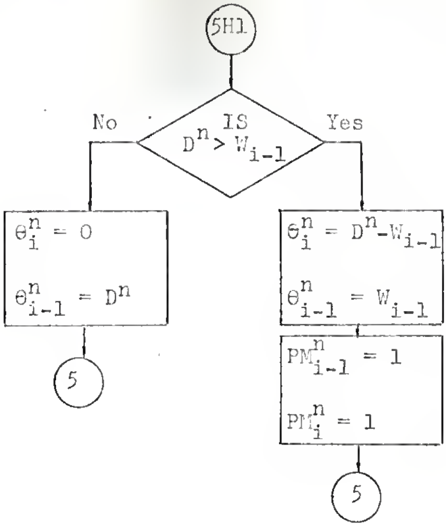
STEP 4

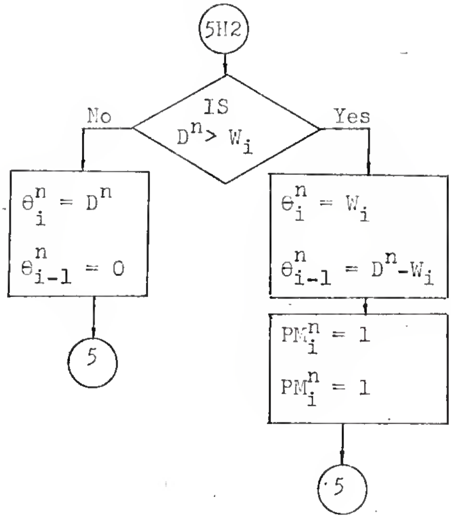

INITIALIZATION
c = 0, p = 0, q = 0
COUNT = 0, ckk = 0
$ZZ_1 = 0$, $ZZ_2 = 0$
$SUM_1 = 0$, $SUM_2 = 0$
$SUM_3 = 0$, $SUMW_1 = W_T$
$SUMW_2 = W_T$, $SUMW_3 = W_T$
kkk = 0 $PM_i^n = 0$
i = 1,2,3 n=1,2,3,4
4
4j
L = 1
$W_T = W_1 + W_2 + W_3$
CONTINUE
DO jo = 1, nj
CONTINUE
DO j = 1, nj
STOP
i = L+1
PUNCH jo, j, ZOP_{i-1}^{jo}, ZOP_i^j
DO n = 1, N
4n
UPPER $Z_{i-1}^n = ZU_{i-1}^n$
UPPER $Z_i^n = ZU_i^n$
< 0
> 0
ZOP_i^j - UPPER Z_i^n
= 0
4A
ZOP_{i-1}^{jo}-UPPER Z_{i-1}^n
$\theta_i^n = D^n + 1$
$\theta_{i-1}^n = 0$
$\theta_i^n = D^n + 1$
$\theta_{i-1}^n = D^n + 1$
5
5R
$\theta_i^n = 0$
$\theta_{i-1}^n = D^n + 1$
$\theta_i^n = 0$
$\theta_{i-1}^n = 0$
5A
ZOP_{i-1}^{jo}-LOWER Z_{i-1}^n
5W1
$\theta_i^n = D^n + 1$
$\theta_{i-1}^n = \dfrac{-(ZOP_{i-1}^{jo} + a_{i-1} - a_s^n)}{2b_{i-1}^n}$
$\theta_i^n = 0$
$\theta_{i-1}^n = \dfrac{-(ZOP_{i-1}^{jo} + a_{i-1}^n - a_s^n)}{2b_{i-1}^n}$


4A
< 0
≥ 0
ZOP^j_i-LOWER Z^n_i
< 0
≥ 0
ZOP$^{jo}_{i-1}$-UPPER Z$^n_{i-1}$
5W2
< 0
≥ 0
ZOP$^{jo}_{i-1}$-UPPER Z$^n_{i-1}$
= 0
$\theta^n_i = \frac{-(\overline{ZOP}^j_i + a^n_i - a^n_s)}{2b^n_i}$
$\theta^n_{i-1} = D^n + 1$
5R
$\theta^n_i = \frac{-(\overline{ZOP}^j_i + a^n_i - a^n_s)}{2b^n_i}$
$\theta^n_{i-1} = 0$
5
< 0
≥ 0
ZOP$^{jo}_{i-1}$-LOWER Z$^n_{i-1}$
$\theta^n_i = D^n + 2$
$\theta^n_{i-1} = \frac{-(\overline{ZOP}^{jo}_{i-1} + a^n_{i-1} - a^n_s)}{2b^n_{i-1}}$
5R
< 0
≥ 0
ZOP$^{jo}_{i-1}$-LOWER Z$_{i-1}$
$\theta^n_i = \frac{-(\overline{ZOP}^j_i + a^n_i - a^n_s)}{2b^n_i}$
$\theta^n_{i-1} = D^n + 2$
5R
$\theta^n_i = \frac{-(\overline{ZOP}^j_i + a^n_i - a^n_s)}{2b^n_i}$
$\theta^n_{i-1} = \frac{-(\overline{ZOP}^{jo}_{i-1} + a^n_{i-1} - a^n_s)}{2b^n_{i-1}}$
5R
$H_{i-1} = (\overline{ZOP}^{jo}_{i-1} + a^n_{i-1} - a^n_s)D^n + b^n_{i-1}(D^n)^2$
$H_i = (\overline{ZOP}^j_i + a^n_i - a^n_s)D^n + b^n_i(D^n)^2$
NO
$H_{i-1} \geq H_i$
YES
5H1
5H2

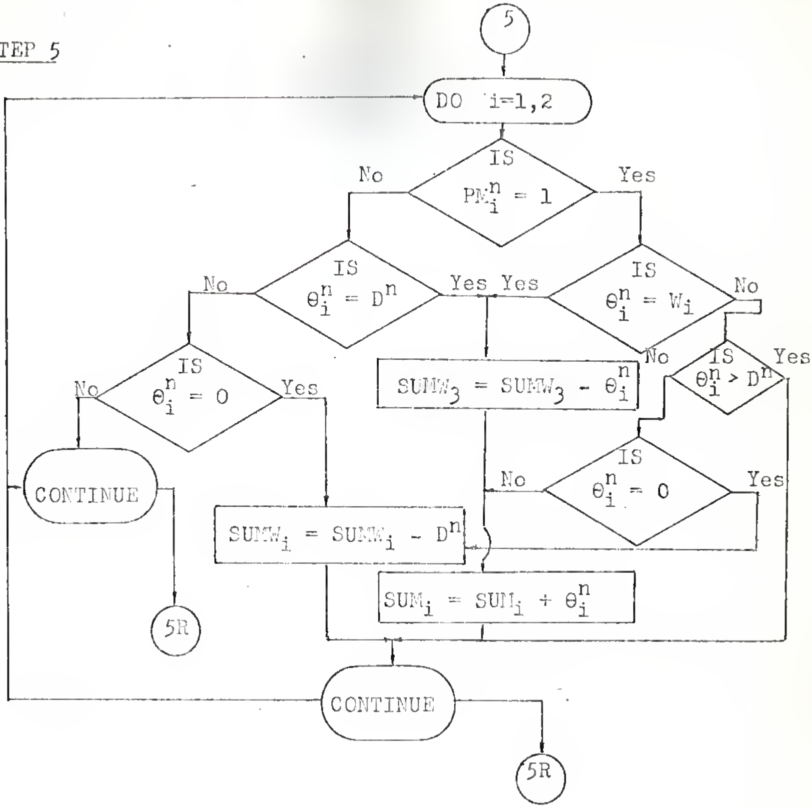

$5W_1$
IS $D^n > W_{i-1}$
No
Yes
$\theta_i^n = 0$
$\theta_{i-1}^n = D^n$
5
$\theta_i^n = 0$
$\theta_i^n = W_{i-1}$
$PM_{i-1,N}^n = 1$
$PM_i^n = 1$
5

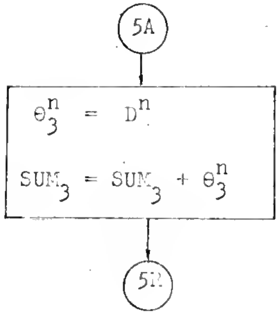

$5W_2$
IS $D^n > W_i$
No
Yes
$\theta_i^n = D^n$
$\theta_{i-1}^n = 0$
5
$\theta_i^n = W_i$
$\theta_i^n = 0$
$PM_{i-1}^n = 1$
$PM_i^n = 1$
5

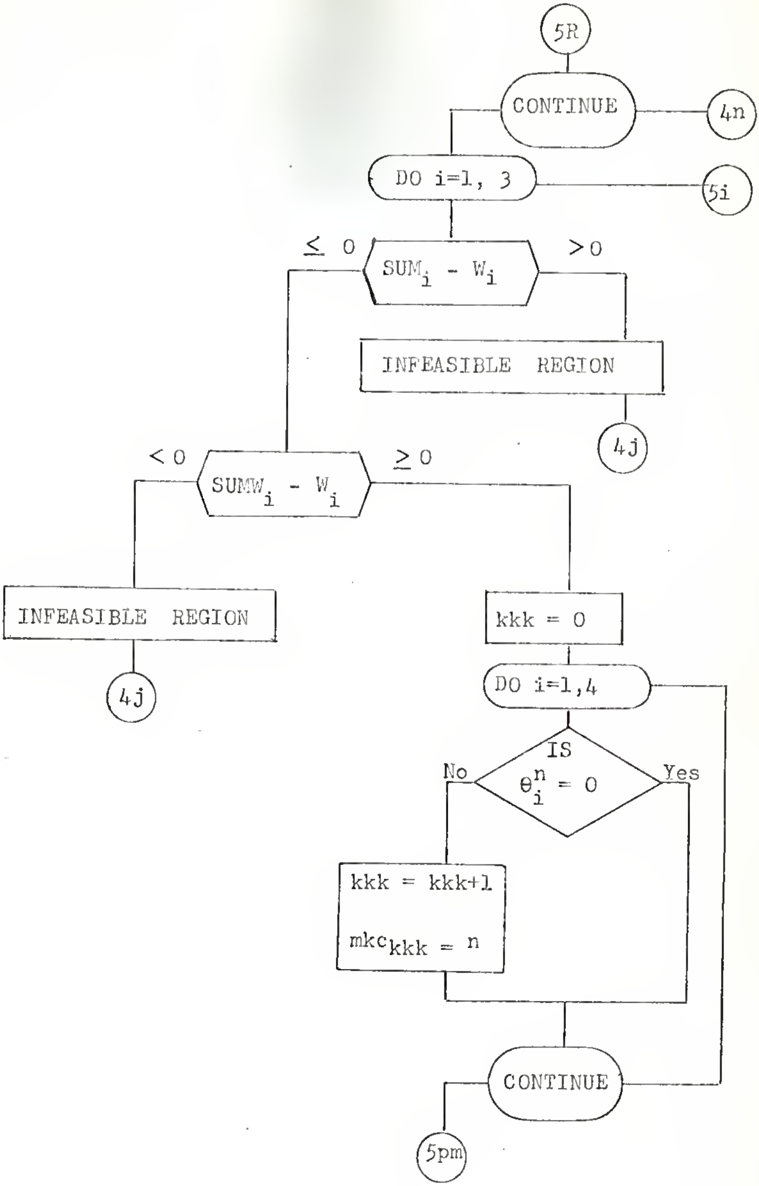

5H1
IS
$D^n > W_{i-1}$
No
Yes
$\theta_i^n = 0$
$\theta_{i-1}^n = D^n$
5
$\theta_i^n = D^n - W_{i-1}$
$\theta_{i-1}^n = W_{i-1}$
$PM_{i-1}^n = 1$
$PM_i^n = 1$
5


5H2
IS
$D^n > W_i$
No
Yes
$\theta_i^n = D^n$
$\theta_{i-1}^n = 0$
5
$\theta_i^n = W_i$
$\theta_{i-1}^n = D^n - W_i$
$PM_i^n = 1$
$PM_i^n = 1$
5

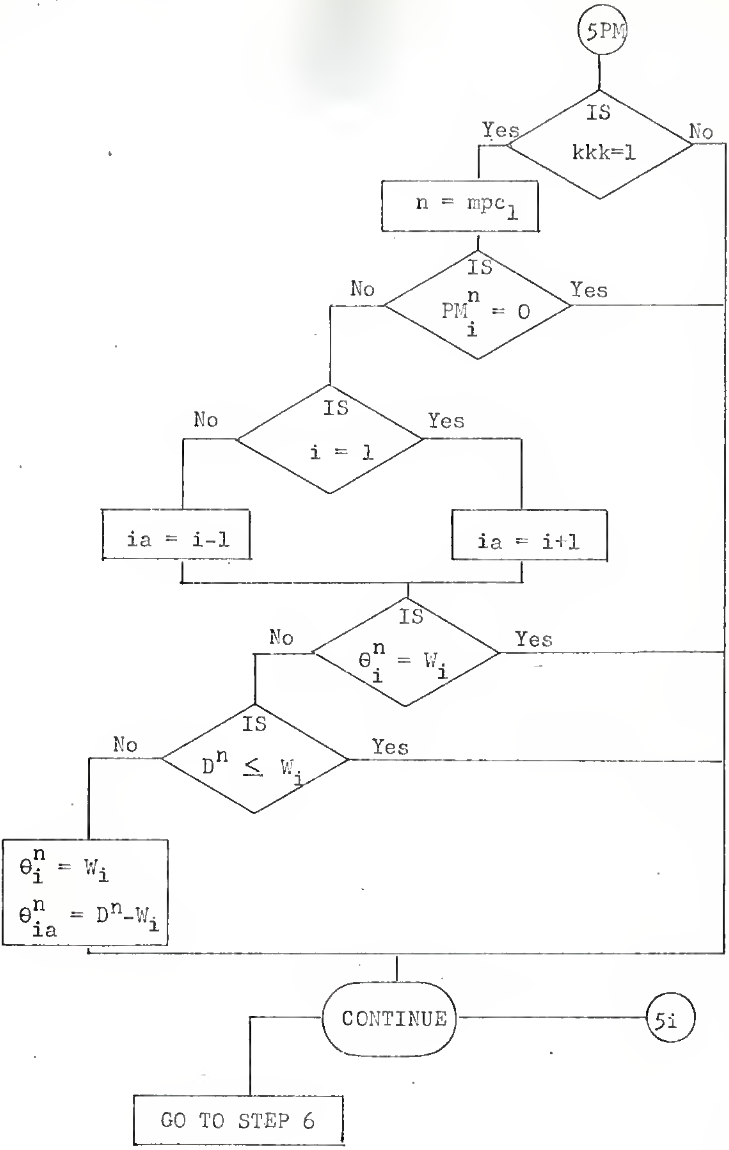
STEP 5


5
DO i=1,2
IS
$PM_i^n = 1$
No
Yes
IS
$\theta_i^n = D^n$
No
Yes
IS
$\theta_i^n = W_i$
Yes
No
No
IS
$\theta_i^n > D^n$
Yes
IS
$\theta_i^n = 0$
No
Yes
$SUMW_3 = SUMW_3 - \theta_i^n$
CONTINUE
IS
$\theta_i^n = 0$
No
Yes
$SUMW_i = SUMW_i - D^n$
$SUM_i = SUM_i + \theta_i^n$
5R
CONTINUE
5R

