INSTRUCTIONS FOR USING EXPERIMENTAL
0-1 INTEGER LINEAR PROGRAMMING CODE RIP23J

A. M. Geoffrion
A. B. Nelson

November 1967
Numerous requests have been received for copies of the experimental code used to obtain the computational experience reported in Refs. 2 and 3. It should be recognized that this is not a production code. It was developed to test the usefulness of certain innovations applied to a simple Balasian algorithm. The central concern was the rate of increase of solution time as a function of the number of variables, rather than how to achieve the smallest possible execution time for particular problems. For this reason, the simplest possible Balasian algorithm was used as the starting point, and concessions were freely made to programming expediency (e.g., no machine language). It would not be difficult to reduce execution times substantially by reprogramming and introducing some of the more sophisticated tests already available in the literature.

We discuss input in Sec. 1; output in Sec. 2; and give an example in Appendix A, and a program listing in Appendix B. For an outline of the working details of the algorithm, sec [1] and [2]. Familiarity with these papers is presumed here.

The program solves integer linear programs of the form

\[
(P) \quad \text{Minimize } cx \text{ subject to } b + Ax \geq 0 \\
\quad x_j = 0 \text{ or } 1
\]

*Any views expressed in this paper are those of the author. They should not be interpreted as reflecting the views of The RAND Corporation or the official opinion or policy of any of its governmental or private research sponsors. Papers are reproduced by The RAND Corporation as a courtesy to members of its staff.*
where $c$ and $x$ are $n$-vectors, $b$ is an $m$-vector, and $A$ is $m \times n$. Any bounded integer linear program can be written in this form, using elementary manipulations if necessary.
1. INPUT

The following parameter and data cards appear for each problem to be run:

(a) Parameter card
(b) S-card(s)
(c) C-card(s)
(d) B-card(s)
(e) A-card(s)
(f) Blank card.

Problems can be stacked by repetition of cards a through f.

Parameter Card

The input parameters are:

- **M**: The number of constraints
- **N**: The number of variables
- **L**: The number of variables in the initial partial solution (L must correspond to the number of entries on the S-card). If $L = 0$, the initial partial solution is empty. If $L < 0$, the initial partial solution consists of all variables fixed at the value 0.
- **SC**: Punch 0 if no imbedded linear program is desired (the algorithm then reduces to a simple Balasian algorithm), and 1 if the imbedded linear program is to be used.
- **KENUM**: When intermediate output is used ($N_{out} = 0$), the fraction of all $2^n$ possible solutions that have been implicitly enumerated is printed out every KENUM times that backtracking occurs. KENUM = 20 is reasonable.
- **ZBAR**: If an upper bound $\bar{z}$ on the optimal value of the objective function of (P) is known, put $ZBAR = \bar{z} - \text{lcd} + 0.0001$, where \text{lcd} is the least common denominator of the cost coefficients $c_j$ (we assume that $\bar{z}$ is a multiple of \text{lcd}). Hence, if all $c_j$ (and $\bar{z}$) are integer, put $ZBAR = \bar{z} - .9999$. The effect will be that the program looks only for feasible solutions with value $< ZBAR$. If no upper bound is known, put $ZBAR = 0$. See Remark 2 below.
ISCMAX  The maximum number of composite constraints that will be carried. ISCMAX = 4 is reasonable.

ISCFR  The frequency with which the imbedded linear program is used. ISCFR = 0 means that it will never be used; ISCFR = j, j a positive integer, means that it will be used every jth time. ISCFR = 1 has proven effective, but frequently a value of 8 or so is even better.

MAXC  If equal to 0, nothing will happen. If equal to 1, all signs on the C- and A-Cards will be reversed automatically when these cards are read in. This is purely a convenience for manuscripting and key-punching for problems with a preponderance of minus signs in C and A.

MAXT  Terminates the calculations after MAXT seconds.

N0P  If equal 1, intermediate output will be suppressed; if equal 0, intermediate output will appear. Normally N0P will be set at 1.

ZKBAR  Put equal to led (see ZBAR) minus 1. Thus, if all c_i are integer, put ZKBAR = 0. The effect is that the program looks only for feasible solutions with value at least (ZKBAR + .99999) less than the best feasible solution currently known; this doesn't exclude any optimal solutions. (A solution within A of the optimum can be found if desired by increasing the above value of ZKBAR by A.)

H1,H2  Arbitrary problem identifiers.

Remark 1: The program is currently dimensioned to use 32,000 words of core in such a way that the following limits must be observed:

\[ M + ISCMAX \leq 50 \]

\[ N \leq 90. \]

Remark 2: If any c_j are negative (after MAXC has changed the input signs, if it has value 1), the program internally makes a trivial change of variables to make such c_j nonnegative: replace \( x_j \) by \( y_j = (1 - x_j) \) if \( c_j < 0 \). The problem is solved in terms of the new variables, and the reverse transformation is made at final output in order to recover the solution to the original problem. ZBAR must be set at a value corresponding to the transformed problem when it is desired to use a
known upper bound; hence, when \( c_j < 0 \) for \( j \in J \), put \( ZBAR = \bar{z} - \text{lcl} + 0.0001 + \sum_{j} |c_j| \).

The fields and formats of the parameter card are as follows:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Column</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M )</td>
<td>1-3</td>
<td>Integer</td>
</tr>
<tr>
<td>( N )</td>
<td>4-6</td>
<td>&quot;</td>
</tr>
<tr>
<td>( L )</td>
<td>7-9</td>
<td>&quot;</td>
</tr>
<tr>
<td>( SC )</td>
<td>10-12</td>
<td>&quot;</td>
</tr>
<tr>
<td>( KENUM )</td>
<td>13-17</td>
<td>&quot;</td>
</tr>
<tr>
<td>( ZBAR )</td>
<td>18-23</td>
<td>E</td>
</tr>
<tr>
<td>( ISCMAX )</td>
<td>24-26</td>
<td>Integer</td>
</tr>
<tr>
<td>( ISCFR )</td>
<td>27-29</td>
<td>&quot;</td>
</tr>
<tr>
<td>( MAXC )</td>
<td>30-32</td>
<td>&quot;</td>
</tr>
<tr>
<td>( MAXT )</td>
<td>33-37</td>
<td>&quot;</td>
</tr>
<tr>
<td>( N@P )</td>
<td>38-40</td>
<td>&quot;</td>
</tr>
<tr>
<td>( ZKBAR )</td>
<td>41-46</td>
<td>E</td>
</tr>
<tr>
<td>( H1 )</td>
<td>47-52</td>
<td>Hollerith</td>
</tr>
<tr>
<td>( H2 )</td>
<td>53-58</td>
<td>&quot;</td>
</tr>
</tbody>
</table>

