An L-Shaped Method Computer Code for Multi-Stage Stochastic Linear Programs

John R. Birge
Department of Industrial & Operations Engineering
The University of Michigan
Ann Arbor, MI 48109

Technical Report 84-19

An L-Shaped Method Computer Code for Multi-Stage Stochastic Linear Programs

John R. Birge
Department of Industrial and Operations Engineering
The University of Michigan
Ann Arbor, MI 48109
USA

Abstract

A computer code implementing the L-shaped method of Van Slyke and Wets is described. The method is generalized to apply to problems with up to three periods and up to three hundred seventy-five different future scenarios. The main subroutines are described and an example of input and output formats is given.

1. Introduction

The L-shaped method for two-stage stochastic linear programs was given by Van Slyke and Wets [1969]. It is an outer linearization procedure that approximates the convex objective term in the stochastic program by successively appending supporting hyperplanes. This paper describes a multi-stage implementation of this algorithm in which the supports are found by optimizing a nested sequence of problems. The mechanics of this algorithm and its convergence properties are described in Birge [1982].

The method is a type of nested decomposition procedure that can be compared with inner linearization procedures such as those of Glassey [1971, 1973] and Ho and Manne [1974]. It is also related to basis factorization approaches (Kall [1979], Strazicky [1980], Birge [1984]) and inner linearization of the dual (Dantzig and Madansky [1961]).

The basic steps of the algorithm are described in Section 2. The main subroutines of the computer code are then given in Section 3. Significant variables and data structures are also described. Input and output formats are detailed in Section 4 along with examples of their form. Section 5 presents some observations and potential extensions.

2. Algorithm Description

The multi-stage stochastic linear program considered by the algorithm is

$$
\begin{aligned}
\min z = c_1x_1 + E_{\xi_2}\, [\min c_2x_2 + \cdots + E_{\xi_t}\, &[\min C_Tx_T]\cdots] && (1)\\
\text{s.t.}\quad A_1x_1 \qquad\qquad\qquad\qquad\qquad & = b_1,\\
B_1x_1 \qquad + \quad A_2x_2 \qquad\qquad\qquad & = \xi_2,\\
\ddots \qquad\qquad\qquad & \quad\vdots\\
B_{T-1}\, x_{T-1} + A_Tx_T & = \xi_T,\\
x_t \geq 0,\ t=1,\dots,T,\quad {}_t^{t}\ \ {}_t,\ t=2,\dots,T, &
\end{aligned}
$$

where c_t is a known vector in R^{n_t} for $t=1,\dots,T$, b_1 is a known vector in R^{m_1}, ξ_t is a random m_2 - vector defined on the probability space (Ξ_t, F_t, F_t) for $t=2,\dots,T$, and A_t and B_t are correspondingly dimensioned dimensioned known real-valued matrices. "E_{ξ_t}" denotes mathematical expectation with respect to ξ_t.

The L-shaped method of Van Slyke and Wets [1969] applies to (1) when T=2. Methods for the multi-stage problem have generally assumed a specific structure for the problem. Beale, et al. [1980] and Ashford [1982] for example, consider a multi-stage production problem and implement an appropriate approximation. The generalization of the L-shaped method implemented in the computer code described here and introduced in Birge [1980, 1982] does not, however, require any special structure except that the random variables ξ_t are finitely distributed.

The algorithm is called the Nested Decomposition for Stochastic Programming Algorithm (NDSPA). It is based on the observation that given a realization ξ_t^j of the random vector in period t and given a solution $x_{t-1}^{a(j)}$ from period t-1, the decision problem at period t can be written (see Wets [1966])

$$\min \quad c_t x_t^j + Q_{t+1}(x_t^j) \tag{2.0}$$

$$\text{s.t.} \quad A_t x_t^j = \xi_t^j + B_{t-1} x_{t-1}^{a(j)} \tag{2.1}$$

$$D_t^{\ell,j} x_t^j \geq d_t^{\ell,j}, \ell=1,\ldots,r_t^j, \tag{2.2}$$

$$x_t \geq 0,$$

where $Q_{t+1}(x_t)$ is a convex function, $D_t^{\ell,j} \varepsilon R^{n_t}$ for all ℓ, and $r_t^j \leq m_{t+1}$.

Program (2) can then be solved using a <u>relaxed master problem</u>:

$$\min \quad c_t x_t^j + \theta_t^j \tag{3.0}$$

$$\text{s.t.} \quad A_t x_t^j = \xi_t^j + B_{t-1} x_{t-1}^{a(j)} \tag{3.1}$$

$$D_t^{\ell,j} x_t^j \geq d_t^{\ell,j}, \ell=1,\ldots,r_t^j \tag{3.2}$$

$$E_t^{\ell,j} x_t^j + \theta_t^j \geq e_z^{\ell,j}, \ell=1,\ldots,s_t^j \tag{3.3}$$

$$x_t^j \geqq 0. \tag{3.4}$$

Program (3) is solved to obtain $(\bar{x}_t^j, \bar{\theta}_t^j)$. If $\bar{\theta}_t^j < Q_{t+1}(\bar{x}_t^j)$ then another <u>optimality cut</u> (3.3) is added to (3) and (3) is re-solved. If $\bar{x}_t^j$ forces infeasibility in any future period then a <u>feasibility cut</u> (3.2) is added to (3). This process is repeated until $\bar{\theta}_t^j \geq Q_{t+1}(\bar{x}_t^j)$.

For implementation in multi-stage problems, it is assumed that there are a finite number K_t of <u>scenarios</u> in each period t. The scenarios consist of all possible realizations of the random vectors from periods 2 through t. For every period t scenario j, there corresponds a unique <u>ancestor</u> scenario a(j) in period t-1 and, perhaps, several descendant scenarios d(j) in period t+1. NDSPA solves (1) by first obtaining a feasible solution to (3) for all t and j and by then sequentially solving (2) using the relaxation in (3) from periods T to one.

NDSPA

Step 0. Solve (3) for t=1 (dropping the scenario index j) where $\theta_1 = 0$, $r_1 = s_1 = 0$ and (3.1) is replaced by $A_1 x_1 = b_1$. Set $\theta_t^j = 0$ and $r_t^j = s_t^j = 0$ in (3) for all t and scenarios j at t. (The indices r_t^j and s_t^j are updated whenever a constraint (3.2) or (3.3) is added to (3)).

Step 1. If (3) is infeasible for t=1, STOP. The problem (1) is infeasible. Otherwise, let $\bar{x}_1$ be the current optimal solution of (3) for t=1. Use $\bar{x}_1$ as in input in (3.1) for t=2. Solve (3) for t=2 and all ξ_2^j, $j=1,\ldots,K_2$. If any period two problem (3) is infeasible, then add a feasibility cut (3.2) to (3) for t=1, resolve (3) for t=1, and return to 1. Otherwise let t=2 and go to 2.

