Wise

UNIVERSITY OF ILLINOIS
DIGITAL COMPUTER

REVISED LIBRARY ROUTINE M 15 - 183

TITLE: Williams Memory Routine for Linear Programming by the Simplex Method (SADOI or DOI)

TYPE: Complete Floating Point Program

REFERENCES:

"Introduction to Linear Programming", Charnes, Cooper, and Henderson; Wiley (1953).

"Activity Analysis of Production and Allocation", Koopmans, T. C. editor; Cowles Commission Monograph No. 13, Wiley (1951).

"Linear Programming Methods", Heady and Candler; Iowa State College Press (1958).

"Linear Programming" Kurt Eissman, Quarterly of Applied Mathematics, Vol. 13, October, 1955.

DESCRIPTION: This program is suitable for maximization or minimization problems.

Maximization Problem

Given a linear functional:

(1) $Z = C_1X_1 + C_2X_2 + \dots + C_jX_j + \dots + C_nX_n$

which is to be maximized subject to the restraints

$$\begin{aligned} &A_{11}X_1 + A_{12}X_2 + \dots + A_{1n}X_n \leq B_1 \\ &\dots \\ (2)\quad &A_{21}X_1 + A_{22}X_2 + \dots + A_{2n}X_n \leq B_2 \\ &A_{m1}X_1 + A_{m2}X_2 + \dots + A_{mn}X_n \leq B_m \end{aligned}$$

and $X_i \geq 0$.

Whenever an inequality appears a new X_i is added to that relation making it an equality. The C_i value for these new X_i's are equal to zero. Relation (2) is then written:

$$\begin{aligned} &A_{11}X_1 + \dots + A_{1n}X_n + X_{n+1} + 0 + \dots + 0 = B_1 \\ &\dots \\ (3)\quad &A_{21}X_1 + \dots + A_{2n}X_n + 0 + X_{n+2} + \dots + 0 = B_2 \\ &A_{m1}X_1 + \dots \quad A_{mn}X_n + 0 + \dots + 0 + X_p = B_m \end{aligned}$$

Relation (1) is unchanged, i.e.

$$Z = C_1X_1 + \dots + C_nX_n + 0\ X_{n+1} + 0\ X_{n+2} + \dots + 0\ X_p$$

Consider the columns of coefficients of the X's as matrix vectors; then (3) can be written:

$$(4)\quad X_1\begin{bmatrix} A_{11} \\ A_{21} \\ \cdot \\ \cdot \\ A_{m1} \end{bmatrix} + X_2\begin{bmatrix} A_{12} \\ A_{22} \\ \cdot \\ \cdot \\ A_{m2} \end{bmatrix} + \dots X_n\begin{bmatrix} A_{1n} \\ A_{2n} \\ \cdot \\ \cdot \\ A_{mn} \end{bmatrix} + X_{n+1}\begin{bmatrix} 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{bmatrix} + \dots + X_p\begin{bmatrix} 0 \\ \cdot \\ \cdot \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} B_1 \\ \cdot \\ \cdot \\ \cdot \\ B_m \end{bmatrix}$$

Designating these vectors by P_k we can write relation (4) as:

$$(5)\quad X_1P_1 + X_2P_2 + \dots + X_pP_p = P_0$$

where P_0 is the vector of the coefficient appearing on the right side of the equations (3). The elements of P_0 should be non-negative. Note that the left side of the equations have been rearranged so that the last **m** of the P_k's are unit vectors and form the so-called "Unit Basis". They are the form.

$$(6)\quad \begin{matrix} 1 \\ 0 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{matrix},\ \begin{matrix} 0 \\ 1 \\ 0 \\ \cdot \\ \cdot \\ 0 \end{matrix},\ \begin{matrix} 0 \\ 0 \\ 1 \\ \cdot \\ \cdot \\ 0 \end{matrix},\ \dots\dots,\ \begin{matrix} 0 \\ \cdot \\ \cdot \\ \cdot \\ 0 \\ 1 \end{matrix}$$

Together they can be thought of as forming an identity matrix of order $\underline{m}$:

$$(7)\quad \begin{bmatrix} 1 & 0 & 0 & 0 \dots & 0 \\ 0 & 1 & 0 & 0 \dots & 0 \\ 0 & 0 & 1 & 0 \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 & 0 & 0 \dots & 1 \end{bmatrix}$$

If a complete set of unit vectors cannot be found in (5), e.g. if one of the relations in (2) is an equality and requires no new X_j $(j > n)$ to make it an equality, then the appropriate unit vectors are introduced to complete the unit basis and are referred to as artificial vectors.