**S-Card(s)**

The algorithm can start with any initial partial solution (see [1]). When the initial partial solution is desired to be nonempty \( (L > 0) \), if \( x_j \) is to be fixed at the value one (zero) then "j" ("-j") is entered on the S-card, followed by "B" when an underline is desired. The S-card is divided into 12 fields of 5 columns each: 1-5, 6-10, ..., 66-70. Only the first four columns of each field are to be used except when underlines are desired, in which case "B" must appear in the fifth column of the field.

The special instruction given above in Remark 2 for \( ZBAR \), when a change of variables is made, also applies here. That is, the sign of \( \pm j \) or \( \pm jB \) must be changed when \( c_j < 0 \).

**C-Card(s)**

The values of the \( c_j \) must be entered in order (negative values are permissible, as noted above). Each card has six fields of eleven columns read in E-format. The fields are separated by an unread column so that the values of the \( c_j \) are in columns 1-11, 13-23, ..., 61-71.
**B-Card(s)**

The values of the $b_{ij}$ must be entered in order. The format is exactly the same as for the C-cards.

**A-Card(s)**

Only nonzero $a_{ij}$ need be entered, and they may be entered in any order. Each value is identified by its row and column. There are four or fewer entries on each of the "A" cards. Each entry has a seventeen column field.

<table>
<thead>
<tr>
<th>Columns</th>
<th>Format</th>
</tr>
</thead>
<tbody>
<tr>
<td>Row</td>
<td>1-3</td>
</tr>
<tr>
<td>Column</td>
<td>4-6</td>
</tr>
<tr>
<td>Value</td>
<td>7-17</td>
</tr>
</tbody>
</table>

The fields are separated by an unread column so that the matrix subscripts and values of the $a_{ij}$ are in columns 1-17, 19-35, 37-53, 55-71.
2. OUTPUT

The preliminary, intermediate, and final outputs are as follows.

The parameter, "S", "C", and "B" cards are printed in that order (six values to a line for the "C" and "B" cards). Then the complete A matrix is printed (with zeros), row by row. If MAXC = 1, the sign reversals in "C" and "A" will be seen to have occurred. If a change of variables was made internally, the new c, b, and A are printed out (if no change of variables was necessary, the identical c, b, and A are printed out again anyway).

If NDF = 0, intermediate output is produced to reveal the course of the calculations - each feasible solution found, each new composite constraint, data concerning each imbedded linear program, and a summary of progress to date after each KENUM "backtrackings." Since this information is likely to be of little incremental value to the user over the final output information, no detailed explanation is given here.

The final output gives the problem designation; the message "implicit enumeration complete" or "time exceeded" according as termination did or did not occur within MAXT seconds; the total execution time in seconds; the solution (obj. fc. value and a list of which variables equal 1) both before and after the variable change (if no variable change occurred, these solutions are identical); and some statistical information on the course of the algorithm, such as the number of feasible solutions found, the number of times the imbedded linear program was solved, the number of iterations, and the time at which the last feasible solution was found. In the event that no feasible solutions were found, this is indicated by the zeros in the solution after variable change and the statistic "no. feasible solutions 0." In the event that the time limit was exceeded, the final output is preceded by a brief report giving the proportion of all $2^n$ possible solutions that have been accounted for and the final "state" vector [1], with "B" signifying an underline. All the information needed to restart the calculations is available: make the S-card correspond to the final state vector (set L accordingly), and put ZBAR equal to LEAST Z AFTER VARIABLE CHANGE - lcf + 0.0001.
Appendix A

EXAMPLE

We shall illustrate the above by solving Petersen's fifth example [4].

For this problem, $M = 10$ and $N = 28$. We shall take $L = 28$, $SC = 1$, $KENUM = 20$, $ZBAR = 0$ (since we will not bother to determine a bound on the objective function), $ISCMAX = 4$, $ISCFR = 1$, $MAXC = 1$ (since we wish to avoid keypunching all the minus signs for $c$ and $A$), $MAXT = 60$, $NQP = 1$, $ZKBAR = 4$ (since the least common denominator of the $c_j$ is 5), and $HI = PETE 5$.

The S-card will contain the numbers 1, 2, ..., 28 (we have elected an initial partial solution with all variables fixed at the value 1).

The output is reproduced below.
|     | 1     | 2     | 3     | 4     | 5     | 6     | 7     | 8     | 9     | 10    | 11    | 12    | 13    | 14    | 15    | 16    | 17    | 18    | 19    | 20    | 21    | 22    | 23    | 24    | 25    | 26    | 27    | 28    |
|-----|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|-------|
|     | -1.0000000 | 02  | -2.2000000 | 02  | -9.0000000 | 01  | -4.0000000 | 02  | -3.0000000 | 02  | -6.0000000 | 02  | -4.0000000 | 02  | -2.0000000 | 02  | -1.0000000 | 02  | -1.0000000 | 02  | -1.0000000 | 02  | -1.0000000 | 02  | -1.0000000 | 02  | -1.0000000 | 02  | -1.0000000 | 02  |
|     | 9.0000000 | 02  | 1.2100000 | 03  | 2.7200000 | 02  | 4.6200000 | 02  | 5.3200000 | 02  | 5.7200000 | 02  | 4.9000000 | 02  | -2.4000000 | 02  | -8.0000000 | 01  | -1.2000000 | 01  | -1.4000000 | 01  | -1.6000000 | 01  | -1.8000000 | 01  | -2.0000000 | 01  | -2.2000000 | 01  |
|     | -8.0000000 | 00  | -2.4000000 | 01  | -1.8000000 | 02  | -2.7000000 | 02  | -2.0000000 | 01  | -5.0000000 | 01  | -4.0000000 | 01  | -6.0000000 | 00  | -8.0000000 | 00  | -1.5000000 | 01  | -1.9000000 | 01  | -2.3000000 | 01  | -1.7000000 | 01  | -1.6000000 | 01  | -1.5000000 | 01  |
|     | -1.0000000 | 00  | -0.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 01  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  | -2.0000000 | 00  |
|     | -2.0000000 | 00  | -4.0000000 | 00  | -6.0000000 | 00  | -8.0000000 | 00  | -10.0000000 | 00  | -12.0000000 | 00  | -14.0000000 | 00  | -16.0000000 | 00  | -18.0000000 | 00  | -20.0000000 | 00  | -22.0000000 | 00  | -24.0000000 | 00  | -26.0000000 | 00  | -28.0000000 | 00  |