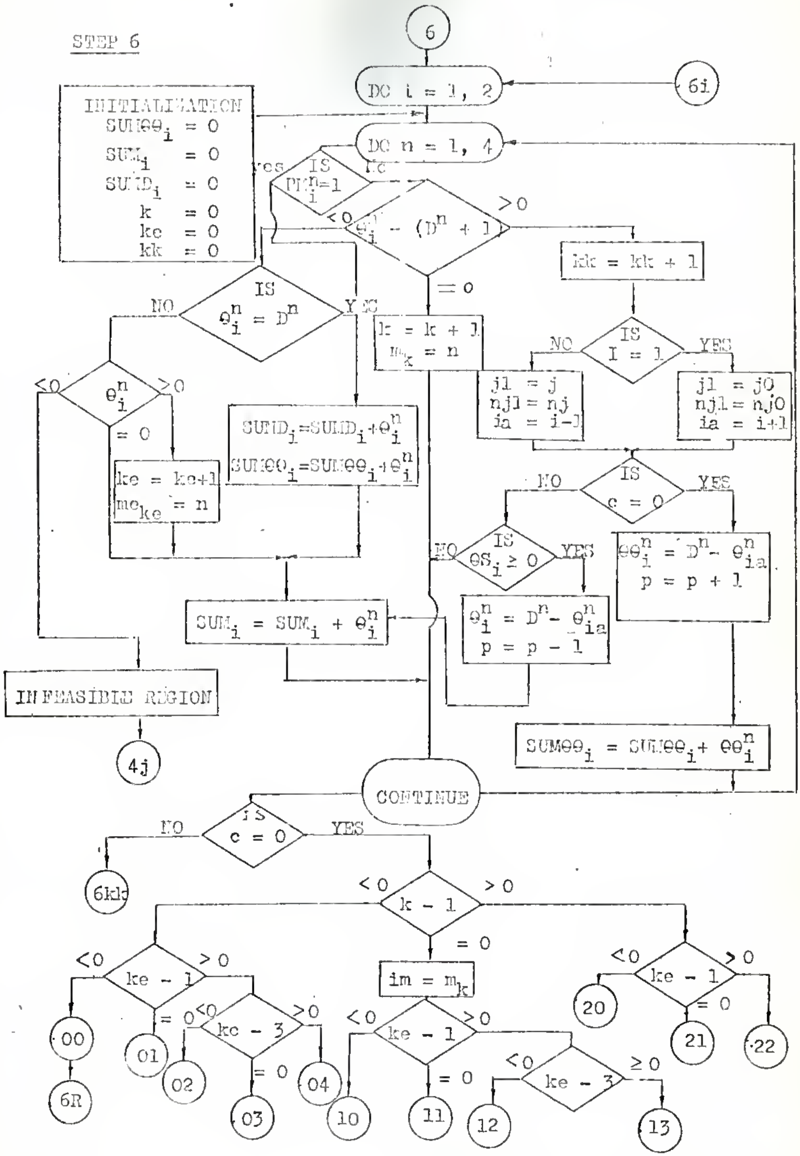

5A
$\theta_3^n = D^n$
$SUM_3 = SUM_3 + \theta_3^n$
5R



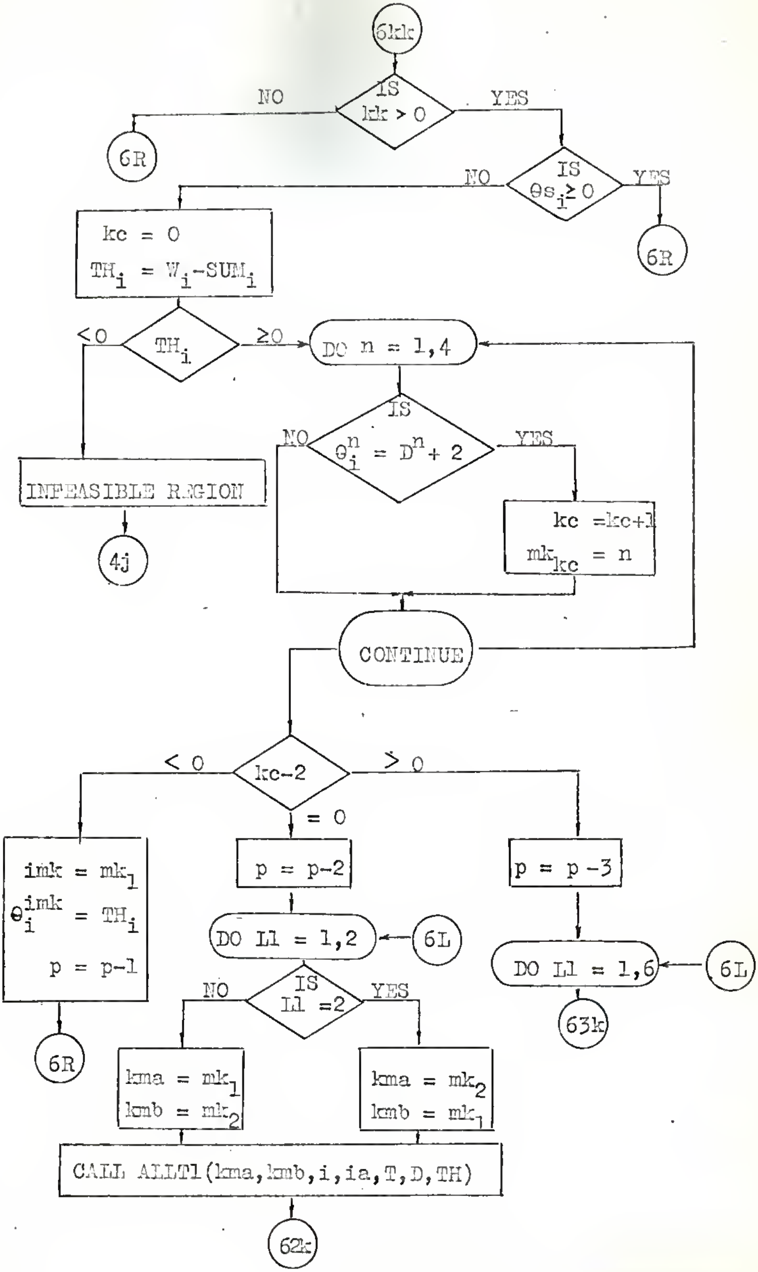
5R
CONTINUE
4n
DO i=1, 3
5i
≤ 0
$SUM_i - W_i$
> 0
INFEASIBLE REGION
4j
< 0
$SUMW_i - W_i$
≥ 0
INFEASIBLE REGION
4j
kkk = 0
DO i=1,4
IS $\theta_i^n = 0$
No
Yes
kkk = kkk+1
$mkc_{kkk} = n$
CONTINUE
5pm

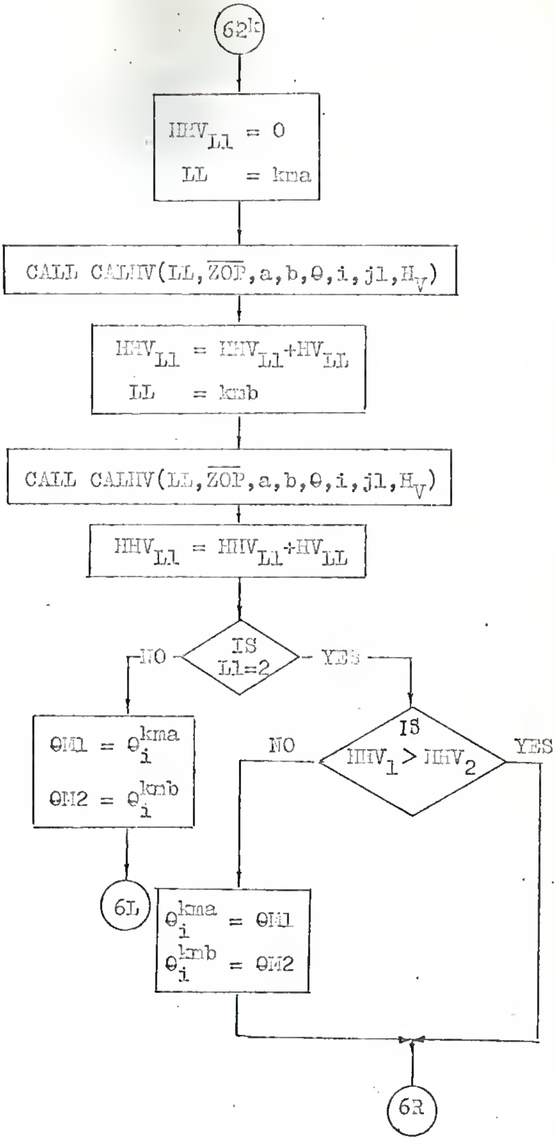

5PM
IS kkk=1
Yes
No
n = mpc1
IS PM n i = 0
No
Yes
IS i = 1
No
Yes
ia = i-1
ia = i+1
IS θ n i = W i
No
Yes
IS D n ≤ W i
No
Yes
θ n i = W i
θ n ia = D n - W i
CONTINUE
5i
GO TO STEP 6

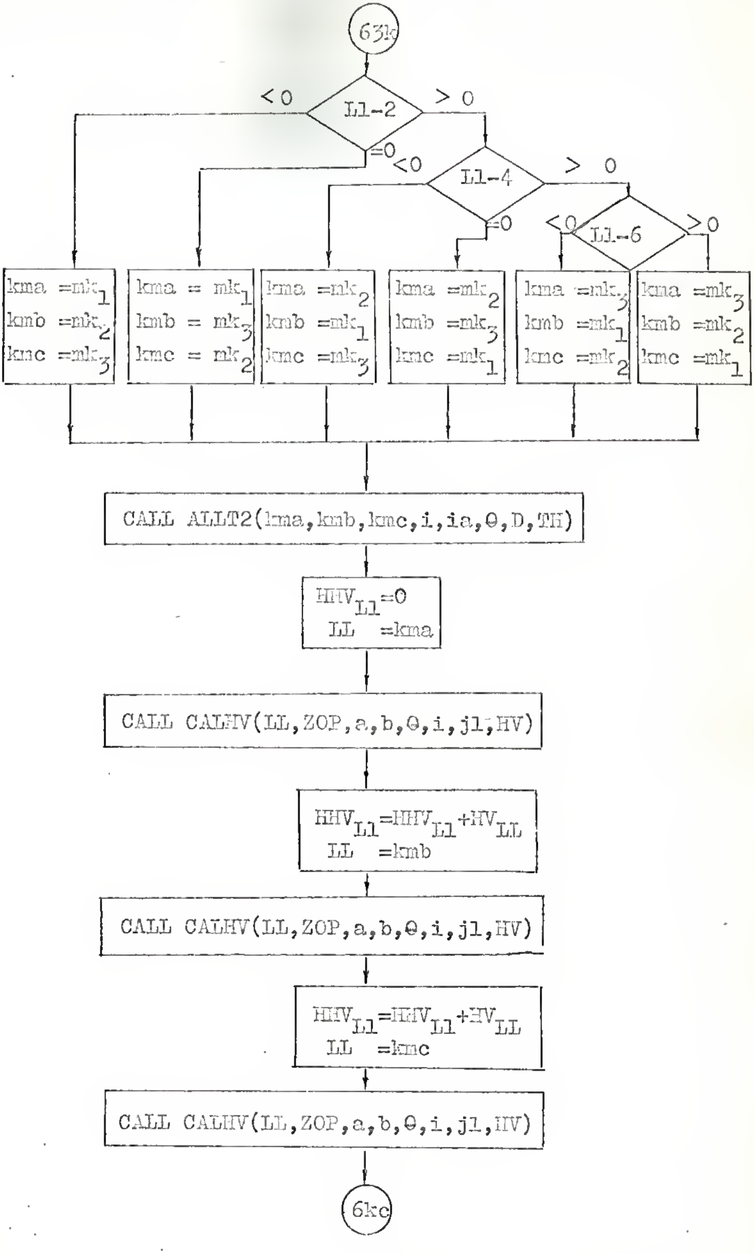
STEP 6

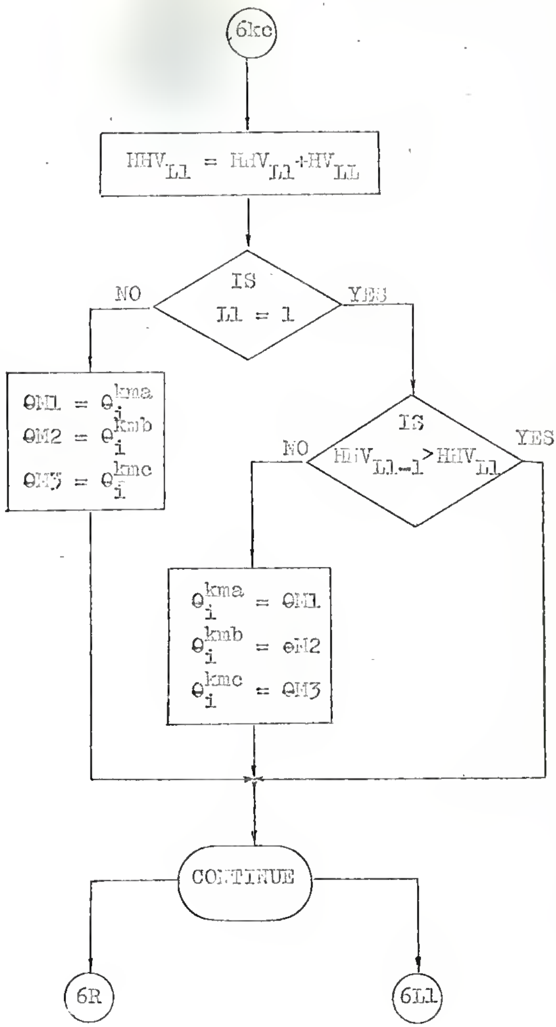


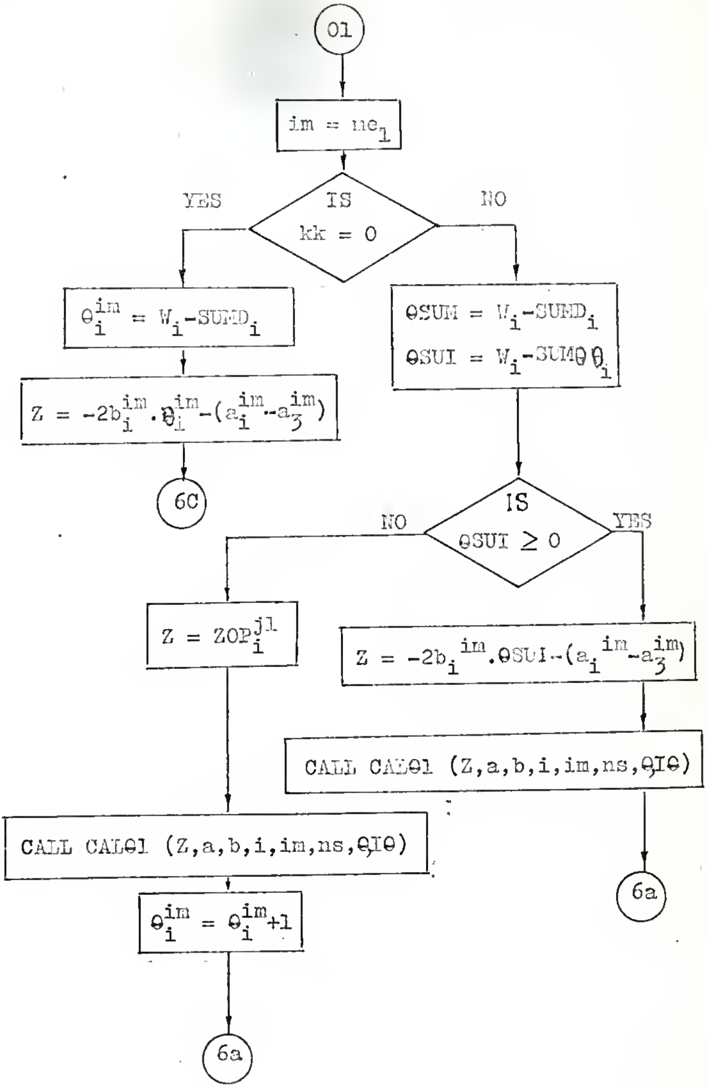
6
DO i = 1, 2
6i
INITIALIZATION
SUMθθ_i = 0
SUM_i = 0
SUMD_i = 0
k = 0
ke = 0
kk = 0
DO n = 1, 4
IS PK^n_i = 1
yes
NO
θ^n_i − (D^n + 1)
< 0
> 0
= 0
kk = kk + 1
IS θ^n_i = D^n
NO
YES
k = k + 1
m_k = n
IS I = 1
NO
YES
j1 = j
nj1 = nj
ia = i−1
j1 = j0
nj1 = nj0
ia = i+1
θ^n_i
< 0
> 0
= 0
SUMD_i = SUMD_i + θ^n_i
SUMθθ_i = SUMθθ_i + θ^n_i
ke = ke+1
me_ke = n
IS c = 0
NO
YES
IS θS_i ≥ 0
NO
YES
θθ^n_i = D^n − θ^n_ia
p = p + 1
SUM_i = SUM_i + θ^n_i
θ^n_i = D^n − θ^n_ia
p = p − 1
INFEASIBLE REGION
4j
SUMθθ_i = SUMθθ_i + θθ^n_i
CONTINUE
IS c = 0
NO
YES
6kk
k − 1
< 0
> 0
= 0
ke − 1
< 0
> 0
= 0
ke − 3
< 0
> 0
= 0
im = m_k
ke − 1
< 0
> 0
= 0
ke − 3
< 0
≥ 0
ke − 1
< 0
> 0
= 0
00
01
02
03
04
6R
10
11
12
13
20
21
22