Set the C_j value for these artificial vectors equal to -M, where M is an extremely large number, which will overpower all other numbers involved. For example, one might use 9×10^5 if all the other numbers were of the order of magnitude of less than 1000. The artificial vectors are a mathematical device only and should have no significance in the problem.

Now relation (1) becomes:

(8) $Z = C_1X_1 + \ldots + C_nX_n - MX_\ell - MX_{\ell+1} - \ldots - MX_{\ell+t}$

where $X_\ell, \ldots, X_{\ell+t}$ are the artificial variables introduced above. Hence it is intuitive that in order to maximize Z, $X_\ell = X_{\ell+1} = \ldots = X_{\ell+t} = 0$ and they have no net effect on relation (3). This can also be proved in theory.

Minimization Problem

Given a linear functional:

(1) $Z = C_1X_1 + C_2X_2 + \ldots C_nX_n$

which is to be minimized subject to the restraints:

$$A_{11}X_1 + A_{12}X_2 + \ldots + A_{1n}X_n \geq B_1$$
$$A_{21}X_1 + A_{22}X_2 + \ldots + A_{2n}X_n \geq B_2$$
(9) $$\cdots\cdots\cdots$$
$$\cdots\cdots\cdots$$
$$A_{m1}X_1 + A_{m2}X_2 + \ldots + A_{mn}X_n \geq B_m$$

Wherever an inequality appears a new X_i is subtracted from that relation to make it an equality. Relations (9) can thus be written:

$$
(10)\quad
\begin{aligned}
&A_{11}X_1 + \dots + A_{1n}X_n - X_{n+1} + 0 + \dots + 0 = B_1\\
&A_{21}X_1 + \dots + A_{2n}X_n + 0 - X_{n+2} + 0 + \dots + 0 = B_2\\
&\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\\
&\cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\ \cdot\\
&A_{m1}X_1 + \dots + A_{mn}X_n + 0 + \dots + 0 + X_p = B_m\\
&\qquad (p \leq m+n)
\end{aligned}
$$

$C_i = 0$ for all new X_i introduced to reduce an inequality to an equality.

As in the maximization problem, select a set of unit vectors, (the -1's introduced above do not constitute unit vectors), introducing sufficient artificial vectors to make a complete unit basis. The C_j's for the artificial vectors in this case are +M where M corresponds to the above. The artificial vectors have no significance in this case either.

Relation (1) now is

$$(11)\quad Z = C_1X_1 + \dots + C_nX_n + MX_{\ell} + MX_{\ell+1} + \dots + MX_{\ell+t}$$

and $X_{\ell} = X_{\ell+1} = \dots = X_{\ell+t} = 0$ by a similar argument as before.

A BRIEF DESCRIPTION OF THE SIMPLEX METHOD:

This is an iterative method which proceeds by introducing a non-basis vector into the basis and removing one of the old basis vectors, each time bringing the functional closer to its optimum value. A best solution is reached in a finite number of steps.

Define the elements of P_0 as $A_{10}, A_{20}, A_{30}, \dots, A_{m0}$. Then if P_1 thru P_m are the basis vectors, $X_1 = A_{10}$, $X_2 = A_{20}$, $X_3 = A_{30}, \dots, X_m = A_{m0}$ and all the other X's are 0 so

$$Z = C_1X_1 + C_2X_2 + \dots + C_mX_m = C_1A_{10} + C_2A_{20} + \dots + C_mA_{m0}$$

Then Z_j is defined as

$$Z_j = C_1A_{1j} + C_2A_{2j} + \dots + C_mA_{mj}$$

where P_j is a non-basis vector.