Step 2. a). Let the current period t optimal solutions be $\bar{x}_t^j$, $j=1,\ldots,K_t$. Solve (3) for t+1 and all $j=1,\ldots,K_{t+1}$ using the appropriate ancestor solution $\bar{x}_t^j$ in (3.1).

b). If any period t+1 problem is infeasible, add a feasibility cut (3.2) to the corresponding ancestor period t problem and resolve that problem.

If the period t problem is infeasible, let t = t-1.

If t=1, go to 1.

Otherwise, return to 2.a.

Otherwise, return to 2.a.

Otherwise, all period t+1 problems (3) are feasible.

If $t \leq T-2$, let $t = t+1$ and return to 2.a.

Otherwise ($t=T-1$), remove the $\theta_\tau^j = 0$ restriction for all periods τ and scenarios j at τ. Let the current value of each θ_τ^j be $\theta_\tau^j = -\infty$ if no constraints (3.3) are present. Go to 3.

Step 3. a). Find $E_t^{\ell,j}$ and $e_t^{\ell,j}$ for a new constraint (3.3) at each scenario t problem (3) using the current period t+1 solutions.

b). If there exists j such that

$$\bar{\theta}_t^j < e_t^{\ell,j} - E_t^{\ell,j} \bar{x}_t^j, \tag{4}$$

then add the new constraint (3.3) to each period t problem (3) for which (4) holds. Solve each period t problem (3). Use the resulting solutions ($\bar{x}_t^j$, θ_t^j) to form (3.1) for the corresponding descendant period t+1 problems (3) and resolve each period t+1 problem (3).

If $t < T-1$, let $t = t+1$ and go to 2.a.

Otherwise, return to 3.a.

Otherwise, $\bar{\theta}_t^j = e_t^{\ell,j} - E_t^{\ell,j} \bar{x}_t^j$ for all scenarios j at t.

If $t > 1$, let $t = t-1$ and return to 3.a.

Otherwise, STOP. The current solutions $\bar{x}_\tau^j$, $\tau = 1,\ldots,T$ form an optimal solution of (1).

Steps 1 and 2 of NDSPA represent a <u>forward pass</u> to obtain feasibility in each scenario subproblems. Step (3) is a <u>backward pass</u> that solves (2) beginning with period T and passing backward to period 1. Unboundedness may be handled explicitly in the program following the procedure in Van Slyke and Wets [1969] but in the computer code of NDSPA all variables are upper bounded and hence unboundedness is avoided. For period T, the computer code also has a special procedure for solving (3). It uses the bunching (see Wets [1983]) method to look through all realizations of ξ_T and find those for which a given basis is optimal. This procedure is described in the next section and represents an alternative to the sifting procedure of Garstka and Rutenberg [1973].

Experimental results using NDSPA have been encouraging. In Birge [1980, 1982], NDSPA is compared with a piecewise linear partitioning algorithm, a basis factorization procedure and the code MINOS (Murtagh and Saunders [1978]) on a set of staircase test problems from Ho and Loute [1981]. NDSPA consistently outperformed the other methods except on one problem in which its storage limitations were exceeded. In general, the results compared favorably with those of Kallberg and Kusy [1976] and Kallberg, White, and Ziemba [1982] for simple recourse problems. Each stochastic problem was solved in less than twice the time required to solve the deterministic problem with expectations substituted for the random variables.

3. The NDST3 Computer Code-Primary Subroutines

NDSPA has been coded in FORTRAN in a current version called NDST3 (see Appendix). This code allows for three periods including three second period scenarios and one hundred twenty-five third period scenarios. Each scenario problem (3) is limited to three hundred fifty rows and six hundred columns. Within any scenario problem (3), there can be at most three thousand nonzero elements. Tolerances can be set in the BLOCK DATA section and in the example are set at 10^{-7} for zero tolerance, 10^{-5} for pivot tolerance, 10^{-4} for reduced cost tolerance and 10^{-10} for small tolerance. The linear programming sections of the code are from L PM-1 written by J.A. Tomlin (Pfefferkorn and Tomlin [1976]).

Many variables in NDST3 have multiple subscripts. This questionable programming technique is used to make the scenario obvious. For example, XLB(2, 3, 2, 1) is the lower bound on the second variable in scenario 1 in period 3 with ancestor scenario 2 in period 2. In general, the last three subscripts of all variables with more than two subscripts are (JCUR, JPER(2), JPER(3)) where JCUR indicates the period of the scenario, JPER(2) indicates the period 2 ancestor scenario and JPER(3) denotes the period 3 scenario. This last period scenario is not used in the current version version of NDST3 but has been used for a four period implementation. The current version is limited to three periods to avoid excessive storage requirements. The code can process four period problems if the period 3 index is incremented in all array definitions and sufficient memory is available. The subroutine SHIFTR, which manipulates data storage, must also be updated if the dimensions are changed.

The main variables in the code are stored in the blank common block. These variables and their descriptions follow

Variable	Definition
B(i,j,k,l)	Current right-hand side element i in period j and scenario k,l
X(i,j,k,l)	Current value of variable basic in row i at period j and scenario k,l
XLB(i,j,k,l) and XUB(i,j,k,l)	Lower and upper bounds of variable i at j,k,l
XKSI(i,j,k,l)	Current realization of random vector in row i at j,k,l
YPI(i,j,k,l)	Current dual variable value for row i at j,k,l
NROW(j,k,l)	Current number of rows at j,k,l
NCOL(j,k,l)	Current number of columns at j,k,l
NELM(j,k,l)	Current number of nonzero elements at j,k,l
JH(i,j,k,l)	Variable basic in row i at j,k,l
KINBAS(i,j,k,l)	Status (basic, nonbasic) of variable i at j,k,l
LA,IA,A	Linked lists of A_t matrix elements
LE,IE,E	Linked lists of elements in eta vector form of basis inverse
LBN,IBN,ABN	Linked lists of elements in B_t matrices
PROB(j,k,l)	Probability of scenario j,k,l

The important variables in BLOCK 3 are

NND(i)	Number of scenarios in peiod i
NPASS	Number of passes from period t to t-1 or t+1

JPER(i)	Current scenario realization in period i
JCUR	Current period
JPASS	Indicator of forward or backward pass; JPASS=1 for forward, JPASS=2 for backward
NPER	Number of periods T

In BLOCK 4, the significant variables are

XTOPT	Value of $- e_T^{\ell,j} + E_T^{\ell,j} \bar{x}_T^j$ for checking for optimality
PRBY(i,j)	Probability of j*th* realization of i*th* random element in stochastic vector in period T
PRST(i,j,k)	Joint probability of i*th* realization of first random element element, j*th* realization of second, and k*th* realization of third for stochastic vector at T
CBST(i,j)	Value of j*th* realization of i*th* random element in stochastic vector at T
NCUR(i)	Current realization of i*th* random element at T
IBST(i)	Row of i*th* random element at T
NST	Number of random elements at T

The code NDST3 assumes that specific random vectors (with specific probabilities) are assigned for periods 2 through T-1 and that at period T the random vector includes NST independent random elements. The bunching approach can then be easily applied to these possibilities.