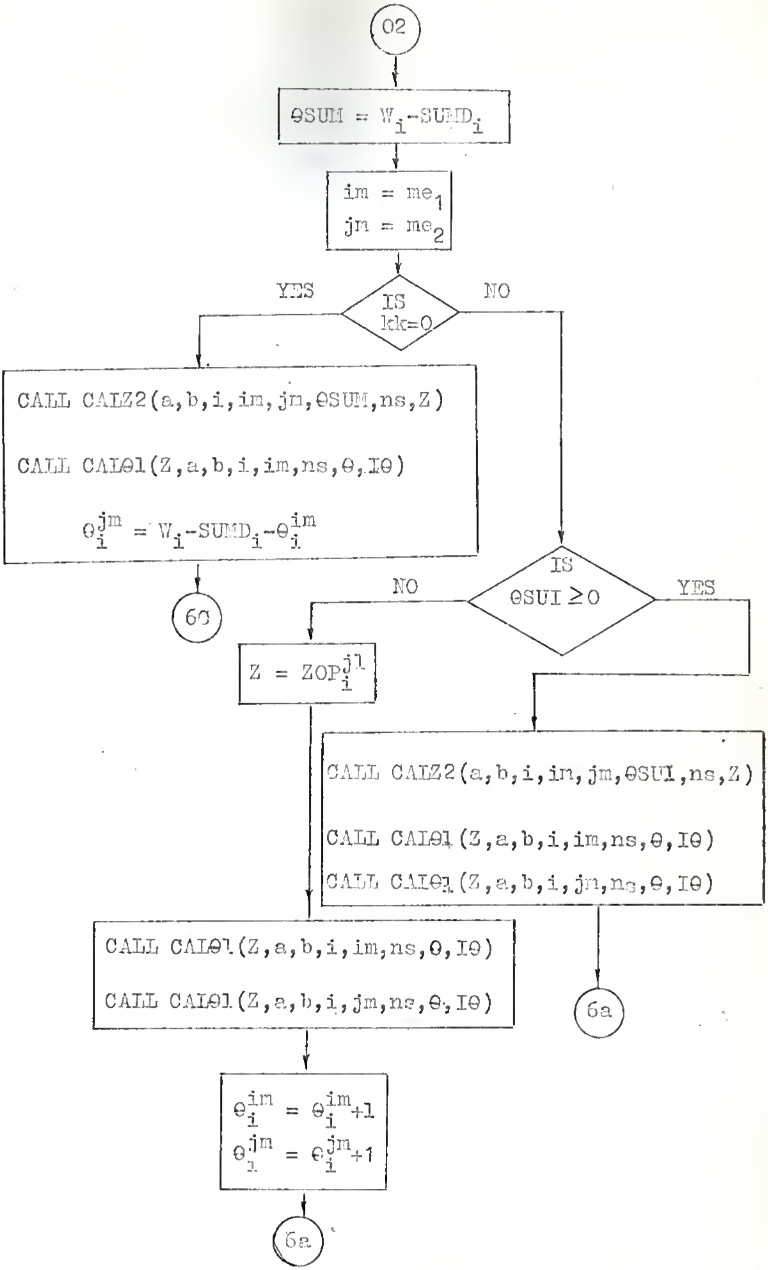

6kk
IS
kk > 0
NO
YES
6R
IS
θs_i ≥ 0
NO
YES
6R
kc = 0
TH_i = W_i-SUM_i
TH_i
<0
≥0
DO n = 1,4
IS
θ_i^n = D^n + 2
NO
YES
kc = kc+1
mk_kc = n
INFEASIBLE REGION
4j
CONTINUE
kc-2
< 0
= 0
> 0
imk = mk_1
θ_i^imk = TH_i
p = p-1
6R
p = p-2
DO L1 = 1,2
6L
IS
L1 = 2
NO
YES
kma = mk_1
kmb = mk_2
kma = mk_2
kmb = mk_1
CALL ALLT1(kma,kmb,i,ia,T,D,TH)
62k
p = p -3
DO L1 = 1,6
6L
63k

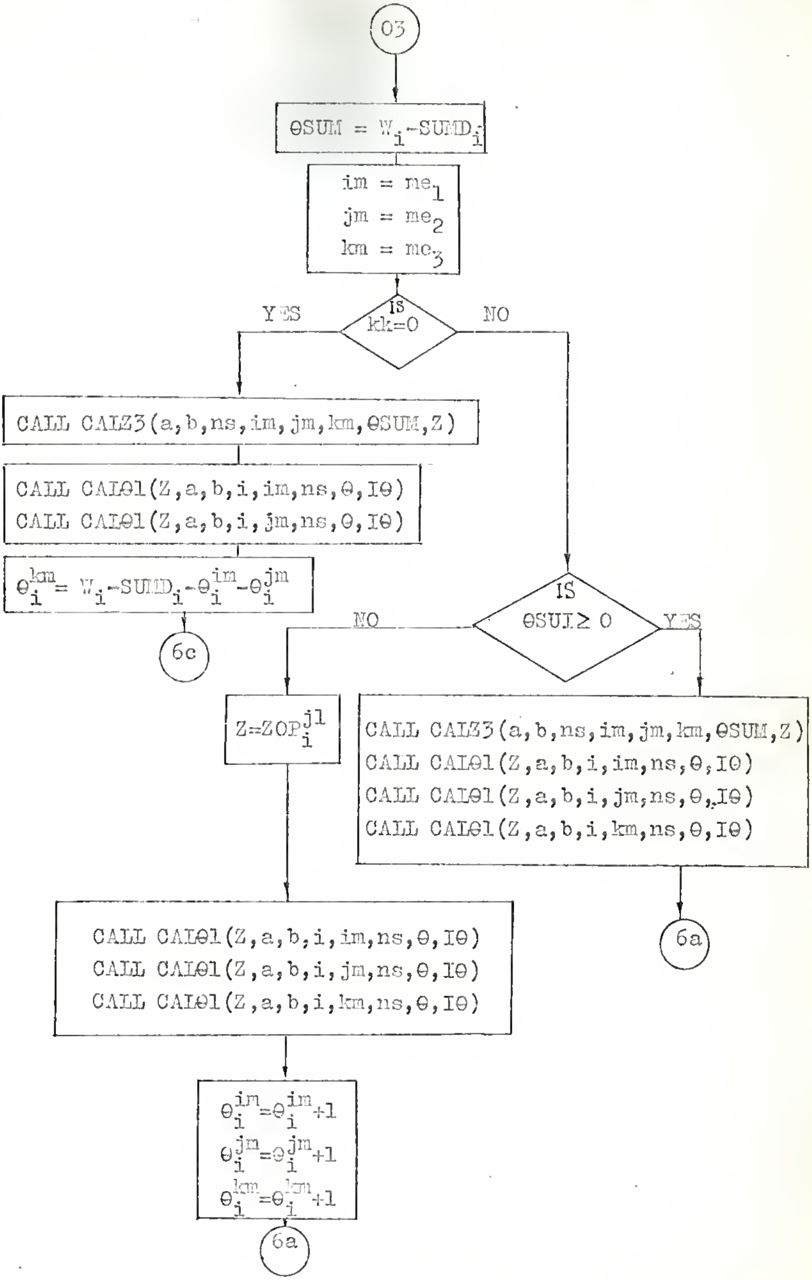

62k
HHV_L1 = 0
LL = kma
CALL CALHV(LL,ZOP,a,b,θ,i,j1,H_V)
HHV_L1 = HHV_L1+HV_LL
LL = kmb
CALL CALHV(LL,ZOP,a,b,θ,i,j1,H_V)
HHV_L1 = HHV_L1+HV_LL
IS L1=2
NO
YES
θM1 = θ_i^kma
θM2 = θ_i^kmb
6L
IS HHV_1 > HHV_2
NO
YES
θ_i^kma = θM1
θ_i^kmb = θM2
6R


63k
L1-2
< 0
> 0
=0
L1-4
<0
> 0
=0
L1-6
<0
>0
kma =mk1
kmb =mk2
kmc =mk3
kma = mk1
kmb = mk3
kmc = mk2
kma =mk2
kmb =mk1
kmc =mk3
kma =mk2
kmb =mk3
kmc =mk1
kma =mk3
kmb =mk1
kmc =mk2
kma =mk3
kmb =mk2
kmc =mk1
CALL ALLT2(kma,kmb,kmc,i,ia,Θ,D,TH)
HHVL1=0
LL =kma
CALL CALHV(LL,ZOP,a,b,Θ,i,j1,HV)
HHVL1=HHVL1+HVLL
LL =kmb
CALL CALHV(LL,ZOP,a,b,Θ,i,j1,HV)
HHVL1=HHVL1+HVLL
LL =kmc
CALL CALHV(LL,ZOP,a,b,Θ,i,j1,HV)
6kc

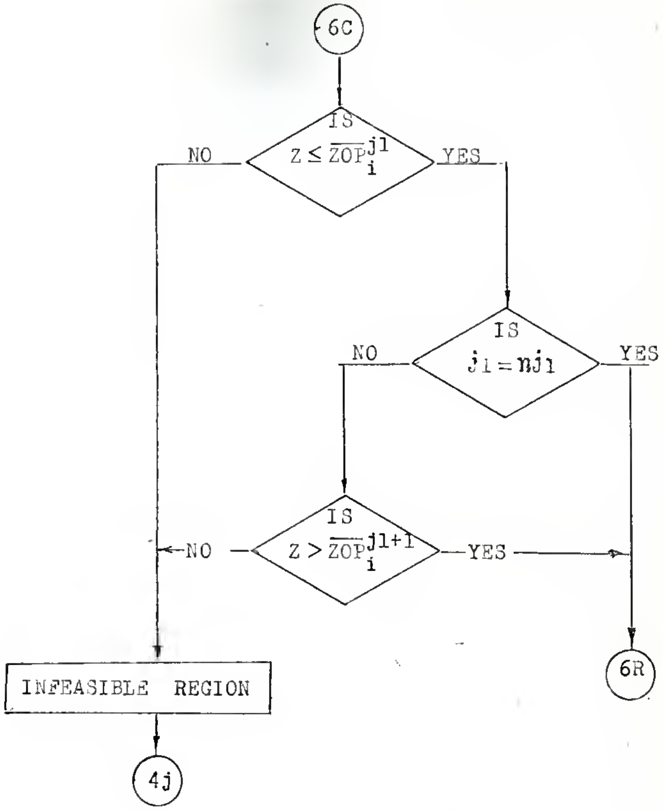

6kc
HHV_L1 = HHV_L1 + HV_LL
IS L1 = 1
NO
YES
ΘM1 = Θ_i^kma
ΘM2 = Θ_i^kmb
ΘM3 = Θ_i^kmc
IS HHV_L1-1 > HHV_L1
NO
YES
Θ_i^kma = ΘM1
Θ_i^kmb = ΘM2
Θ_i^kmc = ΘM3
CONTINUE
6R
6L1


01
im = ne1
IS
kk = 0
YES
NO
θi im = Wi-SUMDi
θSUM = Wi-SUMDi
θSUI = Wi-SUMθ θi
Z = -2bi im . θi im -(ai im - a3 im)
6C
IS
θSUI ≥ 0
NO
YES
Z = ZOPi j1
Z = -2bi im .θSUI -(ai im -a3 im)
CALL CALθ1 (Z,a,b,i,im,ns,θIθ)
6a
CALL CALθ1 (Z,a,b,i,im,ns,θIθ)
θi im = θi im +1
6a


02
θSUM = W_i-SUMD_i
im = me_1
jm = me_2
IS kk=0
YES
NO
CALL CALZ2(a,b,i,im,jm,θSUM,ns,Z)
CALL CALθ1(Z,a,b,i,im,ns,θ,Iθ)
θ_i^jm = W_i-SUMD_i-θ_i^im
6C
IS θSUI ≥ 0
NO
YES
Z = ZOP_i^jl
CALL CALZ2(a,b,i,im,jm,θSUI,ns,Z)
CALL CALθ1(Z,a,b,i,im,ns,θ,Iθ)
CALL CALθ1(Z,a,b,i,jm,ns,θ,Iθ)
CALL CALθ1(Z,a,b,i,im,ns,θ,Iθ)
CALL CALθ1(Z,a,b,i,jm,ns,θ,Iθ)
6a
θ_i^im = θ_i^im+1
θ_i^jm = θ_i^jm+1
6a



03
ΘSUM = W_i - SUMD_i
im = me_1
jm = me_2
km = me_3
IS kk=0
YES
NO
CALL CALZ3(a,b,ns,im,jm,km,ΘSUM,Z)
CALL CALΘ1(Z,a,b,i,im,ns,Θ,IΘ)
CALL CALΘ1(Z,a,b,i,jm,ns,Θ,IΘ)
Θ_i^km = W_i - SUMD_i - Θ_i^im - Θ_i^jm
6c
IS ΘSUM ≥ 0
NO
YES
Z=ZOP_i^jl
CALL CALZ3(a,b,ns,im,jm,km,ΘSUM,Z)
CALL CALΘ1(Z,a,b,i,im,ns,Θ,IΘ)
CALL CALΘ1(Z,a,b,i,jm,ns,Θ,IΘ)
CALL CALΘ1(Z,a,b,i,km,ns,Θ,IΘ)
6a
CALL CALΘ1(Z,a,b,i,im,ns,Θ,IΘ)
CALL CALΘ1(Z,a,b,i,jm,ns,Θ,IΘ)
CALL CALΘ1(Z,a,b,i,km,ns,Θ,IΘ)
Θ_i^im = Θ_i^im + 1
Θ_i^jm = Θ_i^jm + 1
Θ_i^km = Θ_i^km + 1
6a


6C
IS
$Z \leq \overline{ZOP}_i^{j1}$
NO
YES
IS
$j1 = nj1$
NO
YES
IS
$Z > \overline{ZOP}_i^{j1+1}$
NO
YES
INFEASIBLE REGION
4j
6R

(04)

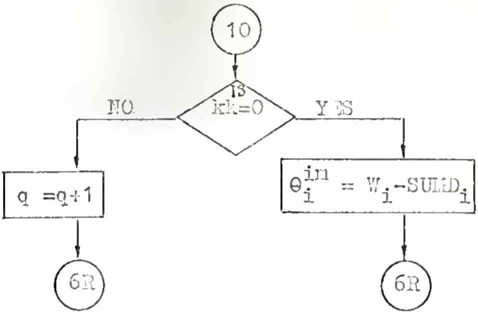
$$im = me_1$$
$$jm = me_2$$
$$km = me_3$$
$$lm = me_4$$

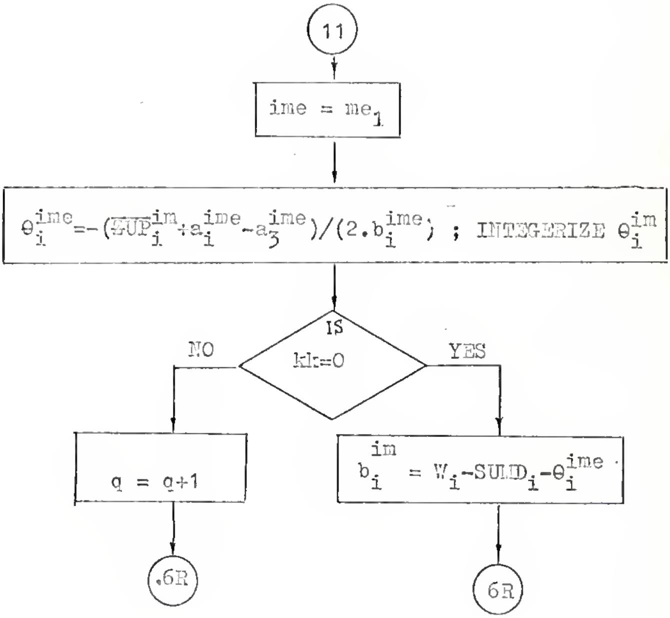
$$Z1 = 2.b_i^{im}.b_i^{jm}.b_i^{km}.b_i^{lm}.W_i$$
$$Z2 = b_i^{jm}.b_i^{km}.b_i^{lm}.(a_i^{im}-a_3^{im})$$
$$Z3 = b_i^{im}.b_i^{km}.b_i^{lm}.(a_i^{jm}-a_3^{jm})$$
$$Z4 = b_i^{im}.b_i^{jm}.b_i^{lm}.(a_i^{km}-a_3^{km})$$
$$Z5 = b_i^{im}.b_i^{jm}.b_i^{km}.(a_i^{lm}-a_3^{lm})$$
$$Z6 = b_i^{jm}.b_i^{km}.b_i^{lm}+b_i^{im}.b_i^{km}.b_i^{lm}$$
$$Z7 = b_i^{im}.b_i^{jm}.b_i^{lm}+b_i^{im}.b_i^{jm}.b_i^{km}$$