In the simplex method it turns out that when a non-basis P_k is introduced into the basis, the new value of the functional, Z', is given by:

$$Z' \equiv Z - X_k(Z_k - C_k)$$

Since the X_k is considered to be positive P_k must be chosen such that

$$Z_k - C_k < 0$$

to increase the value of Z in a maximization or $Z_k - C_k > 0$ to decrease the value of Z in a minimization.

When all the possible $Z_j - C_j$ are positive the maximum solution has been attained or when they are all negative the minimum solution. Thus the end tests for maximum or minimum problems are to require all positive or all negative $(Z_j - C_j)$'s respectively.

For purposes of using the same routine to compute the functional and the $Z_j - C_j$ values, the functional Z is called Z_0 with $C_0 = 0$ so that $Z = Z_0 - C_0$.

METHOD OF USING THIS PROGRAM:

(a) Scaling

To insure a minimum of roundoff error, the equations should be multiplied by a suitable power of ten, so that the elements of the matrix are of the same order of magnitude. For $C_j \neq 0$ such scaling will also change the values of the C_j's. E.g. scaling by rows alters nothing; scaling a column K by ρ is equivalent to the following change of variable.

$X_k = \rho X_k'$ so now $C_k' = \rho C_k$, $A_{1k}' = \rho A_{1k}$, ... $A_{mk}' = \rho A_{mk}$

Therefore; if the kth column is scaled by ρ then C_k of Z must also be scaled by ρ.

(b) Preparing the problem

Those vectors not appearing in the unit basis will be referred to as structural vectors. Number the structural vectors in order starting with 1, 2, ... n where the P_0 vector is as described in equation (5); number the basis vectors in order starting with n + 1, n + 2, ..., n + m.

(c) The coefficient tape

The routine performs its operations in a floating decimal representation so the P_k's and C_k's have to be punched in this form.

Each number is represented in the form $A \times 10^p$ where $1 > |A| \geq 1/10$, and $64 > p \geq -64$. In a single register of the memory the number A is placed in the 33 most significant binary digits $(a_0, a_1, \ldots a_{32})$ in the same way as an ordinary fraction is placed in the entire register. An accuracy of between 8 and 9 decimal digits is therefore achieved. The exponent **p** is stored as the integer **p** + 64 in the 7 least significant digits of the same register.

The only exception to the above rules is the number zero which cannot, of course, be represented as $A \times 10^p$ with $|A| \geq 1/10$. For this reason zero is handled in a special way. It is represented as a number A = 0 and p = -64. This representation happens to correspond exactly with the ordinary machine representation of zero.

All data are prepared by translating into signed floating decimal designation. The first signed set refers to the coefficient, A, and the second to the exponent, **p**.

E.g.
- 439.0 is punched as +439 +03
- .4390 is punched as +439 +00
- .00439 is punched as +439 -02
- -43.90 is punched as -439 +02

The coefficient tape is made by punching in order the elements of the structural vectors starting with P_0, P_1, ... P_n. The basis vectors are not punched.

Following the structural vectors, on the same tape, the C_j values are punched starting with $C_0(=0)$, C_1, C_2, ..., C_n, C_{n+1}, ..., C_{n+m} which are all the C_j values corresponding to all the vectors, both structural and basis.

There are no end marks required on the coefficient tape because the parameters, read in previous to this, specify the number of characters the program will read.

(d) The parameter tape

The parameter tape has the following form:

```
005+
00F 00jF
00F 00iF
0019+
00F 00tF
00F 00bF
24 999N
```

where

j = number of strucutral vectors not including P_0 or basis vectors.

i = number of basis vector (this is also the number of restraints)

t = frequency of the intermediate printout; t can be 0, 1, 2, ... (See f below).

b = 0 for a minimization problem
  = 1 for a maximization problem.