---

**Note:** The table represents numerical data in a structured format. Each row and column represents specific values, possibly related to a mathematical or scientific context. The data appears to be organized in a grid, with values increasing or decreasing in a particular pattern.
<table>
<thead>
<tr>
<th>2.000000 F 01</th>
<th>1.500000 F 01</th>
<th>6.000000 F 00</th>
<th>2.000000 F 01</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.</td>
<td>0.</td>
<td>1.000000 F 00</td>
<td>1.000000 F 01</td>
</tr>
<tr>
<td>6.000000 F 00</td>
<td>0.</td>
<td>6.000000 F 00</td>
<td>1.200000 F 01</td>
</tr>
<tr>
<td>7.000000 F 01</td>
<td>1.000000 F 01</td>
<td>0.</td>
<td>0.</td>
</tr>
<tr>
<td>0.</td>
<td>3.000000 F 01</td>
<td>1.000000 F 01</td>
<td>0.</td>
</tr>
<tr>
<td>1.000000 F 01</td>
<td>5.000000 F 00</td>
<td>0.</td>
<td>1.000000 F 01</td>
</tr>
<tr>
<td>3.000000 E 00</td>
<td>4.000000 E 00</td>
<td>5.000000 F 00</td>
<td>2.000000 F 01</td>
</tr>
<tr>
<td>6.000000 E 00</td>
<td>1.200000 E 01</td>
<td>1.000000 E 01</td>
<td>1.800000 E 01</td>
</tr>
<tr>
<td>1.200000 F 01</td>
<td>1.000000 E 02</td>
<td>2.000000 F 01</td>
<td>5.000000 F 00</td>
</tr>
<tr>
<td>1.000000 E 00</td>
<td>2.000000 E 01</td>
<td>5.000000 F 01</td>
<td>3.000000 F 01</td>
</tr>
<tr>
<td>2.000000 E 01</td>
<td>1.000000 E 01</td>
<td>1.000000 E 01</td>
<td>2.000000 E 01</td>
</tr>
<tr>
<td>3.000000 E 00</td>
<td>6.000000 E 00</td>
<td>9.000000 E 00</td>
<td>3.000000 E 01</td>
</tr>
<tr>
<td>1.200000 E 01</td>
<td>1.200000 E 01</td>
<td>1.000000 E 01</td>
<td>3.000000 E 01</td>
</tr>
<tr>
<td>1.800000 E 01</td>
<td>1.100000 E 02</td>
<td>7.000000 E 01</td>
<td>1.500000 F 01</td>
</tr>
<tr>
<td>2.000000 E 00</td>
<td>4.000000 E 01</td>
<td>6.000000 E 01</td>
<td>5.000000 F 01</td>
</tr>
<tr>
<td>2.500000 E 01</td>
<td>1.500000 E 01</td>
<td>1.000000 F 01</td>
<td>2.800000 E 01</td>
</tr>
<tr>
<td>3.000000 E 00</td>
<td>6.000000 E 00</td>
<td>9.000000 E 00</td>
<td>3.500000 F 01</td>
</tr>
<tr>
<td>1.600000 E 01</td>
<td>1.500000 E 01</td>
<td>1.000000 E 01</td>
<td>3.000000 E 01</td>
</tr>
<tr>
<td>1.800000 E 01</td>
<td>1.200000 E 01</td>
<td>2.000000 E 01</td>
<td>2.000000 E 01</td>
</tr>
<tr>
<td>3.000000 E 00</td>
<td>5.000000 E 01</td>
<td>6.000000 E 01</td>
<td>5.000000 E 01</td>
</tr>
<tr>
<td>2.500000 E 01</td>
<td>1.500000 E 01</td>
<td>1.000000 E 01</td>
<td>2.800000 E 01</td>
</tr>
<tr>
<td>PETE 5</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1)PLICIT ENUMERATION COMPLETE TOTAL TIME 4.371</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEAST Z AFTER VARIABLE CHANGE = 3.0950000E+03</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0 0 0 4 5 6 7 8 0 0 0 0 0 0 24 0 0 0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LEAST Z BEFORE VARIABLE CHANGE = -1.2400000E+04</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1 2 3 0 0 0 0 0 9 0 0 0 0 14 15</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>16 17 18 19 20 21 22 23 0 25 26 27 28</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO. FEASIBLE SOLUTIONS 24</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2S GE ZBAR 5 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>CONSTRAINT INFEASIBLE 15 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUGMENTATION IMPOSSIBLE 2 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>AUGMENTATION POSSIBLE 19 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>INTEGER DUALS 0 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP FATHOMED 2 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LP CALLED 21 TIMES</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>NO. ITERATIONS 9%</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LAST FEASIBLE SOLUTION AT 4.132 SECONDS</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Appendix B

LISTING OF RIP23J
SUBFC RIP23J

DIMENSION A(50,90),JF(50,90)
DIMENSION B(100),C(100),BS(100),S(100),SB(100),NS(100),NF(100)
DIMENSION TEMP(4),JTEMP(4),ATEMP(4),SMAX(100),SMAXB(100),T(100)
DIMENSION CS(100),M(100)
DIMENSION XL(90),DL(90),E(90,90)
DIMENSION JM(100),X(100),Y(100),PE(100),KD(16)

INTEGER S,SMAX,SC,T
COMMON /BLS/MS(90),ZBAR
DATA BCIB/6HB /
DATA BLANK/6H /
100 DO 110 I=1,90
   M(I)=0.0
   B(I)=0.0
   C(I)=0.0
   BS(I)=0.0
   S(I)=0
   SB(I)=BLANK
   NS(I)=0
   NF(I)=0
   SMAX(I)=0
   T(I)=0
   DO 110 J=1,50
      A(I,J)=0.0
      JF(J,I)=0.0
110 CONTINUE