The main program in NDST3 organizes the algorithm and calls subroutines to implement the steps of NDSPA. The main routines called in this segment are:

INPUT	accepts all data input;
INCHK	echoes input;
NORMAL	solves the linear program in (3);
STRPRT	reports on current solution;
NDCOM	directs the algorithm for t<T;
PARSFT	controls the algorithm for t=T;
WRAPUP	writes output.

The main routine calls NORMAL to solve (3) if t<T and then calls NDCOM to determine which problem to solve next. If t=T, PARSFT is called to solve (3) for period T and determine the next step of the algorithm. JCUR(t) is set equal to NPER+1(T+1) whenever a terminating condition (infeasible or optimal) is met.

The following routines are all used by NORMAL in solving the linear program (3):

RHCHCK	checks now residicals;
BTRAN	performs backward transformation;
FORMC	forms objective function vector and checks feasibility;
PRICE	computes reduced costs and picks entering column;
CHUZR	performs minimum ratio test and determines leaving variable;
WRETA	forms new eta-vectors for product form of inverse;
SHIFTR	rearranges data storage;
INVERT	computes basis inverse using LU decomposition;
UNPACK(i)	expands ith column in A;
BUNPCK(i)	expands ith column in B;
SHFTE	shifts eta vectors around;

UPBETA updates right-hand side and basis indicators.

NORMAL reinverts the basis every INVFRQ iterations or if the maximum row residual is greater than 10 times ZTOLZE. A maximum of ITRFRQ iterations is allowed.

The subroutine NDCOM handles all steps for NDSPA for t<T. The forward pass is performed in the statements above statement number 70. The variable MSTAT is used to indicate infeasibility (QN) or feasibility (QF). If an infeasibility is found, then a feasibility cut is added in the subroutine FEASCT (below statement number 201) and t is set to t-1. If the current problem is feasible, then NDCOM determines the next subproblem to solve. If every scenario at period t has been solved, then NDCOM sets up problem (3) for period t+1 between statements 28 and 30. The subroutine BPRODX is called here to compute $B_t \, x_t^j$ and FRMRHS is called to find $\xi_{t+1}^{d(j)} + B_t \, x_t^j$.

If the algorithm has proceeded to the backward pass, the control shifts to the statements following statement 70. Again, if an infeasibility is found, then a feasibility cut is added to the corresponding ancestor scenario problem. The procedures for checking optimality follow statement 80. First, any cuts (3.2) or (3.3) that are slack (satisfied as strict inequalities) are deleted in the subroutine DLETCT. This option saves on storage and does not affect convergence. NFLG = 1 signifies that the current problem (3) solution is optimal. For t < T-1, the code follows Step 2 of NDSPA and continues to t+1. If t = T-1 and condition (4) is not met, then NDCOM follows the iterations in Step 3 of NDSPA in the statements following statement 85. If condition (4) is met in following this backwards iteration then an optimality cut (3.3) is placed on the corresponding ancestor scenario using the subroutine LKHDCT(K) where k is the preceding period. Optimality at period k is checked in the subroutine OPTCHK(K) which sets NFLG =1 if (4) is not met.

Subroutine PARSFT performs Step 3 of NDSPA for t = T. It includes the variable JSTCH(i,j,k) that indicates the number of the basis found optimal for the alternative with realizations i,j,k for random elements 1, 2, and 3 respectively, in the last period. NCUR(i) is the current realization of the i<u>th</u> random element and NXNF(i) is the realization of the i<u>th</u> random element in the first infeasible basis found by the bunching procedure. NETND(i) keeps the number of eta vectors in the i<u>th</u> basis and INFLG = 0 for no infeasibilities and 1 for an infeasibility found in passing through all alternative random vectors at T. YBX is a vector keeping $B_{T-1}\ \bar{x}_{T-1}^{a(j)}$.

The bunching procedure begins with statement 20. For the first time through the loop, NORMAL is called to obtain an optimal solution. On subsequent iterations, the procedure begins with the previous basis which is dual feasible and calls the subroutine DNORML which implements the dual simplex method. In either case, if an infeasibility is caught then a feasibility cut is made and control returns to the main program.

After having found a new optimal basis, the algorithm updates $E_{T-1}^{\ell,j}$ and $e_{T-1}^{\ell,j}$ and then loops through all right-hand side alternatives for which no feasible basis has yet been found. This begins with statement 29 and continues to statement 30 if all scenarios have not been checked. Since every scenario corresponds to the same objective function, an optimal basis for any scenario is dual feasible for all other scenarios. The appropriate right-hand side is set up between statements 50 and 53 and FTRAN is called to find the values of the basic variables. The subroutine DCHUZR is then called to determine a leaving (infeasible) variable. It returns IROWP = 0 for a feasible basis which is then optimal. If a leaving variable is found then DCHUZC is called to find an entering variable. If no entering variable is found then the current scenario is infeasible and control is returned to the main program. If an entering variable is found, then the current scenario is marked as the first scenario to check in the next bunching loop (if no scenario has been found infeasible for the current basis) and the next scenario is tested.

Whenever a scenario is found to be feasible for the current right-hand side then the values of $E_{T-1}^{\ell,j}$ and $e_{T-1}^{\ell,j}$ are updated after statement 60, and the next scenario is chosen. When an optimal basis has been found for all period T scenarios then optimality is checked after statement 29 using the subroutine XOPTCK. NFLG = 1 is returned if (4) is not met and the algorithm proceeds back to period T-1. If (4) is met then a new optimality cut is added to the ancestor period T-1 problem.

The algorithm proceeds through these subroutines until optimality is found in NDCOM (for T > 2) or PARSFT(T = 2) or until infeasibility is found in NDCOM. When one of these terminal conditions is reached, WRAPUP is called and the output described in the next section is produced.

4. Input and Output Formats

The input format for NDST3 basically follows the MPS standard for mathematical programs except in its splitting the data into periods. The example in the appendix is a test problem SCAGR7.S2 which was adapted from the staircase test problems of Ho and Loute [1981]. It contains two periods for the stochastic program, and, in the second period, there are three independent random variables with two values each. This leads to eight total scenarios.

The first row of the input contains five values used in program. Each is entered in I4 format, they are in order:

IFPROB	number of problem;
IOBJ	row of objective function (usually "1");
INVFRQ	iterations between matrix inversions;
ITRFRQ	total number of iterations allowed;
NPER	number of periods.

The next NPER rows contain the number of different right-hand side values (I4 format) for each period. The first and last periods have 1's because the first period is deterministic and the last period right-hand sides are input separately at the end of the program. The fourth row contains the probability of the first right-hand side value in F5.3 format. The next sections are ROWS, COLUMNS, and RHS sections for MPS format for all values in the first period set of constraints, $A_1 x_1 = b_1$. Following an ENDATA, lower bounds on all variables (excluding slacks) in

9F8.0 format and upper bounds in the same format are input. If an initial basis were entered then a section headed by BASIS and including columns and the corresponding row in the basis could be entered after the COLUMNS section. This format is discussed below as part of the output.