The capacity of the routine is defined by :

$$4(i + j + 1) + ij \leq 712$$

(e) Reading in the program

The master tape is read in until the first black switch stop and then the parameter tape is read in. The rest of the master tape is then read in and will stop, after which the coefficient tape is read in.

(f) The punch-out

For intermediate results the X's and $(Z_j - C_j)$'s will be punched out every t iterations as specified on the parameter tape. The number of the vector in sexadecimal notation will be punched followed by the X's for the current basis and by the $(Z_j - C_j)$'s for the current non-basis vectors, in floating decimal form.

If the problem converges, the final punch-out will consist first of the numbers of the final basis vectors in sexadecimal notation followed by the respective X values in floating decimal form. Then the final set of non-basis vectors and their respective $Z_j - C_j$ values.

If the problem diverges, the machine will stop on an FF order.

RUNNING TIME:

Running time is difficult to predict from i and j above. Following are some samples (all with only final printout). In general, requiring the intermediate print-out of every iteration doubles machine time.

Maximization

i	j	time
19	11	8 min
18	25	20 min
*18	25	15 min

Minimization

i	j	time
8	18	5 min
15	31	14 min

Problems utilizing nearly the full capacity have almost always fallen in the range of 10 to 31 minutes (with only final printout).

EXAMPLE:

Consider the following problem:

Maximize:

$$Z = -5X_1 + 38X_2 + 4X_3 + 8X_4$$

subject to the restraints:

$$X_i \geq 0$$

---

* Same coefficients as line above but different C_j.

$$\begin{aligned} (a)\quad & X_2 \qquad\qquad\quad + 2X_4 = 10 \\ & X_1 - 5X_2 - 7X_3 + 2X_4 = 14 \\ & \qquad X_2 + 6X_3 - 4X_4 \leq 4 \end{aligned}$$

the last inequality would be written as:

$$X_2 + 6X_3 - 4X_4 + X_5 = 4$$

but Z will remain unchanged because $C_5 = 0$.

One would then rearrange the subscripts so that the unit vectors would appear on the right. Thus X_2 becomes X_1' or $X_2 \rightarrow X_1'$, $X_3 \rightarrow X_2'$, $X_4 \rightarrow X_3'$, $X_1 \rightarrow X_5'$ $X_5 \rightarrow X_6'$

producing

$$Z = 38X_1' + 4X_2' + 8X_3' - 5X_5' + 0X_6'$$

and

$$\begin{aligned} (b)\quad & X_1' \qquad\qquad + 2X_3' \qquad\qquad\qquad = 10 \\ & -5X_1' - 7X_2' + 2X_3' + X_5' \qquad = 14 \\ & X_1' + 6X_2' - 4X_3' \qquad\quad + X_6' = 4 \end{aligned}$$

which can be written:

$$(c)\quad \begin{bmatrix} 1 & 0 & 2 & 0 & 0 \\ -5 & -7 & 2 & 1 & 0 \\ 1 & 6 & -4 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1' \\ X_2' \\ X_3' \\ X_5' \\ X_6' \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \\ \\ 4 \end{bmatrix}$$

However, this does <u>not</u> give a complete unit basis on the right; the desired form is:

$$(d)\quad \begin{bmatrix} 1 & 0 & 2 & 1 & 0 & 0 \\ -5 & -7 & 2 & 0 & 1 & 0 \\ 1 & 6 & -4 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_1' \\ X_2' \\ X_3' \\ X_4' \\ X_5' \\ X_6' \end{bmatrix} = \begin{bmatrix} 10 \\ 14 \\ \\ 4 \end{bmatrix}$$

which can be achieved by making $C_4' = -100$ (or an even larger negative number).

Then

$$Z' = 38X_1' + 4X_2' + 8X_3' - 100X_4' - 5X_5'$$

It would appear intuitively obvious that in order to make Z' maximum, $X_4 = 0$ necessarily, because $X_4 \geq 0$ (this argument does not apply to $C_5' = -5$ because it is of the same order of magnitude as C_1', C_2' and C_3').