I=0
NCDN=0
NERD=0
NAGU=0
NOPT=0
NID=0
MAP=0
MLPF=0
NSIMP=0
NFATH=0
MENU=0
NICE=0
ITB=0
IPOST=1
INS=5

C READ A NEW SET OF DATA
C PARAMETER CARD FIRST
C 'S' CARD THIRD
C 'C', 'B', 'A' MATRICES FOLLOW 'S'
C MINIMIZE SUM C(I,J)*X(I,J)
C CONSTRAINTS ARE B(I)+SUM A(I,J)*X(I,J) GE ZERO
   READ 9000,M,N,L,SC,KENUM,ZBAR,ISCMAX,ISCFR,MAXC,MAXT,
   NOP,ZKBAR,H1,H2
9000 FORMAT (I13,I5,E6.0,313,15,I3,E6.0,2A6)
PRINT 9993
9993 FORMAT (I13,15,1X,E11.4,313,15,I3,E11.4,1X,2A6)
   PRINT 9001,M,N,L,SC,KENUM,ZBAR,ISCMAX,ISCFR,MAXC,MAXT,
   NOP,ZKBAR,H1,H2
9001 FORMAT (I13,15,1X,E11.4,313,15,I3,E11.4,1X,2A6)
   IF (INS.EQ.0) INS=9999
   IF (MAXC.EQ.0) MAXT=999999
   MAXT=1000*MAXT

-16-
MO=M  
M1=M0+1  
JSCFR=JSCFR  
ZKBAR=ZKBAR+.99999  
PRINT 9010,M,N  
PRINT 9010,M,N  
9010 FORMAT (3HOM,,13,2X,2HNN,,13)  
PRINT 9992  
9991 FORMAT (1H0)  
9992 FORMAT (1H0)  
9993 FORMAT (1H1)  
L1=L  
IF (L.LE.0) L1=0  
READ 9100,((S(K),SB(K)),K=1,L1)  
9100 FORMAT (14(14,A1))  
IF (L.LE.0) GO TO 130  
L=N  
DO 120 K=1,N  
120 S(K)=K  
130 CONTINUE  
READ 9200,(C(J),J=1,N)  
9200 FORMAT (6(E11.0,1X))  
**************  
IF (MAXC.EQ.0) GO TO 141  
DO 140 J=1,N  
140 C(J)=-C(J)  
141 CONTINUE  
READ 9200,(B(I),I=1,M)  
200 READ 9400,((ITEMP(K),JTEMP(K),ATEMP(K)),K=1,4)  
9400 FORMAT (4(2I4,T11.0,1X))  
END=0.0  
DO 250 K=1,4  
K=ITEMP(K)  
KJ=JTEMP(K)  
IF (K.IEQ.0) GO TO 250  
IF (K,J.EQ.0) GO TO 250  
KJF=NF(KI)+1  
NF(KI)=KJF  
IF (K.IEQ.0) ATEMP(K)=-ATEMP(K)  
A(KI,KJ)=ATEMP(K)  
END=1.0  
250 CONTINUE  
IF (END.NE.0) GO TO 200  
PRINT 9992  
PRINT 9500,((S(K),SB(K)),K=1,L)  
9500 FORMAT (14(13X,14,A1))  
PRINT 9992  
PRINT 9600,(C(J),J=1,N)  
PRINT 9992  
PRINT 9600,(B(I),I=1,M)  
PRINT 9992  
DO 251 I=1,M  
251 CONTINUE  
PRINT 9991  
C  
C ALL DATA READ FOR THIS RUN  
C
DO 255 J = 1, N
CS(J) = C(J)
IF (C(J) .GE. 0.0) GO TO 255
C(J) = -C(J)
DO 253 I = 1, M
B(I) = B(I) + A (I, J)
253 A(I, J) = -A(I, J)
255 CONTINUE
PRINT 9600, (C(J), J = 1, N)
PRINT 9992
PRINT 9600, (B(I), I = 1, M)
PRINT 9992
DO 260 I = 1, M
PRINT 9600, (A(I, J), J = 1, N)
PRINT 9991
260 CONTINUE
9600 FORMAT (6(2X, IPE15.8))
IF (ZBAR .GT. 0.0) GO TO 300
ZBAR = 0.0
DO 275 J = 1, N
ZBAR = ZBAR + C(J)
300 ZS = 0.0
DO 325 I = 1, M
DO 325 J = 1, N
325 BS(I) = B(I)
DO 330 J = 1, N
330 NS(J) = J
IF (L .EQ. 0) GO TO 400
DO 375 K = 1, L
J1 = S(K)
K1 = IABS(J1)
NS(K1) = 0
IF (J1 .LE. 0) GO TO 375
ZS = ZS + C(J1)
DO 350 I = 1, M
350 BS(I) = BS(I) + A (I, J1)
375 CONTINUE
380 CONTINUE
IF (MO + ISCMAX .GT. 50) ISCMAX = 50
I1 = MO + ISCMAX
DO 425 I = 1, I1
NF(I) = N
DO 425 J = 1, N
J1F = J
425 CONTINUE
CALL DATIME (0, ITO)
ITI = ITO
GO TO 1910
C C initialization complete
C 1000 CONTINUE
IF (SC .EQ. 0) GO TO 2400
C surrogate constraints go here
JSCFR = JSCFR + 1
IF (JSCFR .GT. JSFIR) GO TO 2400
ML = N - L
IF (ML .LE. 1) GO TO 2400
JSCFR = 0
1050 DO 1060 J = 1, N
1060 MS(J) = 0
19-