The next sections of the code include ROWS and COLUMNS sections to describe the matrix B_1 in (1). This is followed by an ENDATA and the probability of the next period's first right-hand side vector. The data for $A_2 x_2 = \xi_2^j$ would then be entered for each possible ξ_2^j and, if more periods were present, this would be followed in each case by the data for B_2 (possibly depending on j). This process of repeating the probability of ξ_t^j, giving the data for $A_t x_t = \xi_t^j$ and of then giving B_t repeats until all scenarios indicated in the command lines of the code have been input.

The last period scenario input is followed by a section marker STOCH which prompts the program to read in separate values and probabilities for random elements in the last period. For each random element, the row name is given is columns 5-12, the value of the element is given in F12.4 format in columns 25-36 and the probability of that value is given in F12.4 format in columns 50-61. Each independent element is input with at most five values total.

Another version of NDST3, called NDST3.A, has also been developed at IIASA, Laxenburg, Austria. In this code, input follows the standard format set at IIASA except for the first line of input which contains the control parameters.

NDST3 writes two output files on devices 6 and 7. The first listed in the Appendix is for device 6 and contains most of the iteration and result information. The second contains the variables that were basic in the optimal solution found by the program. That output may be inserted into an input file to provide the program with a starting basis.

The first part of the output contains the problem, the densities of A_t matrices for t = 1 and 2 and the stochastic elements in the last period including their values and probabilities. The next section of the code prints out the matrices as they are stored in the code and other information for checking input data.

The iteration log begins on line 178. PRES is the row residual, PIV is the pivot element size, IN is the incoming variable, OUT is the basis position of the outgoing variable, OBJ is the negative of the objective value, CMIN is the minimum reduced cost for variables at lower bound, CMAX is the maximum reduced cost for variables at upper bound, NINF is the number of infeasible variables in the basis, and NOPT is the number of nonoptimal variables. The first pass at the period 1 problem stops with the first feasible solution found and the vector of basic variable values is printed along with JH, KINBAS, the right-hand side, B, the current value of Y (a vector used in several operations), and the current price vector.

Iterations begin on the second period problem after control has passed through NDCOM and an infeasibility condition is met, forcing a feasibility cut. Now, the period 1 problem is solved to optimality and the period 1 solution information on this third pass is output. Again, the period 2 problem is infeasible and another feasibility cut is applied to the period 1 problem. This time, on line 346, the program states the previous cut is slack and that it is deleting row 17 with a slack value of 1500.

This process repeats until line 440 when a feasible solution is found for the second period problem. Here, each scenario is checked and the first optimal basis is feasible for all scenarios. OBJ is the negative objective value in each case, KSIPI is used to calculate $e_{T-1}^{\ell,j}$ and OBJTOT is the sum of all previous objectives times their probabilities. CURR BAS indicates which basis is optimal for each scenario.

After going through the scenarios, $Q(x) = -E_{T-1}^{\ell,j} \bar{x}_{T-1}^{a(j)} + e_{T-1}^{\ell,j}$ and $\bar{\theta}_{T-1}^{a(j)}$ are output and an optimality or "look-ahead" cut is made on the first period. The first period problem is resolved and beginning with line 512, the second period scenarios are again checked. This time four scenarios are not feasible with the first optimal basis and DNORML is used to find another optimal basis. The iteration log in DNORML is similar to that in NORMAL except that MXINF is also output as the maximum infeasibility found.

Condition (4) is again checked and lower and upper bounds on the optimal value of (1) are given. The process repeats until on lines 735 - 739, optimality is checked, and it is found that the lower and upper bounds are within one percent of each other, indicating optimality and ending the algorithm.

Solution times follow the discovery of optimality and the basic variable values for each scenario in the last period are output by the rows in which they are optimal. The solution for the first period problem is output beginning on line 1097 and the optimal dual values follow. The solution of the last problem solved in the last period is then given, ending the solution output.

The next listing in the Appendix is of the output file for the basis. This includes all variables that were basic in rows of the original problem. Variables that were basic in rows for feasibility and optimality cuts are not included. The column is listed in columns 5-12 and the row in which its basic is listed in columns 15-22.

5. Extensions and Observations

As mentioned above, NDST3 can be easily expanded to handle larger problems and more scenarios. Some care, however, must be used in maintaining storage requirements within acceptable limits. Future versions of the code are planned to eliminate some redundancy and to enable more complex problems to be solved. Other planned options are to include the possibilities for some continuous distributions and to use approximating techniques from Birge and Wets [1983] in achieving convergence within a predetermined tolerance. This has been implemented for a single random variable in a new code NDST4 and further refinements are planned.

The code has performed very well in general and in most situations outperforms general purpose linear programming codes. The one problem in which it did not perform well, SCFXM.S2, required that a large number of feasibility cuts be added to the first period problem. These cuts were dense and, without deleting slack cuts, the problem required an excessive number of nonzero elements (i.e., more than three thousand). When slack cuts were deleted, the program obtained an unstable basis that caused it not to obtain a feasible first period solution. This may be a problem inherent in decompostion algorithms because of perhaps unavoidable numerical error present in generating cuts. Two truely identical cuts may be generated that differ only in their error coefficients. This is the cutting plane analogy of the slow convergence characteristics observed in Dantzig-Wolfe decomposition (Ho[1984]). It appears that stability problems are rare but if further testing results in more of these difficulties, some testing of the integrity of cuts may have to be added.

References

Ashford, R.W. 1982. A Stochastic Programming Algorithm for Production Planning. Scicon report, Milton Keynes.

Beale, E.M.L., J.J.H. Forrest, and C.J. Taylor. 1980. Multi-time Period Stochastic Programming. In Stochastic Programming, M.A.H. Dempster (ed.). Academic Press, New York.

Birge, J.R. 1980. Solution Methods for Stochastic Dynamic Linear Programs. Technical Report SOL 80-29, Systems Optimization Laboratory, Stanford University.

Birge, J.R. 1982. Decomposition and Partitioning Methods for Multi-Stage Stochastic Linear Programs. Technical Report 82-6, Department of Industrial and Operations Engineering, The University of Michigan.

Birge, J.R. 1984. The Relationship Between the L-Shaped Method and Dual Basis Factorization for Stochastic Linear Programming. IIASA Volume.

Birge, J.R. and R.J-B. Wets. 1983. Designing Approximation Schemes for Stochastic Optimization Problems. IIASA Report.

Dantzig, G.B. and A. Madansky. 1961. On the Solution of Two-Stage Linear Programs Under Uncertainty. In Proceedings, 4th Berkeley Symposium on Math. Stat. and Prob. U.C. Press, Berkeley.