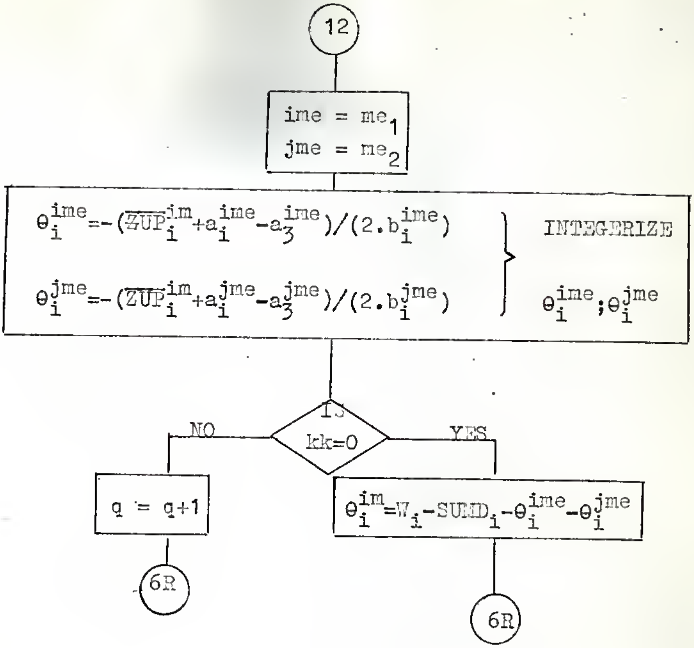
$$Z = (Z1+Z2+Z3+Z4+Z5) \;/\; (Z6+Z7)$$

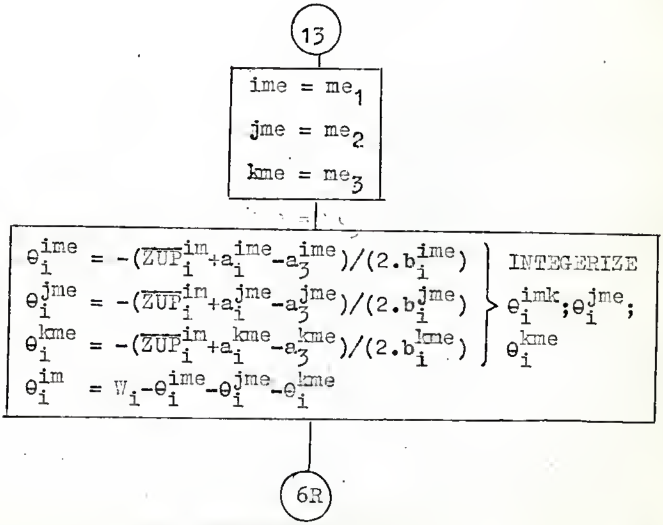
$$\left.\begin{aligned} \theta_i^{im} &= (Z+a_i^{im}-a_3^{im}) \;/\; (2.b_i^{im}) \\ \theta_i^{jm} &= (Z+a_i^{jm}-a_3^{jm}) \;/\; (2.b_i^{jm}) \\ \theta_i^{km} &= (Z+a_i^{km}-a_3^{km}) \;/\; (2.b_i^{km}) \end{aligned}\right\} \text{INTEGERIZE } \theta_i^{im}, \theta_i^{jm}, \theta_i^{km}$$
$$\theta_i^{lm} = W_i-\theta_i^{im}-\theta_i^{jm}-\theta_i^{km}$$

(6R)


10
IS
kk=0
NO
YES
q =q+1
$\theta_i^{im} = W_i - SUMD_i$
6R
6R


11
ime = me_1
$\theta_i^{ime} = -(\overline{SUP}_i^{im} + a_i^{ime} - a_3^{ime})/(2.b_i^{ime})$; INTEGERIZE θ_i^{im}
IS
kk=0
NO
YES
q = q+1
$b_i^{im} = W_i - SUMD_i - \theta_i^{ime}$
6R
6R


12
ime = me$_1$
jme = me$_2$
$\theta_i^{ime}=-(\overline{ZUP}_i^{im}+a_i^{ime}-a_3^{ime})/(2.b_i^{ime})$
$\theta_i^{jme}=-(\overline{ZUP}_i^{im}+a_i^{jme}-a_3^{jme})/(2.b_i^{jme})$
INTEGERIZE
$\theta_i^{ime};\theta_i^{jme}$
IS
kk=0
NO
YES
q = q+1
$\theta_i^{im}=W_i-SUMD_i-\theta_i^{ime}-\theta_i^{jme}$
6R
6R


13
ime = me$_1$
jme = me$_2$
kme = me$_3$
$\theta_i^{ime} = -(\overline{ZUP}_i^{im}+a_i^{ime}-a_3^{ime})/(2.b_i^{ime})$
$\theta_i^{jme} = -(\overline{ZUP}_i^{im}+a_i^{jme}-a_3^{jme})/(2.b_i^{jme})$
$\theta_i^{kme} = -(\overline{ZUP}_i^{im}+a_i^{kme}-a_3^{kme})/(2.b_i^{kme})$
$\theta_i^{im} = W_i-\theta_i^{ime}-\theta_i^{jme}-\theta_i^{kme}$
INTEGERIZE
$\theta_i^{imk};\theta_i^{jme};$
θ_i^{kme}
6R

(20)

$q = q+1$

(6R)

(21)

$ime = me_{ke}$

$\theta_i^{ime} = -(\overline{ZUP}_i^{im} + a_i^{ime} - a_3^{ime})/(2.b_i^{ime})$; INTEGERIZE θ_i^{ime}

$q = q+1$

(6R)

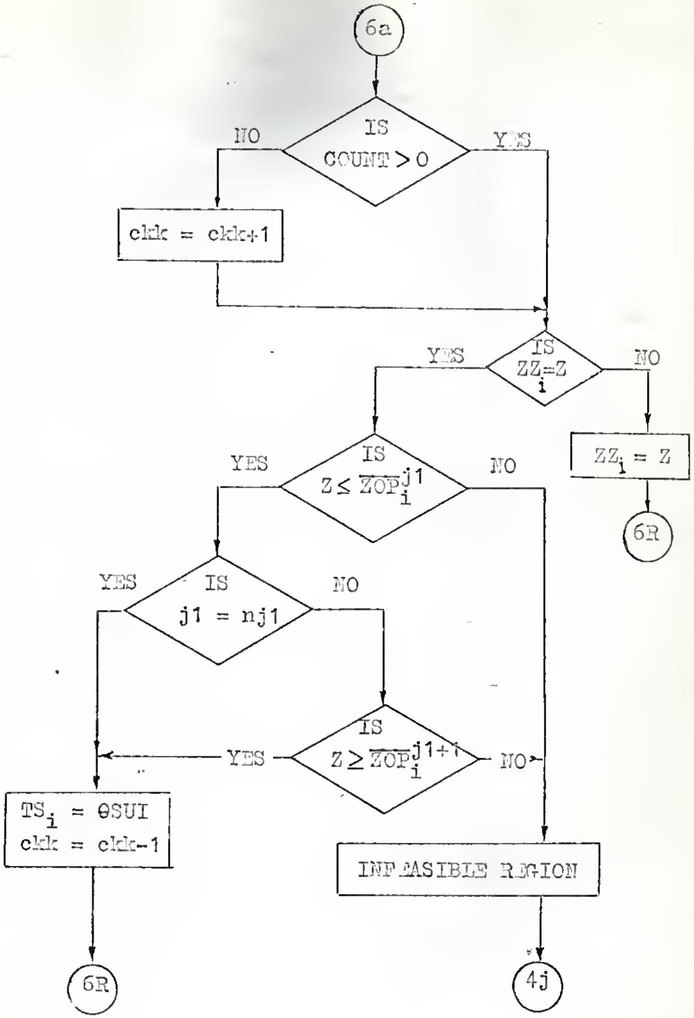
(22)

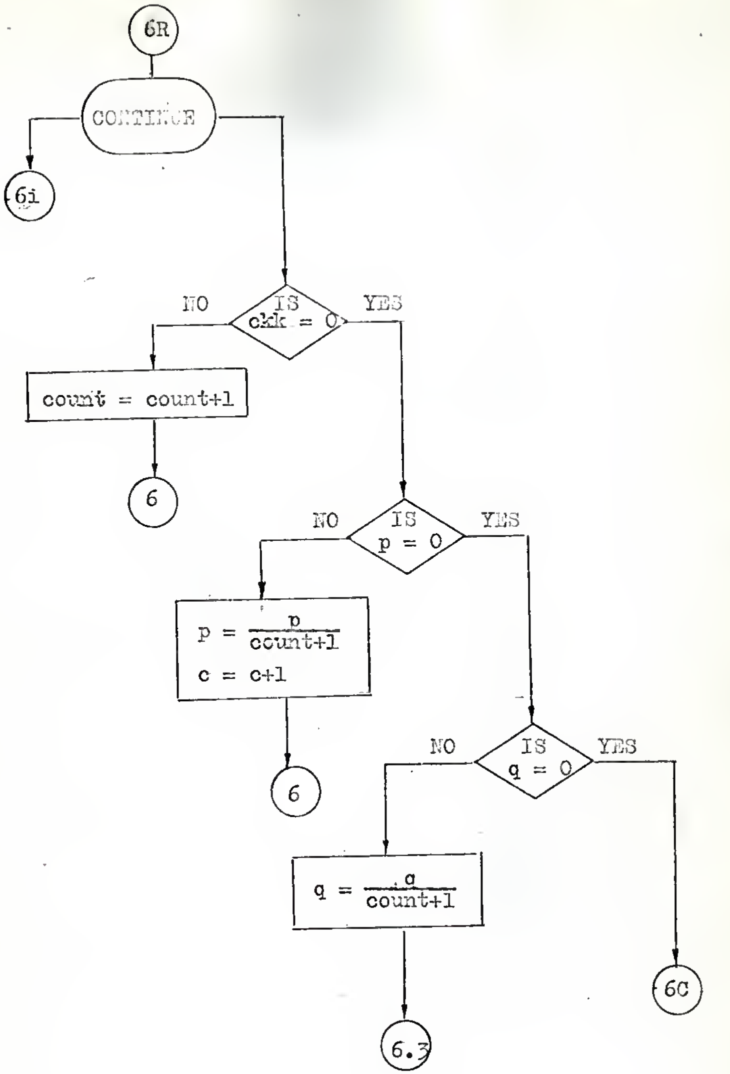
$ime = me_1$
$jme = me_2$

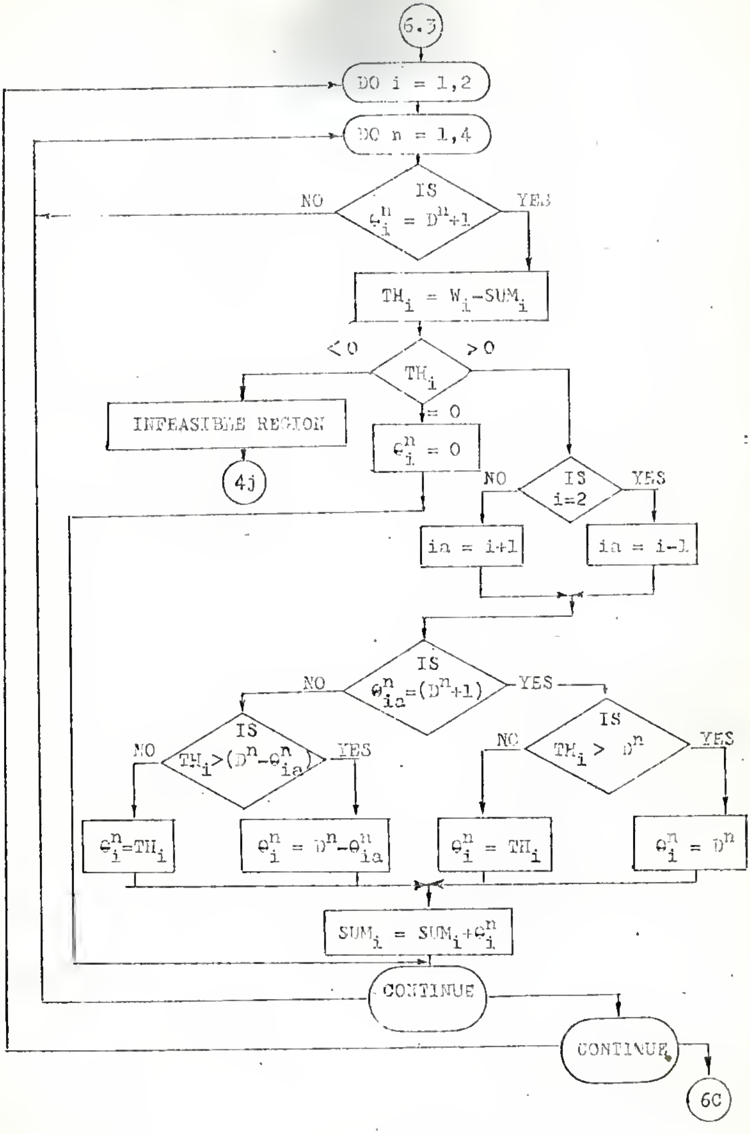
$$\left.\begin{array}{l} \theta_i^{ime} = -(\overline{ZUP}_i^{im} + a_i^{ime} - a_3^{ime})/(2.b_i^{ime}) \\ \theta_i^{jme} = -(\overline{ZUP}_i^{im} + a_i^{jme} - a_3^{jme})/(2.b_i^{jme}) \end{array}\right\} \begin{array}{l} \text{INTEGERIZE} \\ \theta_i^{ime};\theta_i^{jme} \end{array}$$

$q = q+1$

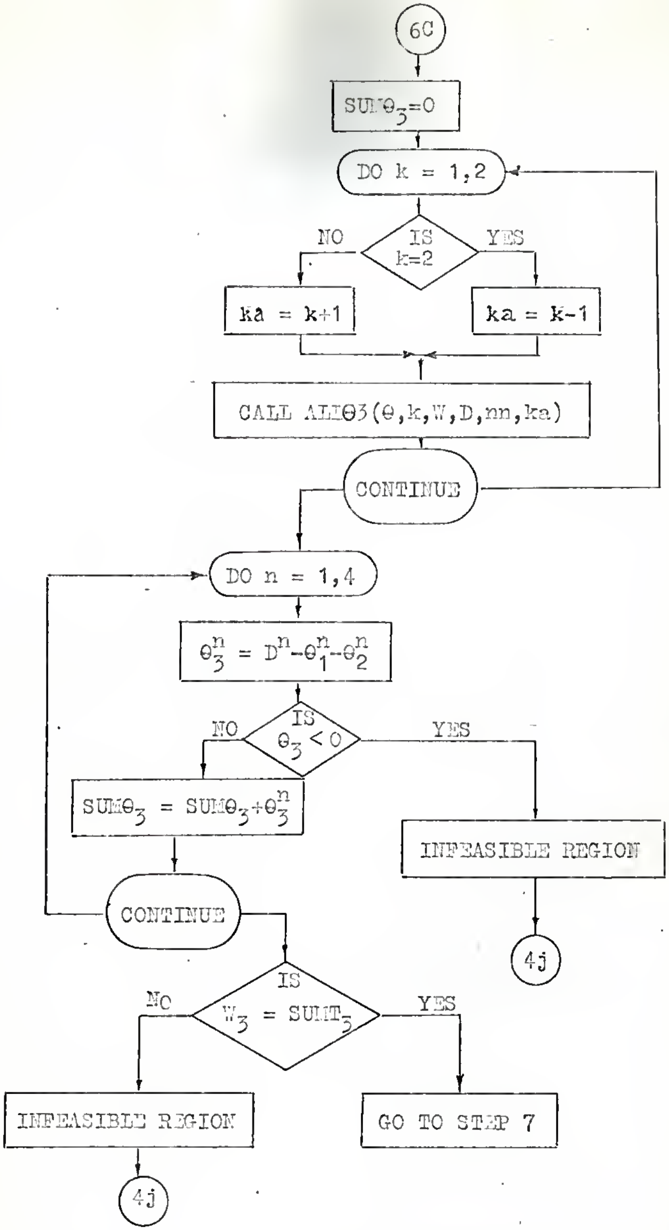
(6R)