NSIMP = NSIMP + 1
IF (IL.EQ.0) GO TO 1076
DO 1075 I = 1, L
   J = ABS(S(I))
1075 NS(J) = -S(I)
   IF (NOP.NE.0) GO TO 1076
   PRINT 3600, ((S(K), SB(K)), XL, D, JH, XX, Y, OBJ, E, NOP)
   IF (NOP.NE.0) GO TO 1077
   PRINT 9600, OBJ, ZBAR
1077 CONTINUE
   I = I + 1
   IF (K(I).EQ.2) GO TO 3400
   IF (K(I).EQ.4) GO TO 100
   IF (K(I).EQ.6) GO TO 1500
   VLPS = OBJ
   IF (VLPS.LE. (-ZBAR)) GO TO 1499
   DO 1350 I = 1, N
      IF (D(I).NE. AINT(D(I)) AND NS(I).NE.0) GO TO 1500
   1350 CONTINUE
   DO 1450 J = 1, N
      IF (NS(J).EQ.0) GO TO 1450
      L = L + 1
      NS(J) = 0
      SB(L) = GC1B
      IF (D(J).NE.0.0) GO TO 1400
      S(L) = -J
      GO TO 1450
1450 CONTINUE
   1400 S(L) = J
   ZS = ZS + C(J)
   DO 1425 L = 1, M
      BS(L) = BS(L) + A(I, J)
1425 CONTINUE
   1450 CONTINUE
   M = M + 1
   BS(M) = ZBAR
   DO 1505 I = 1, MO
      BMP1 = ZBAR
1505 CONTINUE
   DO 1510 I = 1, M
      IF (ISCMAX.LT.0) GO TO 1599
      BMP1 = ZBAR
   1510 CONTINUE
   DO 1505 I = 1, MO
      IF (ABS(BMP1 - B(M)).LE.0.0005) GO TO 1599
      IF (M - MO .LT. ISCMAX) GO TO 1520
      DO 1510 I = M1, M
      BMP1 = B(I + 1)
      BS(I) = BS(I + 1)
      DO 1510 J = 1, N
1510 A(I, J) = A(I, J) + BMP1
      M = M + 1
   1520 BMP1 = B(M) + BMP1
   DO 1550 J = 1, N
      ZJH = XX(J)
      IF (ZJH).GE.(-M) ZJH = -ZJH
   1550 A(M + 1, J) = ZJH
      M = M + 1
   1520 CONTINUE
   DO 1575 K = 1, L
      BS(K) = B(M)
   1575 CONTINUE
   DO 1575 K = 1, L
      K = SK(I)
      IF (K.LT.0) GO TO 1575
   1575 CONTINUE
   00001610 00001620 00001630 00001640 00001650 00001660 00001670 00001680 00001690 00001700 00001710 00001720 00001730 00001740 00001750 00001760 00001770 00001780 00001790 00001800 00001810 00001820 00001830 00001840 00001850 00001860 00001870 00001880 00001890 00001900 00001910 00001920 00001930 00001940 00001950 00001960 00001970 00001980 00001990 00002000 00002010 00002020 00002030 00002040 00002050 00002060 00002070 00002080 00002090 00002100 00002110 00002120 00002130 00002140 00002150 00002160 00002170 00002180 00002190 00002200 00002210 00002220 00002230 00002240 00002250 00002260 00002270 00002280 00002290 00002300 00002310 00002320 00002330 00002340 00002350 00002360 00002370 00002380 00002390 00002400
BS(M)=BS(M)+A(M,K1)
1575 CONTINUE
IF (NOP.NE.0) GO TO 1599
PRINT 1598,M
PRINT 9600,(A(M,J),J=1,N),B(M),BS(M)
1598 FORMAT (22HOSURROGATE CONSTRAINTS,2X,I4)
1599 IF (KU11.EQ.6) GO TO 3400
1900 GO TO 2400
1910 IJK=0
1920 CONTINUE
IF (ZS.GE.ZBAR) GO TO 3100
DO 1950 I=1,M
1950 CONTINUE
IF (BS(I1).LT.0.0) GO TO 1980
GO TO 2320
1980 CONTINUE
DO 2000 J=1,N
IF (NS(J).EQ.0) GO TO 2000
IF (ZS(J).LT.ZBAR) GO TO 2000
NS(J)=0
L=L+1
SB(L)=BCIB
SL(I)=J
2000 CONTINUE
KINS=0
< (IJK.EQ.1) GO TO 2220
(IJK.EQ.2) GO TO 1000
IJK=1
IF (M.LT.M1) GO TO 2025
MSC=0
I=1
I2=M
GO TO 2050
2025 MSC=1
I=1
I2=M
2050 DO 2220 I=11,12
C=BS(I)
DO 2100 J=1,N
IF (NS(J).EQ.0) GO TO 2100
IF (A(I,J).GT.0.0) Q=Q+A(I,J)
2100 CONTINUE
Q=Q
2110 IF (Q.LT.0.0) GO TO 3000
K=NF(11)
DO 2200 K1=1,K
J1=JF(I1,K1)
IF (NS(J1).EQ.0) GO TO 2200
2120 IF (Q.GE.ABS(A(I1,J1))) GO TO 2200
NS(J1)=0
L=L+1
SB(L)=BCIB
IF (A(I1,J1).GT.0.0) GO TO 2150
S(L)=J1
GO TO 2200
2150 S(L)=J1
ZS=ZS+C(J1)
DO 2175 I=1,M
2175 BS(I1)=BS(I1)+A(I1,J1)
KINS=KINS+1
2200 CONTINUE
IF (KINS.GE.KINS) GO TO 1920
2220 CONTINUE
IF (MSC.EQ.0) GO TO 2025
IF (KINS.EQ.0) GO TO 1000
JK=2
GO TO 1920
C 4A
2320 CONTINUE
IF (M.EQ.M0) GO TO 2340
DO 2325 I=M1,M
BO(I)=BO(I)+ZS-ZKBAR-ZBAR
2325 BS1(I)=BS1(I)+ZS-ZKBAR-ZBAR