Garstka, S. and R. Rutenberg. 1973. Computation in Discrete Stochastic Programming with Recourse. Operations Research 21, 112-122.

Glassey, C.R. 1971. Dynamic Linear Programs for Production Scheduling. Operations Research 19, 45-56.

Glassey, C.R. 1973. Nested Decomposition and Multi-Stage Linear Programs. Management Science 20, 282-292.

Ho, J.K. 1984. Convergence Behavior of Decomposition Algorithms for Linear Programs. Working Paper No. 179, College of Business Administration, The University of Tennessee.

Ho, J.K. and E. Loute. 1981. A Set of Staircase Linear Programming Test Problems. Mathematical Programming 20, 245-250.

Ho, J.K. and A.S. Manne. 1974. Nested Decomposition for Dynamic Models. Mathematical Programming 6, 121-140.

Kall, P. 1979. Computational Methods for Solving Two-Stage Stochastic Linear Programming Problems. Journal of Appl. Math. and Physics 30, 261-271.

Kallberg, J.G. and M.I. Kusy. 1976. A Stochastic Linear Program with Simple Recourse. Faculty of Commerce, The University of British Columbia.

Kallberg, J.G., R.W. White, and W.T. Ziemba. 1982. Short Term Financial Planning Under Uncertainty. Management Science 28, 670-682.

Murtagh, B.A. and M.A. Saunders. 1978. Large Scale Linearly Constrained Optimization. Mathematical Programming 14, 41-72.

Pfefferkorn, C.E. and J.A. Tomlin. 1976. Design of a Linear Programming System for ILLIAC IV. Technical Report SOL 76-8, Systems Optimization Laboratory, Stanford University.

Strazicky, B. 1980. Some Results Concerning an Algorithm for the Discrete Recourse Problem. In Stochastic Programming, M.A.H. Dempster (ed.). Academic Press, New York.

Van Slyke, R. and R. Wets. 1969. L-Shaped Linear Programs with Applications to Optimal Control and Stochastic Linear Programs. SIAM Journal on Appl. Math. 17, 638-663.

Wets, R. 1966. Programming Under Uncertainty: The Solution Set. SIAM Journal on Appl. Math. 14, 1143-1151.

Wets, R. 1983. Stochastic Programming: Solution Techniques and Approximation Schemes. In Mathematical Programming: State of-the-Art 1982, A. Bachem, M. Groetschel, B. Korte (eds.). Springer-Verlag, Berlin.