6a
IS
COUNT > 0
NO
YES
ckk = ckk+1
IS
ZZ = Z
i
YES
NO
ZZ
i
= Z
6R
IS
Z ≤ ZOP
j1
i
YES
NO
IS
j1 = nj1
YES
NO
IS
Z ≥ ZOP
j1+1
i
YES
NO
TS
i
= θSUI
ckk = ckk-1
INFEASIBLE REGION
6R
4j

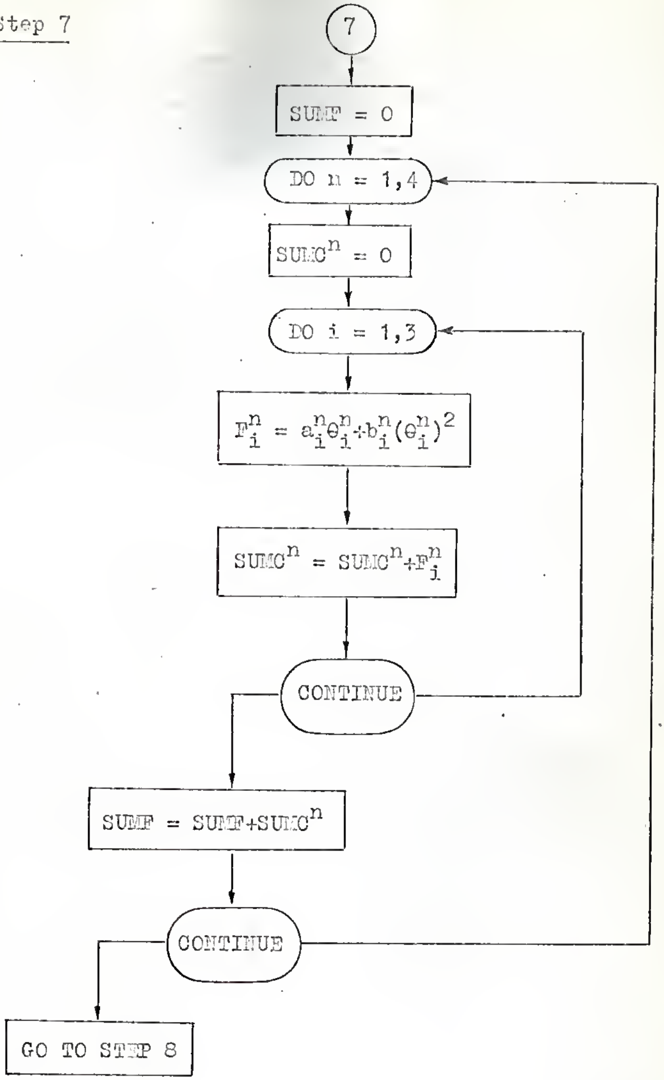

6R
CONTINUE
6i
NO
IS ckk = 0
YES
count = count+1
6
NO
IS p = 0
YES
p = p/(count+1)
c = c+1
6
NO
IS q = 0
YES
q = q/(count+1)
6.3
6C



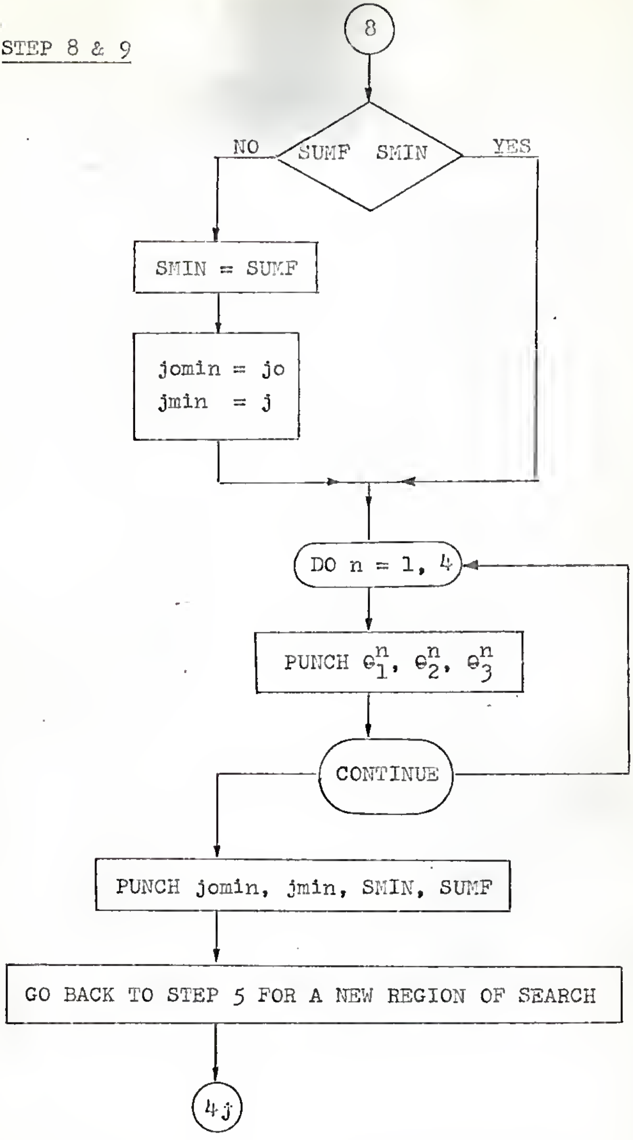
6.3
DO i = 1,2
DO n = 1,4
NO
IS $\theta_i^n = D^n+1$
YES
$TH_i = W_i - SUM_i$
< 0
> 0
TH_i
= 0
INFEASIBLE REGION
4j
$\theta_i^n = 0$
NO
IS i=2
YES
ia = i+1
ia = i-1
NO
IS $\theta_{ia}^n=(D^n+1)$
YES
NO
IS $TH_i>(D^n-\theta_{ia}^n)$
YES
NO
IS $TH_i > D^n$
YES
$\theta_i^n=TH_i$
$\theta_i^n = D^n-\theta_{ia}^n$
$\theta_i^n = TH_i$
$\theta_i^n = D^n$
$SUM_i = SUM_i+\theta_i^n$
CONTINUE
CONTINUE
6C

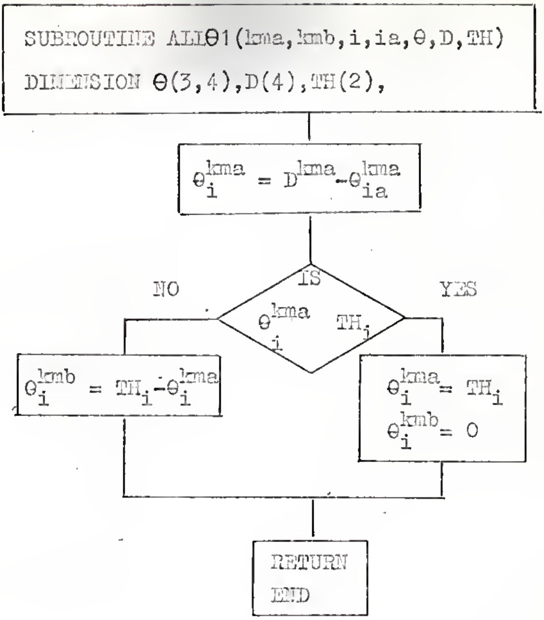

6C
SUMθ_3=0
DO k = 1,2
IS k=2
NO
YES
ka = k+1
ka = k-1
CALL ALIθ3(θ,k,W,D,nn,ka)
CONTINUE
DO n = 1,4
$\theta_3^n = D^n - \theta_1^n - \theta_2^n$
IS $\theta_3 < 0$
NO
YES
SUMθ_3 = SUMθ_3+θ_3^n
INFEASIBLE REGION
4j
CONTINUE
IS W_3 = SUMT$_3$
NO
YES
INFEASIBLE REGION
GO TO STEP 7
4j

Step 7

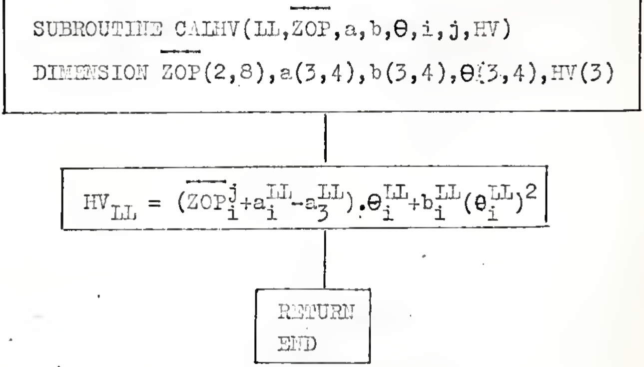

7
SUMF = 0
DO n = 1,4
SUMC^n = 0
DO i = 1,3
$F_i^n = a_i^n\theta_i^n + b_i^n(\theta_i^n)^2$
SUMC^n = SUMC^n+F_i^n
CONTINUE
SUMF = SUMF+SUMC^n
CONTINUE
GO TO STEP 8

STEP 8 & 9

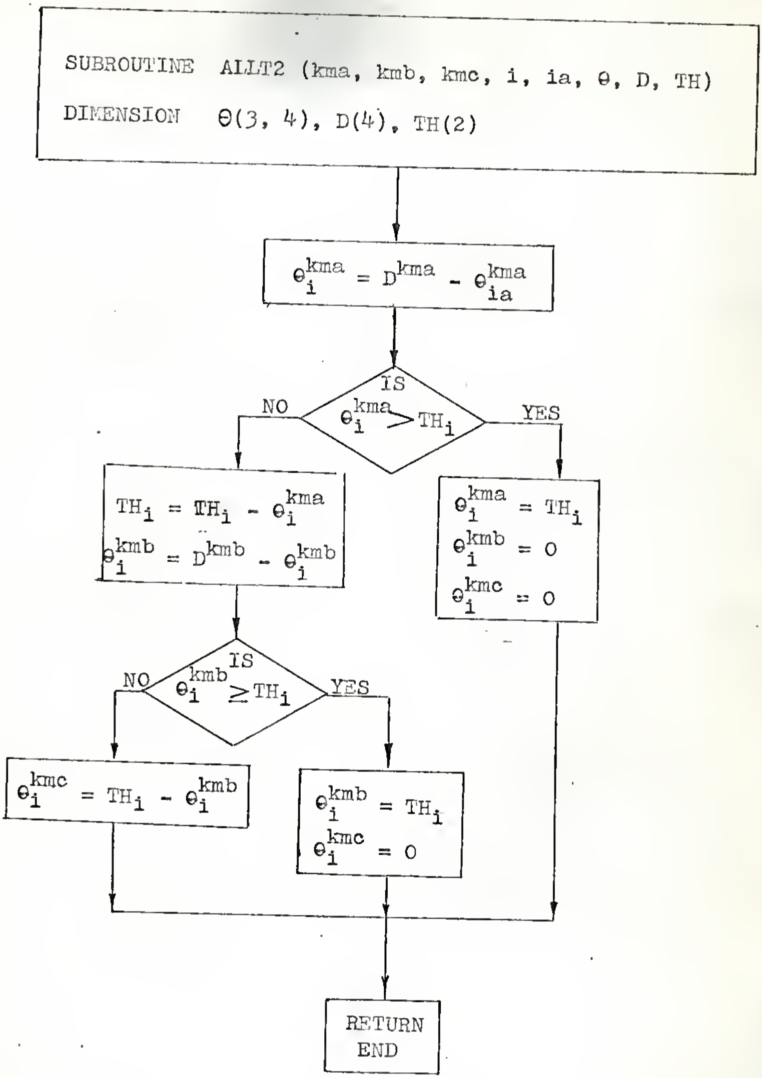

8
NO
SUMF SMIN
YES
SMIN = SUMF
jomin = jo
jmin = j
DO n = 1, 4
PUNCH θ_1^n, θ_2^n, θ_3^n
CONTINUE
PUNCH jomin, jmin, SMIN, SUMF
GO BACK TO STEP 5 FOR A NEW REGION OF SEARCH
4j

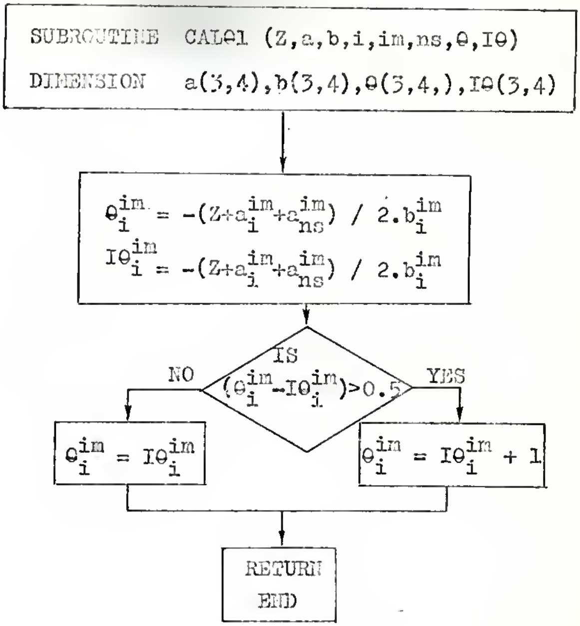


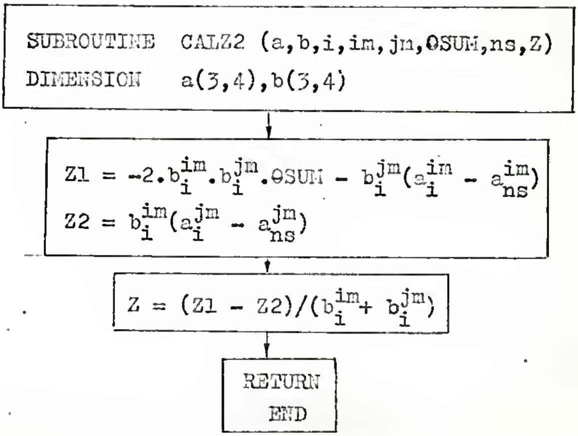
SUBROUTINE ALLΘ1(kma,kmb,i,ia,Θ,D,TH)
DIMENSION Θ(3,4),D(4),TH(2),
$\theta_i^{kma} = D^{kma} - \theta_{ia}^{kma}$
IS
NO
YES
θ_i^{kma} TH_i
$\theta_i^{kmb} = TH_i - \theta_i^{kma}$
$\theta_i^{kma} = TH_i$
$\theta_i^{kmb} = 0$
RETURN
END



SUBROUTINE CALHV(LL,$\overline{ZOP}$,a,b,Θ,i,j,HV)
DIMENSION $\overline{ZOP}$(2,8),a(3,4),b(3,4),Θ(3,4),HV(3)
$HV_{LL} = (\overline{ZOP}_i^j + a_i^{LL} - a_3^{LL}).\theta_i^{LL} + b_i^{LL}(\theta_i^{LL})^2$
RETURN
END


SUBROUTINE ALLT2 (kma, kmb, kmc, i, ia, θ, D, TH)
DIMENSION θ(3, 4), D(4), TH(2)
$\theta_i^{kma} = D^{kma} - \theta_{ia}^{kma}$
IS $\theta_i^{kma} > TH_i$
NO
YES
$TH_i = TH_i - \theta_i^{kma}$
$\theta_i^{kmb} = D^{kmb} - \theta_i^{kmb}$
$\theta_i^{kma} = TH_i$
$\theta_i^{kmb} = 0$
$\theta_i^{kmc} = 0$
IS $\theta_i^{kmb} \geq TH_i$
NO
YES
$\theta_i^{kmc} = TH_i - \theta_i^{kmb}$
$\theta_i^{kmb} = TH_i$
$\theta_i^{kmc} = 0$
RETURN
END