2340 ZBAR=ZS-ZKBAR
DO 2350 J=1,N
2350 SWAX(J)=SW(J)
GO TO 3300
2400 K1=0
DO 2500 J=1,N
IF (NS(J).EQ.0) GO TO 2500
IF (ITB.EQ.0) GO TO 2430
IF (ZS+Z(J).EQ.ZBAR) GO TO 2500
DO 2450 I=1,M
IF (A(I,J).LE.0.0) GO TO 2500
IF (BS(I).GE.0.0) GO TO 2450
2430 CONTINUE
K1=K1+1
T(K1)=J
GO TO 2500
2450 CONTINUE
2500 CONTINUE
IF (K1.EQ.0) GO TO 3200
NAP=NAP+1
P=P-1.0E10
DO 2575 K=1,K1
J=T(K)
P1=0.0
DO 2550 I=1,M
P2=BS(I)+A(1,J)
IF (P2.GE.0.0) GO TO 2550
P1=P1+P2
2550 CONTINUE
IF (P1.LE.P) GO TO 2575
P=P1
J=J+1
2575 CONTINUE
NS(J1)=0
L=L+1
S(L)=J1
ZS=ZS+C(J1)
DO 2600 I=1,M
2600 BS(I)=BS(I)+A(1,J1)
ML=ML+1.0
GO TO 1910
3000 NCOND=NCOND+1
C PRINT 3010,1
3010 FORMAT (1HO,11,26H(TH) CONSTRAINT INFEASIBLE)
GO TO 3500
3100 NRED=NRED+1
C PRINT 3110
3110 FORMAT (33HOZ CANNOT BE REDUCED (ZS GE ZBAR))
GO TO 3500
3200 NAUG=NAUG+1
C PRINT 3210
3210 FORMAT (25HONO AUGMENTATION POSSIBLE)
   GO TO 3500
3300 NOP=NOP+1
   CALL DATETIME (O, IT3)
   IF (NOP.NE.O) GO TO 3500
   PRINT 3310,ZS
   FORMAT (25HONO AUGMENTATION POSSIBLE)
   GO TO 3500
3400 NLPF=NLPF+1
   GO TO 3500
C 4B
3500 CONTINUE
   NENUM=NUM+1
   IF (NUM.LT.KENUM) GO TO 3530
   NUM=0
   DO 3505 K=1,N
510 IF (SB(K).EQ.BCIB) ENUM=ENUM+.5*K
   CALL DATETIME (O, IT2)
   ELT1=IT2-IT0
   ELT2=ELT2-ELT1
   IT1=IT2
   ELT1=ELT1/1000.0
   ELT2=ELT2/1000.0
   IF (IT2-IT0.LT.MAXT) GO TO 3515
   MAXT=I
   GO TO 3517
351 CONTINUE
   IF (NOP.NE.O) GO TO 3700
3517 CONTINUE
   PRINT 3520,ENUM,ELT1,ELT2,L
   3520 FORMAT (25HONO BETTER SOLUTION FOUND, 5X, 2HZ-,1PE15.8)
530 CONTINUE
   IF (MAX.LT.0) PRINT 3600, (15(2X,1E14,A1))
   IF (MAX.LT.0) GO TO 3738
C 4B
3700 NFATH=NFATH+1
3710 IF (SB(L).EQ.BLANK) GO TO 3900
   J=JABS(S(L))
   NSJ=J+1
   IF (S(L).LT.0) GO TO 3735
   5X=25-CJ
   DO 3725 L=1,M
525 BS(L)=BS(L)-A(J,J)
3735 BS(L)=BLANK
   S(L)=0
   L=L-1
   IF (L.GT.0) GO TO 3710
C FINISHED
3738 CONTINUE
   PRINT 3739,H1,H2
3739 FORMAT (1H1, 5X, 2A6)
   DO 3740 J=1,N
   3740
DO 3742 J=1,N
K=IABS(SMAX(J))
IF (K.EQ.0) GO TO 3744
3742 S(K)=1
DO 3744 K=1,N
IF (S(K).NE.0) GO TO 3746
SMAX(J)=K
J=J+1
3746 CONTINUE
CALL DATIME (0,IT2)
ELT1=IT2-IT0
ELT1=ELT1/1000.0
IF (MAX(ELT).GT.0) GO TO 3752
PRINT 3750,ELT1
3750 FORMAT (30HIMPLICIT ENUMERATION COMPLETE,5X,11HTOTAL TIME=,F8.3)
GO TO 3758
3752 PRINT 3755,ELT1
3755 FORMAT (14HOTIME EXCEEDED,5X,11HTOTAL TIME=,F8.3)
3758 CONTINUE
ZBAR=ZBAR+ZBAR
PRINT 3760,ZBAR
3760 FORMAT (32HLEAST Z AFTER VARIABLE CHANGE =,1PE15.8)
I=0
DO 3810 K=1,N
T(K)=0
DO 3820 K=1,N
K=IABS(SMAX(K))
3820 IF (SMAX(K).GT.0) T(K)=K
PRINT 3830,T(K),K=1,N
3830 FORMAT (15H(4X,I3)
IF (T(K).NE.0) GO TO 3845
ZBAR=0.0
DO 3835 J=1,N
K=IABS(SMAX(J))
IF (CS(K).LT.0.0) SMAX(J)=SMAX(J)
IF (SMAX(K).GT.0) ZBAR=ZBAR+CS(K)
3835 CONTINUE
PRINT 3840,ZBAR
3840 FORMAT (32HLEAST Z BEFORE VARIABLE CHANGE =,1PE15.8)
I=1
GO TO 3800
3845 CONTINUE
ELT3=ELT3-IT0
ELT3=ELT3/1000.0
NITER=NFATH*NFATH-1
PRINT 3850,NOPR,NRED,NCON,NAUG,NID,NLPF,NSIMP,NITER,ELT3
3850 FORMAT (23HNO. FEASIBLE SOLUTIONS,15/
* 11H IS GE 2BAR,15,6H TIMES/
* 22H CONSTRAINT INFEASIBLE,15,6H TIMES/
* 24H AUGMENTATION IMPOSSIBLE,15,6H TIMES/
* 27H AUGMENTATION POSSIBLE,15,6H TIMES/
* 14H INTEGER DUALS,15,6H TIMES/
* 12H P FATHOMED,15,6H TIMES/
* 10H P CALLED,15,6H TIMES/
* 15H NO. ITERATIONS,15/
* 26H LAST FEASIBLE SOLUTION AT,5H,9H SECONDS)
GO TO 100
3900 SB(L)=BCIB
SB(L)=S(L)
J=ABS(S(I,L))
IF (S(L).GT.0) GO TO 3950
Z=Z+Z(C,J)
DO 3925 I=1,M
3925 BS(I)=BS(I)-A(I,J)
GO TO 1910
3950 Z=Z+Z(C,J)
DO 3975 I=1,M
3975 BS(I)=BS(I)+A(I,J)
GO TO 1910
END