Then if $X_4' = 0$, relations (c) and (d) are "equivalent" in the sense that when the matrices are multiplied out, X_4' appears only in the top row, thus:

$$X_1' + 2X_3 + X_4' = 10$$

but by making $C_4' = -100$, $X_4' = 0$ is forced which is equivalent to:

$$X_1' + 2X_3' = 10$$

In order to punch the data and parameter tapes we define:

$$P_0 = \begin{bmatrix} 10 \\ 14 \\ 4 \end{bmatrix}$$

and the structural vectors are:

$$P_1 = \begin{bmatrix} 1 \\ -5 \\ 1 \end{bmatrix}, \quad P_2 = \begin{bmatrix} 0 \\ -7 \\ 6 \end{bmatrix}, \quad P_3 = \begin{bmatrix} 2 \\ 2 \\ -4 \end{bmatrix}$$

and P_4, P_5, and P_6 are the unit vectors

$$\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \quad \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \quad \text{and} \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

respectively.

The data tape is as follows:

+1 +02		
+14 +02	P_0	
+ 4 +01		
+ 1 +01		
- 5 +01	P_1	
+ 1 +01		
+ 0 +00		
- 7 +01	P_2	
+ 6 +01		
+ 2 +01		
+ 2 +01	P_3	
- 4 +01		
+ 0 +00	C_0	
+38 +02		
+ 4 +01		
+ 8 +01		
- 1 +03	C_1' then C_6'	
- 5 +01		
+ 0 +00		

Note that P_4 thru P_6 are not punched,as these are generated by the program.

The parameter tape is:

```
005+
00F 003F        (3 structural vectors)
00F 003F        (3 basis vectors)
0019+
00F 001F        (print after each iteration)
00F 001F        (Maximize Z)
24999N
```

The output has the form

004	+100000000	+02	X_4	1st iteration
005	+140000000	+02	X_5	
006	+400000000	+01	X_6	
000	-107000000	+04	Z_0	
001	-113000000	+03	$Z_1 - C_1$	
002	+310000000	+02	$Z_2 - C_2$	
003	-218000000	+03	$Z_3 - C_3$	
003	+500000000	+01	X_3	2nd iteration
005	+400000001	+01	X_5	
006	+240000000	+02	X_6	
000	+200000000	+02	Z_0	
001	-400000003	+01	$Z_1 - C_1$	
002	+310000000	+02	$Z_2 - C_2$	
004	+109000000	+03	$Z_4 - C_4$	
003	+100000000	+01	X_3	3rd iteration and final solution
005	+520000000	+02	X_5	
001	+800000000	+01	X_1	
000	+520000002	+02	Z_0	
006	+133333335	+01	$Z_6 - C_6$	
002	+390000000	+02	$Z_2 - C_2$	
004	+111666667	+03	$Z_4 - C_4$	

As can be seen the problem required 3 iterations. The final solution contains $X_1 = 8$, $X_3 = 1$, and $X_5 = 52$ with the maximum $Z = 52$.

DATE May 19, 1960

CODED BY L. Isaacson

REVISED BY R. Hacker

APPROVED BY J. Snyder

nj

PARAMETERS

0	
1	Temporary storage for X 1 and A 1 (for 0–2)
2	
3	Location of floating accumulator
4	Location of Library Routine A 1
5	Number of j vectors, $j = 1, \ldots, m$
6	Number of i vectors, $i = m + 1, \ldots, p$
7	Number of coefficients to be input
8	Location of temporary storage for P_k
9	Location of C_j $(j = 0, 1, \ldots, m)$
10	Location of C_i $(i = m + 1, \ldots, p)$
11	Location of $Z_j - C_j$
12	00 F
	00 1S5
13	Location of X_{kj} $(j = 0, \ldots, m)$
14	j vector designations
15	i vector designations
16	00 311F
	00 311F
17	00 S6
	00 S6
18	00 F
	00 311F
19	Print parameter
20	Maximization of minimization parameter