Appendix

A1. Listing of the FORTRAN code NDST3 for the Nested Decomposition Stochastic Programming Algorithm

```
C ALGORITHM FOR STOCHASTIC PROGRAMS, WHERE
C THE RANDOM VARIABLES CAN BE IN EITHER THE CONSTRAINT
C MATRIX OR THE RIGHT HAND SIDE, AS LONG AS A DISCRETE
C DISTRIBUTION IS ASSUMED.  THE L.P. SECTIONS OF
C THIS CODE ARE TAKEN FROM LPM-1 BY J. TOMLIN
C AND REVISED BY G. KOCHMAN.
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      COMMON/TIMERS/ ITOT
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
C
C         MAIN PROGRAM
C
C         START TIMER
10    CALL TIME(0,0,ITOT)
      CALL INIT
C         INPUT PROBLEM DATA
      CALL INPUT(IFPROB)
      CALL INCHK
      IF (IFPROB .EQ. 0) GO TO 1000
      CALL TIME(1,0,ITOT)
      JTOT=ITOT
      JCUR=1
      DO 100 I=1,3
      JPER(I)=1
100   CONTINUE
      NODE=0
      NPASS=0
      JPASS=1
 11   CONTINUE
      ITSINV = 99999
      NPASS=NPASS+1
      IF(NPER.EQ.1) GO TO 200
      IF(JCUR .EQ. NPER) GO TO 101
      CALL NORMAL(ITSINV)
      IF(JCUR.EQ.NPER+1) GO TO 201
C CHECK FOR CORRECT CUTS
       CALL STRPRT
C      CALL ACHECK
      CALL NDCOM
      GO TO 200
  101 CONTINUE
      CALL PARSFT
  200 CONTINUE
       IF(NPER.EQ.1) JCUR=2
      IF(JCUR .LE. NPER) GO TO 11
201         CONTINUE
      CALL TIME(1,0,ITOT)
      WRITE (6,1) ITOT
1     FORMAT (/' TOTAL SOLUTION TIME =',I6,' MILLISECONDS')
      KTOT=ITOT-JTOT
      WRITE(6,2) KTOT
2     FORMAT(' SOLUTION TIME WITHOUT INPUT = ',I6,' MILLISECONDS')
C         OUTPUT OPTIMAL SOLUTION
      CALL WRAPUP
      GO TO 10
1000  STOP
```

```
      END
C-----------------------------------------------------------------------
      BLOCK DATA
C
C         INITIALIZES GLOBAL PROGRAM CONSTANTS
C         SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C         BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      DATA ZTOLZE/1.E-7/,ZTOLPV/1.E-5/,ZTCOST/1.E-4/,ZTOLSM/1.E-10/
      DATA NRMAX/350/,NTMAX/1000/,NEMAX/3000/,NEGINF/-100000/
      DATA QBL/'    '/,QA/'A   '/,QI/'I   '/,QF/'F   '/,QN/'N   '/,
     1  QB/'B   '/,QC/'C   '/,QE/'E   '/,QH/'H   '/,
     2  QL/'L   '/,QO/'O   '/,QR/'R   '/,QM/'M   '/,QG/'G   '/,
     3  QS/'S   '/,QP/'P   '/
      END
C-----------------------------------------------------------------------
       SUBROUTINE RHCHCK
C
C THIS SUBROUTINE CHECKS THE ROW RESIDUALS
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2,3,3,1),ICN(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BL5/DRES,ICNAM
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      JC = JCUR
      J2=JPER(2)
      J3 = JPER(3)
      KNR=NROW(JC,J2,J3)
      KNC=NCOL(JC,J2,J3)
      DO 90 I=1,KNR
      YTEMP(I)=B(I,JC,J2,J3)
      YTEMP1(I)=0.
```

```
90      CONTINUE
C       WRITE(6,91)
C1      FORMAT(' CHECKING ROW RESIDUAL')
        DMAX=O.
         DO 200 J=1,KNC
        CALL UNPACK(J)
        KB=KINBAS(J,JC,J2,J3)
        IF(KB.GT.O) GO TO 102
        DO 100 I=1,KNR
        IF(KB.LT.O) GO TO 101
        YTEMP(I)=YTEMP(I) - Y(I)*XLB(J,JC,J2,J3)
        GO TO 100
101     CONTINUE
        YTEMP(I) = YTEMP(I) - Y(I)*XUB(J,JC,J2.J3)
100     CONTINUE
        GO TO 200
102     CONTINUE
         DO 103 I=1,KNR
        YTEMP1(I) = YTEMP1(I) + Y(I)*X(KB,JC,J2,J3)
103     CONTINUE
200     CONTINUE
        DO 300 I=1,KNR
        DF = DABS(YTEMP(I)-YTEMP1(I))
        IF(DF.GT.DMAX) DMAX = DF
         IF(DF.LE.ZTOLZE) GO TO 300
300     CONTINUE
C       WRITE(6,302) DMAX
CO2     FORMAT(' DMAX = ',E9.2)
        DRES = DMAX
         RETURN
         END
C-----------------------------------------------------------------
        SUBROUTINE INPUT(IFPROB)
C
C          INPUTS PROBLEM DATA
C          SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C          BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C          ***DESCRIPTION OF PARAMETERS***
C          IFPROB = NONZERO PROBLEM ID NUMBER(OUTPUT)
C
        IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
       1  INTEGER*4 (I-N,Q)
        INTEGER JH,KINBAS,LA,LE,IA,IE.NODE
        INTEGER ICNAM(602,2,3,3,1),ICN(602,2),NAME(6)
        DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
        REAL A(3000,3,3,1)
C
        COMMON/BL5/DRES,ICNAM
        COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
       1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
        COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
       1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
       1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
       2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
       2  ),NETA,
       3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
       3  602,3,3,1),
       4  LE(1002),IA(3000,3,3,1),IE(3000),
```

```
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C         ***DESCRIPTIONS OF SOME IMPORTANT VARIABLES IN BLANK COMMON***
C         B(I) = RIGHT HAND SIDE OF ROW I
C         X(I) = LP VALUE FOR JH(I), WHICH IS THE VARIABLE BASIC IN ROW I
C         A CONTAINS THE NONZERO ELEMENTS OF THE CONSTRAINT MATRIX, INCL.
C           THE OBJECTIVE ROW IOBJ. LA(J) = LOCATION IN A OF THE FIRST
C           ELEMENT OF COL J. IA(I) = ROW IN WHICH ELEMENT I OF A BELONGS.
C         E CONTAINS THE NONZERO ELEMENTS OF THE CURRENT LP BASIS INVERSE
C           IN ETA VECTOR FORM. LE, IE ARE TO E AS LA, IA ARE TO A.
C         MSTAT FLAGS FEASIBILITY OF CURRENT LP
C         ITCNT = NO. OF SIMPLEX ITERATIONS SO FAR; IF > ITRFRQ, STOP
C         INVFRQ = NUMBER OF SIMPLEX ITERATIONS BEFORE E IS REINVERTED
C         NROW = NO. OF ROWS; NCOL = NO. OF COLUMNS
C         KINBAS(J) = { I IF J IS BASIC IN ROW I, I.E. J = JH(I)
C                       O IF J IS NONBASIC AT ITS LOWER BOUND XLB(J)
C                      -1 IF J IS NONBASIC AT ITS UPPER BOUND XUB(J) }