$IBFTC SIMPLE
C AUTOMATIC SIMPLEX REDUNDANT EQUATIONS CAUSE INFEASIBILITY
SUBROUTINE SIMPLE(INFLAG,MX,NN,A,B,C,KO,KB,F,JH,X,Y,OBJ,E,NOP)
REAL B(I),C(I,J),P(I),X(I),Y(I),OBJ
REAL E(90,90)
INTEGER INFLAG,HX,NN,KO(6),KB(11),JH(11)
FOURIER
C THE FOLLOWING DIMENSION SHOULD BE THE SAME HERE AS IT IS IN CALLER.
REAL A(50,901)
REAL AA,AIJT,BB,COST,DT,RCOST,TEXP,TPIV,TY,XOLO,XX,XY,YI,YMAX,EM
INTEGER I,IA,INVC,IA,ITER,J,JT,K,KB,RJ,L,
INFLAG.MX,NN,KO(6),KB(11),JH(11)
INTEGER NCUT,NUMVR,NUMP
LOGICAL TRIG,VER
LOGICAL FINV,FRZ,SCH
COMMON /BLS/MS(90),ZBAR
SET INITIAL VALUES, SET CONSTANT VALUES
FINV = .FALSE.
TRIG = .FALSE.
ITER = 0
LPSEQ = LPSEQ+1
NUMVR = 0
NUMP = 0
M = MX
N = NN
TEXP = .5**16
NVER = M/2 + 5
NCUT = 4*M + 10
IF (INFLAG.EQ.0) GO TO 1410
C IMPOSE CORRECT TEMPERATURE ON ROWS
FRRZ = .TRUE.
DO 1960 L=1,M
IF (MS(L).EQ.NFIL) GO TO 1955
IF(MS(L).EQ.NFIL).GT.0.OR.(MS(L).EQ.0.AND.X(L).GE.0)) GO TO 1950
1=L
1920 IF (JH(I).GT.0) GO TO 1925
1925 IF (JH(I).GT.0) GO TO 1930
C IF JH DISAGREES WITH MS DO SPECIAL PIVOT
IF (MS(L).GT.0.AND.JH(L).GE.(-M)) GO TO 1950
IF (MS(L).LT.0.AND.JH(L).LT.(-M)) GO TO 1950
C SPECIAL PIVOT: SWITCH SINGLETONS
1925 DO 1926 J=1,M
P(J) = P(J) + E(I,J)
E(J,J) = E(J,J)
1926 CONTINUE
OBJ = OBJ + X(I)
X(I) = -X(I)
JH(I) = JH(I)
IF (JH(I).GE.(-M)) JH(I) = -L-M
IF (JHL.LT.(-M)) JHL = -L
GO TO 1950
C DO FULL PIVOT ON SINGLETON
1930 JT = -I
COST = P(I)
IF (MS(I).GT.0) GO TO 1931
JT = JT-M
COST = 1.-COST
1931 EN = 1.
GO TO 630
C GET COLUMN(JT)
1932 SCH = .FALSE.
IF (COST.GT.O.) GO TO 1938
1935 GO TO 1000
C SELECT ROW(IR)
1936 IF (IR.NE.0.GR.SCH) GO TO 1940
SCH = .TRUE.
1938 EN = EN
DO 1937 J=1,M
Y(J) = -Y(J)
1937 CONTINUE
GO TO 1935
1940 IF(SCH.AND.ABS(COST).GT.TPIV).OR.(IR.EQ.0.) GO TO 1980
1941 IF (EN.GT.O.) GO TO 1945
DO 1942 J=1,M
Y(J) = -Y(J)
1942 CONTINUE
1945 GO TO 901
C PIVOT(IR.JT)
1950 NF(I) = MS(I)
1955 IF(JHL.LT.0) GO TO 1960
IA = JHL
KB(I) = L
1960 CONTINUE
C* START WITH SINGLETON BASIS
1410 DO 1402 J=1,M
KB(J) = 0
1402 CONTINUE
FFRZ = .FALSE.
1400 DO 1401 T = 1,M
JH(T) = -I
NF(I) = MS(I)
IF (INF(I).LT.0.OR.(INF(I).EQ.0.).AND.(B(I).LT.0.)) JH(I) = -I-M
1401 CONTINUE
C* CREATE INVERSE FROM 'KB' AND 'JH' (STEP 7)
1320 VER = .TRUE.
INVC = 0
NUMVR = NUMVR +1
TRIG = .FALSE.
OBJ = 0.
DO 1113 T = 1,M
DO 1151 J=1,M
EIJ(J) = 0.
1113 CONTINUE
IF (JH(I).LT.(-M)) GO TO 1111
IF (JH(I).GT.0) JH(I) = 0
EIJ(I) = 1.
P(I) = 0.
X(I) = B(I)
GO TO 1113