SUBROUTINE CALθ1 (Z,a,b,i,im,ns,θ,Iθ)
DIMENSION a(3,4),b(3,4),θ(3,4,),Iθ(3,4)
$\theta_i^{im} = -(Z+a_i^{im}+a_{ns}^{im}) \ / \ 2.b_i^{im}$
$I\theta_i^{im} = -(Z+a_i^{im}+a_{ns}^{im}) \ / \ 2.b_i^{im}$
IS
$(\theta_i^{im}-I\theta_i^{im})>0.5$
NO
YES
$\theta_i^{im} = I\theta_i^{im}$
$\theta_i^{im} = I\theta_i^{im} + 1$
RETURN
END


SUBROUTINE CALZ2 (a,b,i,im,jm,θSUM,ns,Z)
DIMENSION a(3,4),b(3,4)
$Z1 = -2.b_i^{im}.b_i^{jm}.\theta SUM - b_i^{jm}(a_i^{im} - a_{ns}^{im})$
$Z2 = b_i^{im}(a_i^{jm} - a_{ns}^{jm})$
$Z = (Z1 - Z2)/(b_i^{im}+ b_i^{jm})$
RETURN
END

```
SUBROUTINE CALZ3(a,b,ns,im,jm,km,θSUM,Z)

DIMENSION a(3,4), b(3,4)
```

$$Z1 = 2b_i^{im}.b_i^{jm}.b_i^{km}.\theta SUM$$

$$Z2 = b_i^{jm}.b_i^{km}(a_i^{im}-a_{ns}^{im})$$

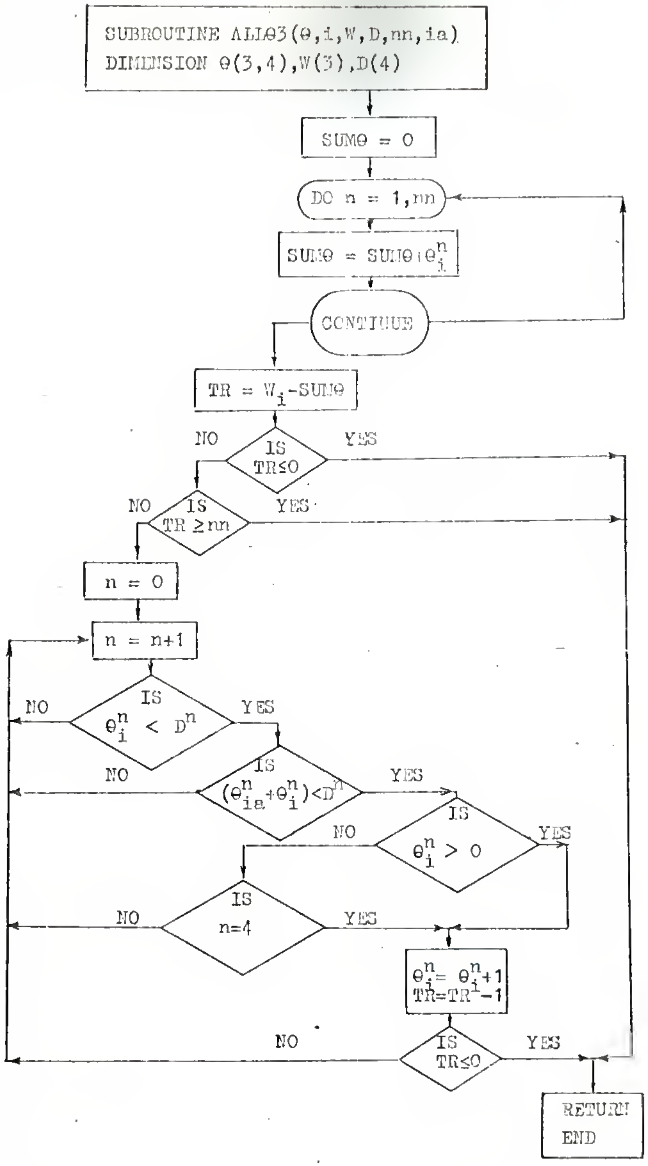
$$Z3 = b_i^{im}.b_i^{km}(a_i^{jm}-a_{ns}^{jm})$$

$$Z4 = b_i^{im}.b_i^{jm}(a_i^{km}-a_{ns}^{km})$$

$$Z5 = b_i^{jm}.b_i^{km}+b_i^{im}.b_i^{km}+b_i^{im}.b_i^{jm}$$

$$Z = \frac{-(Z1+Z2+Z3+Z4)}{Z5}$$

```
RETURN
END
```


SUBROUTINE ALLθ3(θ,i,W,D,nn,ia)
DIMENSION θ(3,4),W(3),D(4)
SUMθ = 0
DO n = 1,nn
SUMθ = SUMθ+θ$_i^n$
CONTINUE
TR = W$_i$-SUMθ
IS TR≤0
YES
NO
IS TR ≥nn
YES
NO
n = 0
n = n+1
IS θ$_i^n$ < D^n
NO
YES
IS (θ$_{ia}^n$+θ$_i^n$)<D^n
NO
YES
IS θ$_i^n$ > 0
NO
YES
IS n=4
NO
YES
θ$_i^n$= θ$_i^n$+1
TR=TR-1
IS TR≤0
NO
YES
RETURN
END

APPENDIX B SYMBOL TABLE FOR COMPUTER PROGRAM

Program Symbol	Mathematical Symbol	Explanation
A(I,N)	a_i^n	The coefficient of the first order theta in the cost function expression.
B(I,N)	b_i^n	The coefficient of the second order theta in the cost function expression.
BB(I,J)	b_i^n	Same as above but the superscript is ordered according to the value of the upper breaking value.
C		A counter to indicate whether the θ_i^n = equation of optimality have been allocated. It is use for allocating the severely constrained condition.
CKK		A counter of the number of the states with the severely constrained condition.
CO		A counter to indicate whether the ZOP_i^{jo} (3c) is out of order.
COUNT		The number of time step 6 is repeating when there is a severely constrained condition.
D(N)	D^n	The number of units of the resource required by the n^{th} demand point.
F(I,N)	$F_i^n(\theta_i^n)$	The cost incurred by transporting θ_i^n
H(I),HV(LL),HV(L1)	H_v^n	The Hamiltonian function.
I	i	Subscript for origin.
IA		Subscript of the next or preceding origin.

IM, IME		The index of the first stage to have a θ_i^n = equation of optimality condition.
IMK		The index of the first stage to have a severely constrained condition.
IT	θ_i^n	The integerized θ_i^n.
J		The subscript for the ordered region of search, ZOP_i^j.
JM, JME		The index of the second stage to have a θ_i^n = equation of optimality condition.
J1		Superscript of the ordered region of search, ZOP_i^{j1}.
JMIN		The optimal region of search of the second origin.
JOMIN		The optimal region of search of the first origin.
K		A counter for the number of free variation of θ_j^n condition; It is also used for subscript in place of i for a loop inside i-"DO-loop".
KC		A counter for the number of free variation of θ_i^n condition.
KE		A counter for the number of θ_i^n = equation of optimality condition.
KK		A counter for the number of severely constrained condition.
KKK		A counter for the number of stages in the origin which has D^n larger than W_i.
KMA		The index of the first stage to have a severely constrained condition.

P		A counter to indicate whether the θ_i^n with the severely condition has been allocated.
PM(I,N)		Identification for stages in the origin where D^n is larger than W_i.
Q		A counter to indicate whether the θ_i^n with the free variation of θ_i^n conditon has been allocated.
SMIN	C_{sN}	The current minimum total cost of transportation.
SUM(I)	x_i^n	The state variable, the total amount of resources which have been transported from the i^{th} depot to the first n stages (demand points).
SUM C(N)	$\sum_{i=1}^{S} F_i^n(\theta_i^n)$	The total cost of transporting θ_i^n from all depots to the n^{th} demand point.
SUM D(I)		Similar to SUM(I) except only $\theta_i^n = D^n$ are added.
SUM F	X_s^n	The total cost of transporting θ_i^n from all depots (origins) to the first n demand points.
SUM TT(I)		Similar to SUM(I) except only $\theta_i^n = D^n$ and $\theta\theta_i^n = D^n - O_{ia}^n$ (Where θ_{ia}^n = equation of optimality and $\theta\theta_{ia}^n$ has a severely constrained condition).
SUM T(3)		Similar to SUM(I) but i=3.
SUM W(I)		The constraint for the stages with $\theta_i^n = 0$.
T(I,N)	θ_i^n	The quantity of the resources sent from the i^{th} depot (origin) to the n^{th} demand point.
TH(I)		The available resource to be allocated to the free variation of θ_i^n condition.

TR		The remainder of θ_i^n in the origin due to integerization of θ_i^n.
TS(I)		The available θ_i^n in the origin when maximum available θ_i^n is allocated to the severely constrained condition.
TSU		Same as TS(I).
TSUM		The resource available for θ_i^n = equation of optimality condition in the state.
TT(I,N)		The θ_i^n in case where maximum available θ_i^n is allocated to the severely constrained condition.
TM1, TM2, TM3		The θ_i^n with the severely constrained conditions which give the minimum value for the Hamiltonion function.
W(I)	W_i	The total number of units of the resource available at the i^{th} depot.
WT	$\sum_{i=1}^{3} W_i = \sum_{n=1}^{4} D^n$	The total resource transported.
Z	Z_i^n	The value of Z_i^n used in finding θ_i^n = equation of optimality.
ZLOW(I,N), ZBL(I,J)	$\underline{Z}_i^n$	The lower limit of the breaking value, $\underline{Z}_i^n$, which is obtained by equating the derivative of H_v^n with respect to θ_i^n.
ZUP(I,N), ZBU(I,J)	$\bar{Z}_i^n$	The upper limit of the breaking value, $\bar{Z}_i^n$.
ZOP(I,J)		The regions of search.
ZZ		The dummy of Z_i^n value from the previous loop of Step 6.
Z1, Z2, Z3, Z4, Z5, Z6, Z7		The terms of the equation for calculation of the Z_i^n value.

KMB		The index of the second stage to have a severely constrained condition.
KMC		The index of the third stage to have a severely constrained condition.
KME		The index of the fourth stage to have a θ_1^n = equation of optimality condition.
L	i	Index for i.
LL, L1		Indexes for stages with severely, constrained condition.
M(K)		Indexes for stages with free variation of θ_1^n condition.
ME(KE)		Indexes for stages with θ_1^n = equation of optimality condition.
MK(KC)		Indexes for stages with severely constrained condition.
N, N1	n	Superscript of the demand points or stages n = 1, 2, ..., N.
NJ		Total number of regions of search for the second origin.
NJO		Total numbers of regions of search for the first origin.
NK(I)		Total number of regions of search in each origin.
NN	N	Total number of the demand points or stages.
NO		The index of the largest breaking value in the origin.
NS		Total number of the depots.
NJ1		Total number of regions of search in each origin.