      DO 21 K=1,3
      DO 21 K1=1,2
      DO 21 K2=1,2
      DO 10 I=1,NRMAX
      XKSI(I,K,K1,K2)=0.0
10       B(I,K,K1,K2) = 0.
      XKSI(I,K,K1,K2) = 0.0
      NROW(K,K1,K2) = 0
      L = 500
      DO 20 J=1,L
20       KINBAS(J,K,K1,K2) = 0
 21   CONTINUE
      ITCNT = 0
      ICS1=0
      ICS2=0
C  SET FOR MXNST POSSIBILITIES
      MXNST=5
      READ (5,7000,END=9999) IFPROB,IOBJ,INVFRQ,ITRFRQ,NPER
7000  FORMAT (5I4)
      DO 7002 I=1,NPER
      READ(5,7001,END=9999) NND(I)
7001  FORMAT(I4)
7002  CONTINUE
      IF (IFPROB .EQ. 0) RETURN
      IF (IOBJ .EQ. 0)IOBJ = 1
      IF (INVFRQ .EQ. 0)INVFRQ = 99999
      IF (ITRFRQ .EQ. 0)ITRFRQ = 99999
      WRITE(6,8010) IFPROB
8010  FORMAT(' PROBLEM ',I4)
C
C START TO ITERATE
C
      JCUR=1
      DO 19 I=1,3
```

```
      JPER(I)=1
 19   CONTINUE
      JPER(1)=0
 22   CONTINUE
      NR=0
      JPER(JCUR)=JPER(JCUR)+1
      IF(JPER(JCUR) .LE. NND(JCUR)) GO TO 190
      K=JCUR-1
187   CONTINUE
      IF(K .LE. 1) GO TO 188
      IF(JPER(K) .GE. NND(K)) K=K-1
      IF(JPER(K) .GE. NND(K)) GO TO 187
      JPER(K)=JPER(K)+1
      GO TO 186
188   CONTINUE
      IF(JCUR .EQ. NPER) GO TO 9998
      JCUR=JCUR+1
186   CONTINUE
      KK=K+1
      IF(K .EQ. 0) K=1
      DO 189 I=KK,JCUR
      JPER(I)=1
189   CONTINUE
190   CONTINUE
      JC=JCUR
      J1=JPER(1)
      J2=JPER(2)
      J3=JPER(3)
      NROW(JC,J2,J3)=0
      NCOL(JC,J2,J3)=0
      NELM(JC,J2,J3)=0
      READ(5,222) PROB(JC,J2,J3)
 222  FORMAT(F5.3)
5     READ(5,101) K1,K2,K3,K4,(NAME(I),I=1,4),ATEMP1,NAME(5),NAME(6),
     1  ATEMP2
101   FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
      IF(K1 .EQ. QE) GO TO 600
      IF(K1 .EQ. QBL) GO TO 50
      IF(K1 .EQ. QN) GO TO 5
      IF(K1 .EQ. QR .AND. K2 .EQ. QO) L=1
      IF(K1 .EQ. QR .AND. K2 .EQ. QO) GO TO 5
      IF(K1 .EQ. QC) L=2
      IF(K1 .EQ. QC) GO TO 5
      IF(K1 .EQ. QB .AND. K2 .EQ. QA) L=3
      IF(K1 .EQ. QB .AND. K2 .EQ. QA) GO TO 5
      IF(K1 .EQ. QR .AND. K2 .EQ. QH) L=4
      IF(K1 .EQ. QR .AND. K2 .EQ. QH) GO TO 5
50    GO TO(210,320,410,500),L
C
210   NROW(JC,J2,J3)=NROW(JC,J2,J3)+1
      NCOL(JC,J2,J3)=NROW(JC,J2,J3)
      ICNAM(NROW(JC,J2,J3),1,JC,J2,J3) = NAME(1)
      ICNAM(NROW(JC,J2,J3),2,JC,J2,J3) = NAME(2)
C
C        TEST ROW TYPE
C
      IF(K2.EQ.QL .OR. K3.EQ.QL)  GO TO 220
      IF(K2.EQ.QE .OR. K3.EQ.QE)  GO TO 230
```

```
      IF(K2.EQ.QG .OR. K3.EQ.QG)  GO TO 240
      IF(K2.EQ.QN .OR. K3.EQ.QN)  GO TO 250
      GO TO 230
220   XLB(NROW(JC,J2,J3),JC,J2,J3) = O.
      XUB(NROW(JC,J2,J3),JC,J2,J3) = 1.E7
      GO TO 250
230   XLB(NROW(JC,J2,J3),JC,J2,J3) = O.
      XUB(NROW(JC,J2,J3),JC,J2,J3) = O.
      GO TO 250
240   XLB(NROW(JC,J2,J3),JC,J2,J3) = O.
      XUB(NROW(JC,J2,J3),JC,J2,J3) = 1.E7
      A(NROW(JC,J2,J3),JC,J2,J3) = -1.
      GO TO 260
250   A(NROW(JC,J2,J3),JC,J2,J3) = 1.
260   IA(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
      LA(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
      JH(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
      KINBAS(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
      NELEM=NROW(JC,J2,J3)
      GO TO 5
C
C        MATRIX ELEMENTS
C
320   J = 3
      K = 4
      IF (DABS(ATEMP1) .LE. ZTOLZE)  GO TO 321
      GO TO 324
321   J=5
      K=6
      IF (DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
      ATEMP1=ATEMP2
324   IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2)  GO TO 330
      NCOL(JC,J2,J3) = NCOL(JC,J2,J3) + 1
      ICS1 = NAME(1)
      ICS2 = NAME(2)
      ICNAM(NCOL(JC,J2,J3),1,JC,J2,J3) = ICS1
      ICNAM(NCOL(JC,J2,J3),2,JC,J2,J3) = ICS2
      LA(NCOL(JC,J2,J3),JC,J2,J3) = NELEM + 1
C
C        TEST FOR ROW MATCH
C
 330  KNR=NROW(JC,J2,J3)
      DO 340 I=1,KNR
      IF(NAME(J).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(K).NE.
     1 ICNAM(I,2,JC,J2,J3)) GO TO 340
         NELEM = NELEM + 1
         IA(NELEM,JC,J2,J3) = I
         A(NELEM,JC,J2,J3) = ATEMP1
         LA(NCOL(JC,J2,J3)+1,JC,J2,J3)=NELEM+1
335      IF(K .GT. 5)  GO TO 5
         IF(DABS(ATEMP2) .LE. ZTOLZE)  GO TO 5
         J = 5
         K = 6
         ATEMP1 = ATEMP2
         GO TO 330
340      CONTINUE
      WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
8300  FORMAT(17HNO MATCH FOR ROW ,2A4,10HAT COLUMN ,2A4)
```

```
      STOP
C
C        BASIS CARDS
C
410   KNC=NCOL(JC,J2,J3)
      DO 420 I=1,KNC
      IF(NAME(1).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(2).NE.
     1 ICNAM(I,2,JC,J2,J3)) GO TO 420
         IBVEC = I
         GO TO 425
420      CONTINUE
      WRITE(6,8400) NAME(1),NAME(2)
8400  FORMAT(20HNO MATCH FOR VECTOR ,2A4)
      GO TO 5
425   KNR=NROW(JC,J2,J3)
      DO 430 I=1,KNR
      IF(NAME(3).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(4).NE.
     1 ICNAM(I,2,JC,J2,J3)) GO TO 430
         IBROW=I
         GO TO 440
430      CONTINUE
      WRITE(6,8300) NAME(3),NAME(4)
      GO TO 5
440   JH(IBROW,JC,J2,J3) = IBVEC
      KINBAS(IBROW,JC,J2,J3) = 0
      KINBAS(IBVEC,JC,J2,J3) = IBROW
      GO TO 5
C
C        RHS
C
500   J = 3
      K = 4
      IF (DABS(ATEMP1) .LE. ZTOLZE)  GO TO 521
      GO TO 530
521   J=5
      K=6
      IF (DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
      ATEMP1=ATEMP2
C
C        TEST FOR ROW MATCH
C
530   KNR=NROW(JC,J2,J3)
      DO 540 I=1,KNR
      IF(NAME(J).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(K).NE.
     1 ICNAM(I,2,JC,J2,J3)) GO TO 540
         B(I,JC,J2,J3) = ATEMP1
         XKSI(I,JC,J2,J3) = ATEMP1
535      IF(K .GT. 5)  GO TO 5
         IF(DABS(ATEMP2) .LE. ZTOLZE)  GO TO 5
         J = 5
         K = 6
         ATEMP1 = ATEMP2
         GO TO 530
540      CONTINUE
      WRITE(6,8300) NAME(J),NAME(K)
      STOP
C
C          END OF INPUT
```

```
C