1111 E(I) = -1.
P(I) = +1.
OBJ = OBJ + B(I)
X(I) = -B(I)

1113 CONTINUE
DO 1102 JT = 1,N
IF (KBJ(JT).EQ.0) GO TO 1107
GO TO 600
GET COLUMN(JT)
C
1114 TV = TPIV
IR = 0.
COST = C(JT)
DO 1104 I = 1,M
COST = COST + A(J,T)*P(I)
IF(JH(JT).NE.0.OR.X(I).NE.0.OR.ABS(Y(I)).LE.TY) GO TO 1104
TV = ABS(Y(I))
IR = I
1104 CONTINUE
IF (IR.NE.0) GO TO 1119
1119 IF (IR.NE.0) GO TO 900
C
PIVOT IR,JT
FINV = .TRUE.
IF (INP.EQ.0) PRINT 1199,LPSEP
1199 FORMAT(15H0INVERT FAIL LP,14)
GO TO 1410
1102 CONTINUE
C* PERFORM A SIMPLEX ITERATION
1200 VER = .FALSE.
500 DO 503 I = 1,M
IF (INF(I).EQ.0.AND.X(I).LT.0.) X(I) = 0.
503 CONTINUE
C* FIND MINIMUM REDUCED COST (STEP 3)
599 JT = 0
BB = 0.0
DO 701 J = 1,N
IF (KBJ(J).NE.0) GO TO 701
DT = C(J)
DO 303 I = 1,M
DT = DT + A(J,I)*P(I)
303 CONTINUE
IF (DT.GE.BB) GO TO 701
BB = DT
JT = J
701 CONTINUE
DO 702 I = 1,M
IF (JH(I).LT.0) GO TO 702
IF (P(I).LT.BB) GO TO 702
IF (I .LT.EQ.P(I)) GO TO 702
BB = 1.-P(I)
JT = -I-M
GO TO 702
703   BB = P(I)
    JT = -1
702 CONTINUE
    COST = BB
    IF (JT.EQ.0) GO TO 203
    IF (ITER.GE.NCUT) GO TO 160
    ITER = ITER + 1
C*  MULTIPLY INVERSE TIMES A(I,JT)  (STEP 4)
    IF (JT.LT.0) GO TO 630
    CONTINUE
C BEGIN SUBROUTINE GET COLUMN(JT)
600   DO 610  I = 1,M
      Y(I) = 0.0
610 CONTINUE
    DO 605  I = 1,M
      AIJT = AIJT(I)
      IF (AIJT.EQ.0.) GO TO 605
      DO 606  J = 1,M
        Y(J) = Y(J) + AIJT*E(J,I)
606 CONTINUE
605 CONTINUE
    GO TO 640
630   JT2 = -JT
    EM = 1.
    IF (JT2.LE.M) GO TO 631
    JT2 = JT2 - M
    EM = -1.
631   DO 632  I = 1,M
      Y(I) = EM*E(I,JT2)
632 CONTINUE
    YMAX = 0.
    DO 620  I = 1,M
      YMAX = AMAX1(A,I),YMAX)
620 CONTINUE
    TPIV = YMAX*TEXP
    END OF GET COLUMN
    IF (FFRZ) GO TO 1932
    IF (VFR) GO TO 1114
    RCOST = YMAX/BB
    IF (TRIG.AND.BB.GE.(1-TPIV)) GO TO 203
    TRIG = BB.GE.(1-TPIV)
    TPIV = TPIV
C*  SELECT PIVOT ROW
1000  AA = TPIV
    IR = 0
1002   DO 1003  I = 1,M
      IF (X(I).NE.0..OR.Y(I).LE.AA.OR.NF(I).NE.0) GO TO 1003
      AA = Y(I)
    IR = I
1003 CONTINUE
    IF (IR.NE.0) GO TO 1020
    AA = 0.
    DO 1010  I = 1,M
      IF (NF(I).NE.0..OR.Y(I).LE.TPIV.OR.Y(I).LE.AA*X(I)) GO TO 1010
      AA = Y(I)/X(I)
    IR = I
1010 CONTINUE
1020   IF (FFRZ) GO TO 1936
    IF (IR.EQ.0) GO TO 207
C*  PIVOT ON (IR,JT)  (STEP 6)
901   IA = JM(IR)
    IF (IA.GT.IR) KB(IA) = 0
    IF (IA.LT.IR)
C BEGIN SUBROUTINE PIVOT(IR,JT)
900 NUMPV = NUMPV + 1
JH(IR) = JT
IF (JT.GT.0) KBJ(JT) = IR
YI = -Y(IR)
Y(IR) = -1.0
DO 904 J = 1,M
XY = E(J,J)/YI
IF (XY.EQ.0.) GO TO 904
P(J) = P(J) + COST * XY
E(IR,J) = 0.
DO 906 I = 1,M
E(I,J) = E(I,J) + XY * Y(I)
906 CONTINUE
904 CONTINUE
XY = X(IR)/YI
DO 908 I = 1,M
XOLD = X(I)
X(I) = XOLD + XY * Y(I)
908 CONTINUE
Y(IR) = -YI
X(IR) = -XY
C END OF PIVOT
OBJ = OBJ + XY*COST
IF (VER) GO TO 1102
C EXCHANGE ROWS IF SLACK PIVOTED IN WRONG ROW
IF (JT.GT.0.OR.JT2.EQ.IR) GO TO 907
XY = X(IR)
X(IR) = X(JT2)
X(JT2) = XY
DO 909 I = 1,M
XY = E(I,I)
E(IR,I) = E(JT2,I)
E(JT2,I) = XY
909 CONTINUE
IA = JH(JT2)
JH(JT2) = JT
JH(IR) = IA
KBIA = IR
907 INVC = INVC + 1
C TO STEP 1 IF NOT INVERTING, TO STEP 7 IF INVERTING
IF (FFRZ) GO TO 1950
IF (OBJ.GE.ZBAR) GO TO 180
IF (FINV) GO TO 1200
910 IF (INVC.GE.NVER) GO TO 1320
GO TO 1200
C END OF ALGORITHM, SET EXIT VALUES
C 207 IF (RCOST.LE.(-1000.)) GO TO 203
C INFINITE SOLUTION
K = 2
GO TO 250
180 K = 6
GO TO 250
C PROBLEM IS CYCLING PERHAPS
160 K = 4
PRINT 161,LPSEQ
161 FORMAT(31,H0,40)
GO TO 250
C FEASIBLE OR INFEASIBLE SOLUTION
203 K = 0
250 DO 1399 J = 1,N
   XX = 0.0
   KBJ = KB(J)
   IF (KBJ.NE.0) XX = X(KBJ)
   KB(J) = LL
1399 CONTINUE
   KO(1) = K
   KO(2) = ITER
   KO(3) = INVC
   KO(4) = NUMVR
   KO(5) = NUMPV
   KO(6) = JT
   IF (NOP.NE.0) RETURN
   PRINT 162,LPSEQ,(KO(I),I=1,6)
   162 FORMAT(3HLP,l$(16H KO  ,616)
   RETURN
   IF (NOP.EQ.O) PRINT 1981,LPSEQ,L,IR,SCH,COST
   1981 FORMAT(3HLP,l$(14,12H FAIL, SLACK,13,4H IR=13,5H SCH=L1,3H C=F19.6100008040
   IF (IR.NE.0) GO TO 1941
   1941 FORMAT(6HGO TO 1410
   END
REFERENCES


3. --------- (October 1967), "Recent Computational Experience With Three Classes of Integer Linear Programs," P-3699, The RAND Corporation.