APPENDIX C COMPUTER PROGRAM

```
//MASTER       JOB  00I504066C01,SKONCKATONGS          20,20,900
                                 Q            Q
// EXEC FTECLGKS
//FORT.SYSLIN DD UNIT=194
//FORT.SYSIN   DD   *
      DIMENSION ZOP(2,8),ZU(3,4),ZBL(2,4),NK(2),W(3),D(4)
      DIMENSION H(2),SUM(3),SUMW(3),TT(3,4),SUMD(2),ME(4)
      DIMENSION A(3,4),B(3,4),T(3,4),BB(3,4),ZUP(3,4)
      DIMENSION SUMTT(2),PM(3,4),MPC(4),ZLOW(3,4),MK(4)
      DIMENSION TH(2),SUMC(4),F(3,4),IT(3,4),SUMT(3),TS(2)
      DIMENSION HHV(6),ZZ(2),M(4),HV(3),ZBU(2,4)
  100 FORMAT(3F8.4,3F10.0)
  102 FORMAT(2HI=,I3,5H   J=,I3,16H        ZBU(I,J)=,F9.5)
  103 FORMAT(28HTHIS IS AN INFEASIBLE REGION)
  107 FORMAT(I4)
  108 FORMAT(3F15.3)
  101 FORMAT(2HI=,I3,4H  N=,I3,11H  ZUP(I,N)=,F9.3,
     C12H  ZLOW(I,N)=,F9.3)
  104 FORMAT(3HJO=,I2,5H   J=,I2,15H   ZOP(I-1,JO)=,F9.3,
     C12H   ZOP(I,J)=,F9.3)
  109 FORMAT(8H  JOMIN=,I2,8H   JMIN=,I2,9H    SMIN=,F10.2,
     C9H    SUMF=,F16.2)
  110 FORMAT(1X,11HTHIRDCHECK=,2F9.2)
  111 FORMAT(F10.3)
  112 FORMAT(2F15.6)
  113 FORMAT(2F9.1)
  114 FORMAT(1X,11HFIRSTCHECK=,4I3,9F6.1)
      LO=0
      NN=4
      NS=3
  900 DO 1 I=1,2
      DO 2 N=1,NN
      READ(1,100)A(I,N),B(I,N),A(NS,N),D(N),W(I),W(NS)
```

```
      IF(B(I,N))40,50,60
   40 T(I,N)=D(N)/2.
      ZUP(I,N)=-A(I,N)+A(NS,N)-2.*B(I,N)*T(I,N)
      ZLOW(I,N)=ZUP(I,N)
      GO TO 2
   50 ZUP(I,N)=-A(I,N)+A(NS,N)
      ZLOW(I,N)=ZUP(I,N)
      GO TO 2
   60 ZUP(I,N)=-A(I,N)+A(NS,N)
      ZLOW(I,N)=-A(I,N)+A(NS,N)-2.*B(I,N)*D(N)
    2 WRITE(2,101)I,N,ZUP(I,N),ZLOW(I,N)
      SMIN=1000000.
      DO 3 J=1,NN
      ZBU(I,J)=-1000.
      DO 4 N=1,NN
      IF(ZBU(I,J)-ZUP(I,N))41,51,61
   41 ZBU(I,J)=ZUP(I,N)
   51 NO=N
      GO TO 4
   61 GO TO 4
    4 CONTINUE
      ZU(I,NO)=ZUP(I,NO)
      ZUP(I,NO)=-1000
      BB(I,J)=B(I,NO)
      ZBL(I,J)=ZLOW(I,NO)
    3 WRITE(2,102)I,J,ZBU(I,J)
      K=0
      DO 5 J=1,NN
      IF(BB(I,J))42,52,62
   42 K=K+1
      ZOP(I,K)=ZBU(I,J)-.005
      GO TO 5
   52 K=K+1
      ZOP(I,K)=ZBU(I,J)
      K=K+1
```

```
      ZOP(I,K)=ZBU(I,J)-.005
      GO TO 5
   62 K=K+1
      ZOP(I,K)=ZBU(I,J)-.005
      K=K+1
      ZOP(I,K)=ZBL(I,J)-.005
    5 CONTINUE
      NK(I)=K
    1 WRITE(2,107)NK(I)
      NJO=NK(1)
      NJ=NK(2)
      I=1
      NJO=NJO-1
  316 CO=0.
      DO 300 JO=1,NJO
      IF(ZOP(I,JO)-ZOP(I,JO+1))340,360,360
  340 ZAP=ZOP(I,JO)
      ZOP(I,JO)=ZOP(I,JO+1)
      ZOP(I,JO+1)=ZAP
      CO=CO+1.
  360 GO TO 300
  300 CONTINUE
      IF(CO)361,351,361
  361 GO TO 316
  351 NJO=NJO+1
      DO 500 JO=1,NJO
  500 WRITE(2,111)ZOP(I,JO)
      I=2
      NJ=NJ-1
  317 CO=0.
      DO 301 JO=1,NJ
      IF(ZOP(I,JO)-ZOP(I,JO+1))343,363,363
  343 ZAP=ZOP(I,JO)
      ZOP(I,JO)=ZOP(I,JO+1)
      ZOP(I,JO+1)=ZAP
```

```
    CO=CC+1.
363 GU TC 301
301 CONTINUE
    IF(CO)364,354,364
364 GO TO 317
354 NJ=NJ+1
    DO 501 JO=1,NJ
501 WRITE(2,111)ZOP(I,JO)
    WT=W(1)+W(2)+W(3)
    DO 6 JO=1,NJO
    DO 8 J=1,NJ
    L=1
    I=L+1
    Q=0.
    C=0.
    P=0.
    COUNT=0.
    CKK=0.
    ZZ(1)=0.
    ZZ(2)=0.
    DO 7 K=1,3
    SUM(K)=0.
    SUMW(K)=WT
    DO 800 N=1,NN
800 PM(K,N)=0.
  7 CONTINUE
    WRITE(2,104)JO,J,ZOP(I-1,JO),ZOP(I,J)
    DO 9 N=1,NN
    ZUP(I,N)=ZU(I,N)
    ZUP(I-1,N)=ZU(I-1,N)
    IF(ZOP(I,J)-ZUP(I,N))45,55,65
 45 IF(ZOP(I,J)-ZLOW(I,N))46,56,56
 46 IF(ZOP(I-1,JO)-ZUP(I-1,N))47,57,57
 47 IF(ZOP(I-1,JO)-ZLOW(I-1,N))48,58,58
 48 H(I-1)=(ZOP(I-1,JO)+A(I-1,N)-A(NS,N))*D(N)+B(I-1,N)*D(N)**2
```

```
      H(I)=(ZOP(I,J)+A(I,N)-A(NS,N))*D(N)+B(I,N)*L(N)**2
      IF(H(I-1)-H(I))49,59,59
   49 IF(D(N)-W(I-1))830,830,850
  830 T(I,N)=0.
      T(I-1,N)=D(N)
      GO TO 116
  850 T(I,N)=D(N)-W(I-1)
      T(I-1,N)=W(I-1)
  801 PM(I-1,N)=1.
      PM(I,N)=1.
      GO TO 116
   59 IF(D(N)-W(I))831,831,851
  831 T(I-1,N)=0.
      T(I,N)=D(N)
      GO TO 116
  851 T(I,N)=W(I)
      T(I-1,N)=D(N)-W(I)
      GO TO 801
   58 T(I-1,N)=-(ZOP(I-1,JO)+A(I-1,N)-A(NS,N))/(2.*B(I-1,N))
      T(I,N)=D(N)+2.
      GO TO 9
   57 IF(D(N)-W(I))832,832,852
  832 T(I,N)=D(N)
      T(I-1,N)=0.
      GO TO 116
  852 T(I,N)=W(I)
      T(I-1,N)=C.
      GO TO 801
   56 IF(ZOP(I-1,JO)-ZUP(I-1,N))70,80,90
   70 IF(ZOP(I-1,JO)-ZLOW(I-1,N))71,81,81
   71 T(I,N)=-(ZOP(I,J)+A(I,N)-A(NS,N))/(2.*B(I,N))
      T(I-1,N)=D(N)+2.
      GO TO 9
   81 T(I,N)=-(ZOP(I,J)+A(I,N)-A(NS,N))/(2.*B(I,N))
      T(I-1,N)=-(ZOP(I-1,JO)+A(I-1,N)-A(NS,N))/(2.*B(I-1,N))
```

```
      GO TO 9
   8D T(I,N)=-(ZOP(I,J)+A(I,N)-A(NS,N))/(2.*B(I,N))
      T(I-1,N)=D(N)+1.
      GO TD 9
   90 T(I,N)=-(ZDP(I,J)+A(I,N)-A(NS,N))/(2.*B(1,N))
      T(I-1,N)=0.
      GD TO 1I6
   55 IF(ZDP(1-1,JD)-ZUP(1-1,N))72,82,92
   72 IF(ZOP(I-1,JO)-ZLOW(I-1,N))73,83,83
   73 IF(D(N)-W(I-1))833,833,853
  833 T(1,N)=0.
      T(I-1,N)=D(N)
      GO TD 116
  853 T(I,N)=0.
      T(1-1,N)=W(I-1)
      GO TO 801
   83 T(I,N)=D(N)+1.
      T(I-I,N)=-(ZDP(I-I,JO)+A(I-1,N)-A(NS,N))/(2.*B(I-1,N))
      GO TD 9
   82 T(1,N)=D(N)+1.
      T(I-1,N)=D(N)+1.
      GD TD 9
   92 T(1,N)=D(N)+1.
      T(I-1,N)=0.
      GO TD 116
   65 IF(ZDP(I-I,JO)-ZUP(I-1,N))74,84,94
   74 IF(ZOP(1-1,JD)-ZLOW(I-I,N))75,85,85
   75 GO TC 73
   85 T(I,N)=0.
      T(I-I,N)=-(ZDP(I-1,JO)+A(I-1,N)-A(NS,N))/(2.*B(I-I,N))
      GO TC 116
   84 T(I,N)=0.
      T(I-I,N)=D(N)+1.
      GO TC II6
   94 T(I,N)=0.
```

```
      T(1-1,N)=0.
      T(NS,N)=D(N)
      GO TO 117
  116 DO 10 K=1,2
      IF(PM(K,N)-1.)834,844,834
  834 IF(T(K,N)-D(N))76,86,76
  844 IF(T(K,N)-W(K))835,86,835
  835 IF(T(K,N))836,87,836
  836 IF(T(K,N)-D(N))846,10,10
  846 SUM(K)=SUM(K)+T(K,N)
      GO TO 10
   76 IF(T(K,N))77,87,77
   77 GO TO 10
   87 SUMW(K)=SUMW(K)-D(N)
      GO TO 10
   86 SUM(K)=SUM(K)+T(K,N)
      SUMW(3)=SUMW(3)-T(K,N)
   10 CONTINUE
      GO TO 9
  117 T(3,N)=D(N)
      SUM(3)=SUM(3)+T(3,N)
    9 WRITE(2,112)T(1,N),T(2,N)
      DO 11 K=1,3
      IF(SUM(K)-W(K))78,78,98
   78 IF(SUMW(K)-W(K))79,89,
   79 WRITE(2,103)
      IX=1
      DO 405 N=1,NN
      WRITE(2,113)T(1,N),T(2,N)
  405 CONTINUE
      WRITE(2,107)IX
      GO TO 8
   89 KKK=0.
      DO 802 N1=1,NN
      IF(T(K,N1))838,802,838
```

```
838 KKK=KKK+1
    MPC(KKK)=N1
802 CONTINUE
    IF(KKK-1)11,849,11
849 N=MPC(1)
    IF(PM(K,N))840,11,840
840 IF(K-1)870,880,870
870 KA=K-1
    GO TO 810
880 KA=K+1
810 IF(T(K,N)-W(K))871,11,871
871 IF(D(N)-W(K))11,11,882
882 T(K,N)=W(K)
    T(KA,N)=D(N)-W(K)
    GO TO 11
 98 GO TO 79
 11 CONTINUE
200 DO 12 I=1,2
    SUM(I)=0.
    SUMD(I)=0.
    SUMTT(I)=0.
    K=0
    KE=0
    KK=0
    DO 13 N=1,NN
    IF(PM(I,N)-1.)837,847,837
837 IF(T(I,N)-(D(N)+1.))140,150,160
847 GO TO 151
140 IF(T(I,N)-(D(N)))141,151,151
141 IF(T(I,N))142,152,162
142 GO TO 79
152 GO TO 118
162 KE=KE+1
    ME(KE)=N
118 SUM(I)=SUM(I)+T(I,N)
```

```
      GO TO 13
  151 SUMD(I)=SUMD(I)+T(I,N)
      SUMTT(I)=SUMTT(I)+T(I,N)
      GO TO 118
  150 K=K+1
      M(K)=N
      GO TO 13
  160 KK=KK+1
      IF(I-1)143,153,143
  143 IA=I-1
      J1=J
      NJ1=NJ
  119 IF(C)144,154,144
  144 IF(TS(I))347,357,357
  347 GO TO 13
  357 T(I,N)=D(N)-T(IA,N)
      P=P-1.
      GO TO 151
  154 TT(I,N)=D(N)-T(IA,N)
      P=P+1.
      SUMTT(I)=SUMTT(I)+TT(I,N)
      GO TO 13
  153 IA=I+1
      J1=JO
      NJ1=NJO
      GO TO 119
   13 CONTINUE
      IF(C)145,155,145
  145 IF(KK)348,348,368
  348 GO TO 12
  368 IF(TS(I))349,359,359
  359 GO TO 12
  349 KC=0
      TH(I)=W(I)-SUM(I)
      IF(TH(I))370,380,380
```

```
370 GO TO 79
380 DO 303 N=1,4
    IF(T(I,N)-(D(N)+2.))371,381,371
371 GO TO 303
381 KC=KC+1
    MK(KC)=N
303 CONTINUE
    IF(KC-2)373,383,393
373 IMK=MK(1)
    T(I,IMK)=TH(I)
    P=P-1.
    GO TO 12
383 P=P-2.
    DO 304 L1=1,2
    IF(L1-2)374,384,374
374 KMA=MK(1)
    KMB=MK(2)
    GO TO 319
384 KMA=MK(2)
    KMB=MK(1)
319 WRITE(3,114)KMA,KMB,I,IA,T(I,1),T(I,2),T(I,3),T(I,4),
   CD(1),D(2),D(3),D(4),,TH(I)
    CALL ALLT1 (KMA,KMB,I,IA,T,D,TH)
    WRITE(3,110)T(I,KMA),T(I,KMB)
    LL=KMA
    HHV(L1)=0.
    CALL CALHV (LL,ZOP,A,B,T,I,J1,HV)
    HHV(L1)=HHV(L1)+HV(LL)
    LL=KMB
    CALL CALHV (LL,ZOP,A,B,T,I,J1,HV)
    HHV(L1)=HHV(L1)+HV(LL)
    IF(LI-2)375,385,375
375 TM1=T(I,KMA)
    TM2=T(I,KMB)
    GO TO 304
```

```
304 CONTINUE
385 IF(HHV(1)-HHV(2))376,386,386
376 T(I,KMA)=TM1
    T(I,KMB)=TM2
    GO TO 12
386 GO TO 12
393 P=P-3.
    DO 305 L1=1,6
    IF(L1-2)377,387,397
397 IF(L1-4)378,388,398
398 IF(L1-6)379,389,389
389 KMA=MK(3)
    KMB=MK(2)
    KMC=MK(1)
    GO TO 320
379 KMA=MK(3)
    KMB=MK(1)
    KMC=MK(2)
    GO TO 320
388 KMA=MK(2)
    KMB=MK(3)
    KMC=MK(1)
    GO TO 320
378 KMA=MK(2)
    KMB=MK(1)
    KMC=MK(3)
    GO TO 320
387 KMA=MK(1)
    KMB=MK(3)
    KMC=MK(2)
    GO TO 320
377 KMA=MK(1)
    KMB=MK(2)
    KMC=MK(3)
    GO TO 320
```

```
320 CALL ALLT2 (KMA,KMB,KMC,I,IA,T,D,TH)
    HHV(L1)=0.
    LL=KMA
    CALL CALHV (LL,ZOP,A,B,T,I,J1,HV)
    HHV(L1)=HHV(L1)+HV(LL)
    LL=KMB
    CALL CALHV (LL,ZOP,A,B,T,I,J1,HV)
    HHV(L1)=HHV(L1)+HV(LL)
    LL=KMC
    CALL CALHV (LL,ZOP,A,B,T,I,J1,HV)
    HHV(L1)=HHV(L1)+HV(LL)
    IF(L1-2)440,450,450
440 TM1=T(I,KMA)
    TM2=T(I,KMB)
    TM3=T(I,KMC)
    GO TO 305
450 IF(HHV(L1-1)-HHV(L1))441,451,451
441 T(I,KMA)=TM1
    T(I,KMB)=TM2
    T(I,KMC)=TM3
451 GO TC 305
305 CONTINUE
    GO TC 12
155 IF(K-1)147,157,167
147 IF(KE-1)148,158,168
148 GO TO 12
158 IM=ME(1)
    IF(KK)470,470,490
470 T(I,IM)=W(I)-SUMD(I)
    Z=-2.*B(I,IM)*T(I,IM)-(A(I,IM)-A(NS,IM))
744 IF(Z-ZOP(I,J1))745,79,79
745 IF(J1-NJ1)746,12,12
746 IF(Z-ZOP(I,J1+1))79,79,12
490 TSUM=W(I)-SUMD(I)
    TSUI=W(I)-SUMTT(I)
```

```
      IF(TSUI)471,481,481
  471 Z=ZOP(I,J1)
      CALL CALT1 (Z,A,B,I,KM,NS,T,IT)
      T(I,IM)=T(I,IM)+1.
      GO TO 323
  481 Z=-2.*8(I,IM)*TSUI-(A(I,IM)-A(NS,IM))
      CALL CALT1 (Z,A,B,I,IM,NS,T,IT)
      GO TC 323
  168 IF(KE-3)149,159,169
  149 TSUM=W(I)-SUMD(I)
      IM=ME(1)
      JM=ME(2)
      IF(KK)472,472,492
  472 CALL CALZ2 (A,B,I,IM,JM,TSUM,NS,Z)
      CALL CALT1 (Z,A,B,I,IM,NS,T,IT)
      T(I,JM)=W(I)-SUMO(I)-T(I,IM)
      GO TO 744
  492 IF(TSUI)473,483,483
  473 Z=ZOP(I,J1)
      CALL CALT1 (Z,A,B,I,IM,NS,T,IT)
      CALL CALT1 (Z,A,B,I,JM,NS,T,IT)
      T(I,IM)=T(I,IM)+1.
      T(I,JM)=TII,JM)+1.
      GO TC 323
  483 CALL CALZ2 IA,B,I,IM,JM,TSUI,NS,Z)
      CALL CALT1 IZ,A,B,I,IM,NS,T,IT)
      CALL CALT1 (Z,A,B,I,JM,NS,T,IT)
      GO TO 323
  159 TSUM=W(I)-SUMO(I)
      IM=ME(1)
      JM=ME(2)
      KM=ME(3)
      IF(KK)474,474,494
  474 CALL CALZ3 (A,B,NS,IM,JM,KM,TSUM,Z)
      CALL CALT1 (Z,A,B,I,IM,NS,T,IT)
```

```
      CALL CALT1 (Z,A,B,I,JM,NS,T,IT)
      T(I,KM)=W(I)-SUMD(I)-T(I,IM)-T(I,JM)
      GO TO 744
  494 IF(TSUI)475,485,485
  475 Z=ZOP(I,J1)
      CALL CALT1 (Z,A,B,I,IM,NS,T,IT)
      CALL CALT1 (Z,A,B,I,JM,NS,T,IT)
      CALL CALT1 (Z,A,B,I,KM,NS,T,IT)
      T(I,IM)=T(I,IM)+1.
      T(I,JM)=T(I,JM)+1.
      T(I,KM)=T(I,KM)+1.
      GO TO 323
  485 CALL CALZ3 (A,B,NS,IM,JM,KM,TSUM,Z)
      CALL CALT1 (Z,A,B,I,IM,NS,T,IT)
      CALL CALT1 (Z,A,B,I,JM,NS,T,IT)
      CALL CALT1 (Z,A,B,I,KM,NS,T,IT)
      GO TO 323
  323 IF(CCUNT)445,445,465
  445 CKK=CKK+1.
  465 IF(ZZ(I)-Z)447,457,447
  447 ZZ(I)=Z
      GO TO 12
  457 IF(Z-ZOP(I,J1))448,448,468
  448 IF(J1-NJ1)543,469,469
  543 IF(Z-ZOP(I,J1+1))468,468,469
  469 TS(I)=TSUI
      CKK=CKK-1.
      GO TO 12
  468 GO TO 79
  169 IM=ME(1)
      JM=ME(2)
      KM=ME(3)
      LM=ME(4)
      Z1=2.*B(I,IM)*B(I,JM)*B(I,KM)*B(I,LM)*W(I)
      Z2=B(I,JM)*B(I,KM)*B(I,LM)*(A(I,IM)-A(NS,IM))
```

```
      Z3=B(I,IM)*B(I,KM)*B(I,LM)*(A(I,JM)-A(NS,JM))
      Z4=B(I,IM)*B(I,JM)*B(I,LM)*(A(I,KM)-A(NS,KM))
      Z5=B(I,IM)*B(I,JM)*B(I,KM)*(A(I,LM)-A(NS,LM))
      Z6=B(I,JM)*B(I,KM)*B(I,LM)+B(I,IM)*B(I,KM)*B(I,LM)
      Z7=B(I,IM)*B(I,JM)*B(I,LM)+B(I,IM)*B(I,JM)*B(I,KM)
      Z=-(Z1+Z2+Z3+Z4+Z5)/(Z6+Z7)
      IT(I,IM)=-(Z+A(I,IM)-A(NS,IM))/(2.*B(I,IM))
      T(I,IM)=IT(I,IM)
      IT(I,JM)=-(Z+A(I,JM)-A(NS,JM))/(2.*B(I,JM))
      T(I,JM)=IT(I,JM)
      IT(I,KM)=-(Z+A(I,KM)-A(NS,KM))/(2.*B(I,KM))
      T(I,KM)=IT(I,KM)
      T(I,LM)=W(I)-T(I,IM)-T(I,JM)-T(I,KM)
      GO TO 12
  157 IM=M(K)
      IF(KE-1)170,180,190
  170 IF(KK)171,181,171
  171 Q=Q+1.
      GO TO 12
  181 T(I,IM)=W(I)-SUMD(I)
      GO TO 12
  180 IME=ME(1)
      IT(I,IME)=-(ZUP(I,IM)+A(I,IME)-A(NS,IME))/(2.*B(I,IME))
      T(I,IME)=IT(I,IME)
      IF(KK)172,182,172
  172 Q=Q+I.
      GO TO 12
  182 T(I,IM)=W(I)-SUMD(I)-T(I,IME)
      GO TO 12
  190 IF(KE-3)173,183,183
  173 IME=ME(1)
      JME=ME(2)
      IT(I,IME)=-(ZUP(I,IM)+A(I,IME)-A(NS,IME))/(2.*B(I,IME))
      T(I,IME)=IT(I,IME)
      IT(I,JME)=-(ZUP(I,IM)+A(I,JME)-A(NS,JME))/(2.*B(I,JME))
```

```
      T(I,JME)=IT(I,JME)
      IF(KK)174,184,174
  174 Q=Q+1.
      GO TO 12
  184 T(I,IM)=W(I)-SUMO(I)-T(I,IME)-T(I,JME)
      GO TO 12
  183 IME=ME(1)
      JME=ME(2)
      KME=ME(3)
      IT(I,IME)=-(ZUP(I,IM)+A(I,IME)-A(NS,IME))/(2.*B(I,IME))
      T(I,IME)=IT(I,IME)
      IT(I,JME)=-(ZUP(I,IM)+A(I,JME)-A(NS,JME))/(2.*B(I,JME))
      T(I,JME)=IT(I,JME)
      IT(I,KME)=-(ZUP(I,IM)+A(I,KME)-A(NS,KME))/(2.*B(I,KME))
      T(I,KME)=IT(I,KME)
      T(I,IM)=W(I)-T(I,IME)-T(I,JME)-T(I,KME)
      GO TO 12
  167 IF(KE-1)175,185,195
  175 Q=Q+1.
      GO TO 12
  185 IME=ME(1)
      IT(I,IME)=-(ZUP(I,IM)+A(I,IME)-A(NS,IME))/(2.*B(I,IME))
      T(I,IME)=IT(I,IME)
      GO TO 175
  195 IME=ME(1)
      JME=ME(2)
      IT(I,IME)=-(ZUP(I,IM)+A(I,IME)-A(NS,IME))/(2.*B(I,IME))
      T(I,IME)=IT(I,IME)
      IT(I,JME)=-(ZUP(I,IM)+A(I,JME)-A(NS,JME))/(2.*B(I,JME))
      T(I,JME)=IT(I,JME)
   12 CONTINUE
      IF(CKK)454,454,444
  444 COUNT=COUNT+1.
      GO TO 200
  454 IF(P)177,187,177
```

```
177 C=C+1.
    P=P/(COUNT+1.)
    GO TO 200
187 IF(Q)178,188,178
178 Q=Q/(COUNT+1.)
    GO TO 120
188 GO TO 121
120 DO 14 I=1,2
    DO 15 N=1,NN
    IF(T(I,N)-(D(N)+1.))240,250,240
240 GO TO 15
250 TH(I)=W(I)-SUM(I)
    IF(TH(I))241,251,261
241 GO TO 79
251 T(I,N)=0.
261 IF(I-2)242,252,242
242 IA=I+1
    GO TO 122
252 IA=I-1
122 IF(T(IA,N)-(D(N)+1.))243,253,243
243 IF(TH(I)-(D(N)-T(IA,N)))244,244,264
244 T(I,N)=TH(I)
    GO TO 123
264 T(I,N)=D(N)-T(IA,N)
    GO TO 123
253 IF(TH(I)-D(N))245,245,265
245 T(I,N)=TH(I)
    GO TO 123
265 T(I,N)=D(N)
123 SUM(I)=SUM(I)+T(I,N)
 15 CONTINUE
 14 CONTINUE
121 SUMT(3)=0.
    DO 607 K=1,2
    IF(K-1)642,642,652
```

```
642 KA=K+1
    GU TC 607
652 KA=K-1
607 CALL ALLT3 (T,K,W,D,NN,KA)
    DO 16 N=1,NN
    T(3,N)=D(N)-T(1,N)-T(2,N)
    IF(T(3,N))246,256,256
246 GO TO 79
256 SUMT(3)=SUMT(3)+T(3,N)
 16 CONTINUE
    IF(W(3)-SUMT(3))247,257,247
247 GO TO 79
257 SUMF=0.
    DO 17 N=1,NN
    SUMC(N)=0.
    DO 18 I=1,3
    F(I,N)=A(I,N)*T(I,N)+B(I,N)*T(I,N)**2
    SUMC(N)=SUMC(N)+F(I,N)
 18 CONTINUE
    SUMF=SUMF+SUMC(N)
 17 CONTINUE
    IF(SUMF-SMIN)248,248,258
248 SMIN=SUMF
    JOMIN=JO
    JMIN=J
    GO TO 258
258 DO 19 N=1,NN
    WRITE(2,108)T(1,N),T(2,N),T(3,N)
 19 CONTINUE
    WRITE(2,109)JOMIN,JMIN,SMIN,SUMF
  8 CONTINUE
  6 CONTINUE
    LO=LO+1
    NN=3
    IF(LO-1)921,922,921
```

```
922 GO TO 900
921 STOP
    END
    SUBROUTINE ALLT1(KMA,KMP,I,IA,T,O,TH)
    DIMENSION T(3,4),D(4),TH(2)
110 FORMAT(1X,11HTHIRDCHECK=,2F9.2)
115 FORMAT(1X,12HSECONDCHECK=,F9.2)
    T(I,KMA)=0(KMA)-T(IA,KMA)
    WRITE(3,115)T(I,KMA)
    IF(T(I,KMA)-TH(I))442,442,462
442 T(I,KMB)=TH(I)-T(I,KMA)
    GO TO 321
462 T(I,KMA)=TH(I)
    T(I,KMB)=0.
321 WRITE(3,110)T(I,KMA),T(I,KMB)
    RETURN
    END
    SUBROUTINE CALHV(LL,ZOP,A,B,I,J,HV)
    DIMENSION ZOP(2,8),A(3,4),B(3,4),T(3,4),HV(3)
    HV(LL)=(ZOP(I,J)+A(I,LL)-A(3,LL))*T(I,LL)+B(I,LL)*T(I,LL)**2
    RETURN
    END
    SUBROUTINE ALLT2(KMA,KMB,KMC,I,IA,T,D,TH)
    DIMENSION T(3,4),D(4),TH(2)
    T(I,KMA)=D(KMA)-T(IA,KMA)
    IF(T(I,KMA)-TH(I))443,443,463
443 TH(I)=TH(I)-T(I,KMA)
    T(I,KMB)=D(KMB)-T(I,KMB)
    IF(T(I,KMB)-TH(I))444,444,464
444 T(I,KMC)=TH(I)-T(I,KMB)
    GO TO 322
464 T(I,KMB)=TH(I)
    T(I,KMC)=0.
    GO TO 322
463 T(I,KMA)=TH(I)
```

```
      T(I,KMB)=0.
      T(I,KMC)=0.
  322 RETURN
      END
      SUBROUTINE ALLT3(T,I,W,D,NN,IA)
      DIMENSION T(3,4),W(3),D(4)
      SUMT=0.
      DO 1 N=1,NN
    1 SUMT=SUMT+T(I,N)
      TR=W(I)-SUMT
      IF(TR)20,20,10
   10 IF(TR-NN)11,20,20
   11 N=0
    2 N=N+1
      IF(T(I,N)-D(N))12,2,2
   12 IF((T(IA,N)+T(I,N))-D(N))13,2,2
   13 IF(T(I,N))3,3,14
    3 IF(N-4)2,14,14
   14 T(I,N)=T(I,N)+1.
      TR=TR-1.
      IF(TR)20,20,2
   20 RETURN
      END
      SUBROUTINE CALZ3(A,B,I,NS,IM,JM,KM,TSUM,Z)
      DIMENSION A(3,4),B(3,4)
      Z1=2.*B(I,IM)*B(I,JM)*B(I,KM)*TSUM
      Z2=B(I,JM)*B(I,KM)*(A(I,IM)-A(NS,IM))
      Z3=B(I,IM)*B(I,KM)*(A(I,JM)-A(NS,JM))
      Z4=B(I,IM)*B(I,JM)*(A(I,KM)-A(NS,KM))
      Z5=B(I,JM)*B(I,KM)+B(I,IM)*B(I,KM)+B(I,IM)*B(I,JM)
      Z=-(Z1+Z2+Z3+Z4)/Z5
      RETURN
      END
      SUBROUTINE CALZ2(A,B,I,IM,JM,TSUM,NS,Z)
      DIMENSION A(3,4),B(3,4)
```

```
      Z1=-2.*B(I,IM)*B(I,JM)*TSU -B(I,JM)*(A(I,IM)-A(NS,IM))
      Z2=B(I,IM)*(A(I,JM)-A(NS,J  ))
      Z=(Z1-Z2)/(B(I,I  )+ (I,JM))
      RETURN
      END
      SUBROUTINE CALT1(Z,A,B,I,IM,NS,T,IT)
      DIMENSION A(3,4),B(3,4),T(3,4),IT(3,4)
      IT(I,IM)=-(Z+A(I,IM)-A(NS,IM))/(2.*B(I,IM))
      T(I,IM)=-(Z+A(I,IM)-A( S,I ))/(2.*B(I,IM))
      IF(T(I,IM)-IT(I,IM)-0.5)1,2,2
    1 T(I,IM)=IT(I,IM)
      GO TO 3
    2 T(I,IM)=IT(I,IM)+1.
    3 RETURN
      END
/*
  2.5000  0.0000  2.0000      50.     130.     80.
  2.5000 -0.0500  5.0000      70.     130.     80.
  3.5000  0.0000  1.0000     100.     130.     80.
  4.0000   .0300  3.0000      80.     130.     80.
  1.0000 -0.0300  2.0000      50.      90.     80.
  1.5000  0.0000  5.0000      70.      90.     80.
  3.0000  0.0200  1.0000     100.      90.     80.
  2.0000  1.0000  3.0000      80.      90.     80.
//GO.SYSIN     DD   *
  2.5000  0.0000  1.0000      20.      50.     40.
  3.0000  0.0100  9.0000      60.      50.     40.
  6.0000  0.0000  6.6000      40.      50.     40.
  2.6000  0.0000  1.0000      20.      30.     40.
  2.7000  0.0000  9.0000      60.      30.     40.
  5.0000  0.0100  6.6000      40.      30.     40.
//LKED.SYSLMOD DD UNIT=193
/*
```