INTERLUDE TO SET PARAMETERS

00 21K

0	50 12F
	75 6F
1	S5 F
	L4 12F
2	L4 6F
	40 7F
3	L0 12F
	L0 6F
4	L4 18F
	40 9F

```
5      L4 12F
       40 10F
6      L4 6F
       40 14F
7      L4 12F
       40 15F
8      L4 6F
       40 11F
9      L4 12F
       40 13F
10     L4 12F
       40 8F
11     50 12L
       26 999F
12     00 F
       00 L
13     00 F
       26 L
       26 1N
```

INTERLUDE TO GENERATE VECTOR INTEGERS

```
0      L1 6F
       L0 5F
1      40 8L
       41 SF
2      F5 SF
       40 1SF
3      F5 8L
       40 8L
4      36 9L
       L5 2L
5      L4 7L
       40 2L
6      26 2L
       00 F
7      00 1F
       00 1F
```

8	00 F
	00 F
9	50 10L
	26 999F
10	00 F
	00 L
11	00 F
	26 L
	26 1N

Note: See "An Introduction to Linear Programming", Charnes, Cooper and Henderson.

K' - column integer

r' - row integer

$P_{K'}$ - column vector to be moved into basis

LOCATION	ORDER		NOTES PAGE 1	M 15
0	41 11F			
	L5 16F			
1	42 4L			
	50 1L		Enter Routine A 1	
2	26 S4			
	0K SN			
3	1K S6			
	8K F			
4	8S 3F			
	15 F			
5	17 SK		$Z_j = \Sigma x_{ij} C_i$	
	84 3F			
6	8S 3F			
	12 4L			
7	85 3F			
	00 S9		$Z_j - C_j$	
8	0S SS			
	8J 69L			
9	03 3L			
	2K S5			
10	8K F			
	8S 4F			
11	21 1SS		Test $Z_j - C_j$ (+) reject, (-) store new	
	80 4F		$Z_j - C_j$	
12	83 14L			
	21 1SS			
13	8S 4F			
	8J 71L		Store K'	
14	8J 73L		Step K'	
	23 11L			

LOCATION	ORDER	NOTES
15	85 4F	**Is program over?**
	82 68L	(+) yes, (-) no
16	8J 7SL	
	3K S6	
17	8K F	
	3N F	Is $x_{iK} = 0$? (+) yes, (-) no
18	83 25L	
	31 F	Is $x_{iK} < 0$? (+) yes, (-) no
19	83 25L	
	83 25L	Waste
20	31 311F	$-\theta_i$
	36 F	
21	8S 3F	Is $\theta - \theta_i \lessgtr 0$? (+) store new θ, (-)
	84 120L	reject
22	82 23L	
	81 3F	
23	83 25L	
	81 3F	
24	8S 120L	
	8J 81L	Store r'
25	8J 73L	Step r'
	33 17L	
26	85 120L	
	80 121L	Has problem diverged? (+) yes, (-) no
27	82 65L	
	82 65L	Waste
28	4K S6	
	45 F	Store $P_{K'}$
29	4S S8	
	42 28L	
30	8J 85L	
	5K S6	
31	8K F	Clear column K' to zeros
	5S F	
32	53 31L	
	8K 1F	

M 15

LOCATION	ORDER	NOTES	PAGE 3
33	82 33L	Waste	Store 1 at (r', K')
	8S F		
34	6K SN		
	85 F		
35	86 20F	x'_{Kj}	
	6S SJ		
36	8J 93L		
	62 34L		
37	3K SN		
	4K S6		
38	41 S8	$x'_{ij} = x_{ij} - \left(\frac{x_{rj}}{x_{rK}}\right) x_{iK}$	
	37 SJ		
39	44 F		
	4S F		
40	43 38L		
	8J 95L		
41	32 37L		
	5K SN		
42	55 SJ		
	8S F	Restore x'_{Kj}	
43	8J 97L		
	53 42L		
44	8J 99L		
	2K S6		
45	8F 1F		
	8J 56L	Print X's	
46	23 311F		
	89 9F		
47	22 45L		
	22 45L	Waste	
48	8F 1F		
	7K SN		
49	8J 60L	Print $Z_j - C_j$	
	75 SS		
50	89 9F		
	73 49L		

LOCATION	ORDER	NOTES PAGE 4 M 15
51	73 49L	Waste
	8J 64L	
52	OF F	
	50 52L	
53	26 S4	
	4K S7	
54	88 F	Input
	4S 311F	
55	43 54L	
	8J L	
56	92 131F	
	L5 SL	
57	00 28F	
	82 12F	Print λ designation
58	92 963F	
	F5 56L	
59	40 56L	
	26 29S4	
60	92 131F	
	L5 SF	
61	00 28F	
	82 12F	Print $Z_j - C_j$ designations
62	92 963F	
	F5 60L	
63	40 60L	
	26 29S4	
64	L1 19F	
	40 12F	Reset print counter
65	26 83L	
	8J 66L	
66	FF F	
	00 52L	
67	L5 66L	
	42 51L	Prepare to stop program
68	26 116L	
	8J 67L	

LOCATION	ORDER	NOTES
69	L5 4L	
	L4 6F	From 8L
70	40 4L	
	26 29S4	
71	F5 11F	
	40 13F	Store K'
72	26 29S4	
	00 F	
73	F5 11F	
	40 11F	Step K' of r'
74	26 29S4	
	00 F	
75	50 13F	
	75 6F	
76	L5 16F	
	S4 F	
77	42 17L	From 16L
	42 18L	
78	42 20L	
	42 28L	
79	42 31L	
	41 11F	
80	26 114L	
	00 F	
81	L5 11F	
	40 18F	Store r'
82	26 29S4	
	00 F	
83	41 S3	
	50 83L	
84	26 S4	
	82 16L	
85	L5 28L	
	L4 18L	
86	42 33L	
	42 89L	

LOCATION	ORDER	NOTES PAGE 6
87	L5 16F	
	L4 18F	From 30L
88	42 34L	
	42 42L	
89	22 89L	
	L5 F	Store X_{rK}
90	40 20F	
	L5 16F	
91	42 39L	
	46 39L	
92	26 29S4	
	00 F	
93	L5 34L	
	L4 6F	
94	40 34L	From 36L
	26 29S4	
95	L5 39L	
	L4 17F	From 40L
96	40 39L	
	26 29S4	
97	L5 42L	
	L4 6F	From 43L
98	40 42L	
	26 29S4	
99	L5 9F	
	L4 13F	
100	42 103L	
	42 105L	
101	L5 10F	
	L4 18F	
102	42 104L	
	42 106L	
103	22 103L	
	50 F	Interchange i and j designation; also c_i and c_j
104	22 104L	
	L5 F	

M 15

LOCATION	ORDER	NOTES	PAGE 7
105	22 105L		
	40 F		
106	S5 F		
	40 F		
107	L1 53L		
	40 53L	Binary switch	
108	36 L		
	L5 14F		
109	L4 13F		
	42 103L		
110	42 105L		
	L5 15F		
111	L4 18F		
	42 104L		
112	42 106L		
	L5 121L		
113	40 120L		
	22 103L		
114	26 29S4	For non-zero print is modified	
	00 F		
115	36 116L		
	26 29S4		
116	L5 14F		
	42 60L		
117	L5 15F		
	42 56L		
118	41 S3		
	50 118L		
119	26 S4		
	82 44L		
120	40 F		
	00 127F		
121	40 F		
	00 127F		

The following interlude sets the print and also maximization or minimization.

LOCATION	ORDER	NOTES
	00 143K	
0	L1 19F	
	32 2L	
1	40 12F	
	L5 8L	
2	40 135F	
	L1 20F	
3	36 6L	
	L5 9L	
4	40 32F	
	L5 10L	
5	40 33F	
	26 6L	
6	50 7L	
	26 999F	
7	00 F	
	00 L	
8	F5 12F	
	40 12F	
9	25 1SS	
	80 4F	
10	83 35F	
	25 1SS	
11	00 F	
	26 L	
12	26 1N	
	Routine A 1	
	20 73N	