APPENDIX D RESULTS

EXAMPLE 1 3 ORIGINS AND 4 DEMAND POINTS

```
I=  1  N=  1  ZUP(I,N)=    -0.500  ZLOW(I,N)=    -0.500
I=  1  N=  2  ZUP(I,N)=     6.000  ZLOW(I,N)=     6.000
I=  1  N=  3  ZUP(I,N)=    -2.500  ZLOW(I,N)=    -2.500
I=  1  N=  4  ZUP(I,N)=    -1.000  ZLOW(I,N)=    -5.800

I=  1   J=  1        ZBU(I,J)=  6.00000
I=  1   J=  2        ZBU(I,J)= -0.50000
I=  1   J=  3        ZBU(I,J)= -1.00000
I=  1   J=  4        ZBU(I,J)= -2.50000

I=  2  N=  1  ZUP(I,N)=     2.500  ZLOW(I,N)=     2.500
I=  2  N=  2  ZUP(I,N)=     3.500  ZLOW(I,N)=     3.500
I=  2  N=  3  ZUP(I,N)=    -2.000  ZLOW(I,N)=    -6.000
I=  2  N=  4  ZUP(I,N)=     1.000  ZLOW(I,N)=  -159.000

I=  2   J=  1        ZBU(I,J)=  3.50000
I=  2   J=  2        ZBU(I,J)=  2.50000
I=  2   J=  3        ZBU(I,J)=  1.00000
I=  2   J=  4        ZBU(I,J)= -2.00000

     5.995
    -0.500
    -0.505
    -1.005
    -2.500
    -2.505
    -5.805
     3.500
     3.495
     2.495
     0.995
```

```
    -2.005
    -6.005
  -159.005

JO= 5   J= 5   ZOP(I-1,JJ)=   -2.500   ZOP(I,J)=   -2.005

       T(1,N)          T(2,N)
        0.0            50.000000
     70.000000          0.0
    101.000000          0.124979
     25.000000          1.502500

          T(1,N)          T(2,N)          T(3,N)
           0.0            50.000           0.0
          70.000           0.0             0.0
          35.000          38.000          27.000
          25.000           2.000          53.000

  JOMIN= 5   JMIN= 5    SMIN=    483.13    SUMF=    483.13
```

EXAMPLE 2 3 ORIGINS AND 3 DEMAND POINTS

```
I=  1  N=  1  ZUP(I,N)=   -1.500  ZLOW(I,N)=   -1.500
I=  1  N=  2  ZUP(I,N)=    6.000  ZLOW(I,N)=    4.800
I=  1  N=  3  ZUP(I,N)=    0.600  ZLOW(I,N)=    0.600

I=  1   J=  1        ZBU(I,J)=  6.00000
I=  1   J=  2        ZBU(I,J)=  0.60000
I=  1   J=  3        ZBU(I,J)= -1.50000

I=  2  N=  1  ZUP(I,N)=   -1.600  ZLOW(I,N)=   -1.600
I=  2  N=  2  ZUP(I,N)=    6.300  ZLOW(I,N)=    6.300
I=  2  N=  3  ZUP(I,N)=    1.600  ZLOW(I,N)=    0.800

I=  2   J=  1        ZBU(I,J)=  6.30000
I=  2   J=  2        ZBU(I,J)=  1.60000
I=  2   J=  3        ZBU(I,J)= -1.60000

     5.995
     4.795
     0.600
     0.595
    -1.500
    -1.505
     6.300
     6.295
     1.595
     0.795
    -1.600
    -1.605

JO= 3   J= 2   ZOP(I-1,JO)=    0.600   ZOP(I,J)=    6.295

     T(I,J)        T(2,N)
     0.0           0.
```

```
50.000000      10.000000
41.000000       0.0
```

T(1,N)	T(2,N)	T(3,N)
0.0	0.0	20.000
30.000	30.000	0.0
20.000	0.0	20.000

```
JOMIN= 3   JMIN= 2    SMIN=    452.00    SUMF=    452.00
```

COMPUTER PROGRAM FOR AN OPTIMUM TRANSPORTATION SOLUTION BY THE DISCRETE MAXIMUM PRINCIPLE

by

PRADIT KONGKATONG

B.Sc. (Chemical Technology), Chulalongkorn University
Bangkok, Thailand, 1963

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

1968

The transportation problem is to minimize the cost of sending a resource from s orgins, where the resource is located, to N destinations (demand points) where the demand for this resource exists. Suppose that there is one type of resource and that its total supply is equal to the total demand.

The aim of this report is to explore the application of a discrete version of the maximum principle to the problem and to modify and define some of the concepts so that a computer program may be written for solving a problem with a large number of origins and demand points.

The computer program is written primarily for a problem with three origins and four demand points with one origin consisting of linear cost functions. This can in some instances be generalized for 3 origins and N demand points.

A problem with three origins and four demand points is used as an example to test the logic of the computer program.