600   NSCOL = NCOL(JC,J2,J3) - NROW(JC,J2,J3)
      K = NROW(JC,J2,J3) + 1
C        INPUT LOWER AND UPPER BOUNDS ON DECISION VARIABLES
      KNC=NCOL(JC,J2,J3)
      READ (5,650) (XLB(J,JC,J2,J3), J=K,KNC)
      READ (5,650) (XUB(J,JC,J2,J3), J=K,KNC)
650   FORMAT (9F8.0)
      NELM(JC,J2,J3)=NELEM
      NELEM = NELEM - NROW(JC,J2,J3)
      RELEM = NELEM
      RDENS = RELEM/(NROW(JC,J2,J3)*NSCOL)
      WRITE (6,8500) RDENS
8500  FORMAT (' DENSITY OF CONSTRAINT MATRIX IS ',F6.3)
      NCOLP(JC,J2,J3)=NCOL(JC,J2,J3)
      NROWP(JC,J2,J3)=NROW(JC,J2,J3)
C ADD NROW VALUES FOR NPER IN LKHDCT
      IF(JC.NE.NPER) GO TO 8499
      JJ2=NND(2)
      DO 8498 I=2,JJ2
      NROW(3,I,1)=NROW(3,1,1)
      NCOL(3,I,1)=NCOL(3,1,1)
8498  CONTINUE
8499  CONTINUE
      IF(JC .EQ. NPER) GO TO 2222
C
C INPUT FOR THE B MATRIX FOLLOWS
C
      NELEM=0
15     READ(5,1101) K1,K2,K3,K4,(NAME(I),I=1,4),ATEMP1,NAME(5),NAME(6),
     1  ATEMP2
1101   FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
      IF(K1 .EQ. QE) GO TO 1600
      IF(K1 .EQ. QBL) GO TO 150
      IF(K1 .EQ. QN) GO TO 15
      IF(K1 .EQ. QR .AND. K2 .EQ. Q0) L=1
      IF(K1 .EQ. QR .AND. K2 .EQ. Q0) GO TO 15
      IF(K1 .EQ. QC) L=2
      IF(K1 .EQ. QC) GO TO 15
150   GO TO (115,1320),L
C
C        MATRIX ELEMENTS
C
 115  NR=NR+1
      ICN(NR,1)=NAME(1)
      ICN(NR,2)=NAME(2)
      IF(NR.GT.NROW(JC,J2,J3)) GO TO 15
      ABN(NR,JC,J2,J3) = 0.
      IBN(NR,JC,J2,J3) = NR
      LBN(NR,JC,J2,J3) = NR
      NELEM=NR
      NC=NR
      GO TO 15
1320   J = 3
      K = 4
      GO TO 1324
1321   J=5
      K=6
```

```
      ATEMP1=ATEMP2
1324   IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2)  GO TO 1330
      NC = NC + 1
      ICS1 = NAME(1)
      ICS2 = NAME(2)
      LBN(NC,JC,J2,J3) = NELEM + 1
C
C        TEST FOR ROW MATCH
C
1330   DO 1340 I = 1,NR
         IF(NAME(J).NE.ICN(I,1) .OR. NAME(K).NE.ICN(I,2))GO TO 1340
         NELEM = NELEM + 1
         IBN(NELEM,JC,J2,J3) = I
         ABN(NELEM,JC,J2,J3) = ATEMP1
         LBN(NC+1,JC,J2,J3)=NELEM+1
1335      IF(K .GT. 5)  GO TO 15
         IF(DABS(ATEMP2) .LE. ZTOLZE)  GO TO 15
         J = 5
         K = 6
         ATEMP1 = ATEMP2
         GO TO 1330
1340      CONTINUE
      WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
1600  CONTINUE
      GO TO 22
2222  CONTINUE
C
C             HERE WE HAVE NEW POSSIBILITIES
C
C                    INITIALIZE
C
      NST=0
      DO 2102 I=1, MXNST
      INST(I)=0
2102  CONTINUE
C
C             READ IN VALUES
C
25    READ(5,2101) K1,K2,K3,K4,(NAME(I), I=1,4),ATEMP1,
     1 NAME(5),NAME(6),ATEMP2
2101  FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
      IF(K1 .EQ. QS) L=1
      IF(K1 .EQ. QS) GO TO 2505
      IF(K1 .EQ. QE) GO TO 2600
C
C             CHECK FOR NEW VALUES
C
      IF(NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 2330
      ICS1=NAME(1)
      ICS2=NAME(2)
      NST=NST+1
      DO 2340 I=1,KNR
      IF(NAME(1).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(2).NE.
     1  ICNAM(I,2,JC,J2,J3)) GO TO 2340
      IBST(NST)=I
      INST(NST)=INST(NST) + 1
      CBST(NST,INST(NST))=ATEMP1
      PRBV(NST,INST(NST))=ATEMP2
```

```
      GO TO 2505
 2340 CONTINUE
       GO TO 2349
2330  CONTINUE
      INST(NST)=INST(NST)+1
      CBST(NST,INST(NST))=ATEMP1
      PRBV(NST,INST(NST))=ATEMP2
      GO TO 2505
C
C              CHECK FOR ROW MATCH
C
C
2349   CONTINUE
C               NO MATCH
C
      WRITE(6,2350) NAME(3),NAME(4),NAME(1),NAME(2)
 2350 FORMAT(17HNO MATHCH FOR ROW,2A4,18HAT STOCH VARIABLE,2A4)
      STOP
C
C               SET UP RHS
C
 2600 CONTINUE
      DO 2360 I=1,KNR
      BND(I)=XKSI(I,NPER,1,1)
 2360 CONTINUE
C
C               CHECK IF MXNST>NST
C
      IF(NST .EQ. 3) GO TO 2376
      NNST=3
      IF(NST.LT.3) NNST=NST+1
      DO 2375 I=NNST,3
      INST(I)=1
      PRBV(I,1)=1.0
 2375 CONTINUE
 2376 CONTINUE
      INS=INST(1)
      JNS=INST(2)
      KNS=INST(3)
      DO 2380 I=1,INS
      DO 2380 J=1,JNS
      DO 2380 K=1,KNS
      PRST(I,J,K)=PRBV(1,I)*PRBV(2,J)*PRBV(3,K)
 2380 CONTINUE
      WRITE(6,2400)
2400  FORMAT(/,' STOCHASTIC VARIABLES')
      WRITE(6,2401)
2401  FORMAT(/,' ROW',8X,' VALUE',4X,' PROB',/)
      DO 2410 I=1,3
      IF(I.GT.NST) GO TO 2410
      J1 = INST(I)
      DO 2410 J=1,J1
      I1=IBST(I)
      A1=CBST(I,J)
      A2=PRBV(I,J)
      WRITE(6,2402) ICNAM(I1,1,NPER,1,1),ICNAM(I1,2,NPER,1,1),A1,A2
2402  FORMAT(1X,2A4,3X,F8.2,2X,F8.6)
2410  CONTINUE
```

```
9998  CONTINUE
C SET FOR NO THETA CUTS TO START
      DO 1601 I=1,NPER
      NNDD=NND(I)
      DO 1602 J=1,NNDD
      NTH(I,J,J)=0
1602  CONTINUE
1601  CONTINUE
        RETURN
9999  IFPROB = 0
      RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE INCHK
C
C CHECKS OUT THE CURRENT VALUES
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      L=1
      DO 100 J=1,NPER
      NNDD=NND(J)
      DO 99 K=1,NNDD
      IF((J .EQ. 1) .AND. (K .GT. 1)) GO TO 99
      NNPP=NND(3)
      DO 80 L=1, NNPP
      IF((J .NE. 3) .AND. (L .GT.1)) GO TO 80
      KLM=NELM(J,K,L)
      KNC=NCOL(J,K,L)
      KNR=NROW(J,K,L)
      WRITE (6,1) J,K,L
 1    FORMAT(' CHECK THE A MATRIX OF NODE ',I4,',',I4,',',I4,//)
      WRITE (6,2) (A(I,J,K,L),I=1,KLM)
 2    FORMAT(15F7.2)
      WRITE(6,3)
 3    FORMAT(//,' IA')
      WRITE(6,5) (IA(I,J,K,L),I=1,KLM)
```

```
 5    FORMAT(15I5)
      WRITE(6,4)
 4    FORMAT(//,' LA')
      WRITE(6,5) (LA(I,J,K,L),I=1,KNC)
      WRITE(6,6)KNR,KNC,KLM
 6    FORMAT(//,' NROW=',I5,' NCOL=',I5,' NELM=',I5)
      WRITE(6,7)
 7    FORMAT(' KINBAS')
      WRITE(6,5)(KINBAS(I,J,K,L),I=1,KNC)
      WRITE(6,8)
 8    FORMAT(' JH')
      WRITE(6,5) (JH(I,J,K,L),I=1,KNR)
      WRITE(6,9)
 9    FORMAT(' X VALUES')
      WRITE(6,2)(X(I,J,K,L),I=1,KNR)
      WRITE(6,10)
 10   FORMAT(' XKSI"S ')
      WRITE(6,2) (XKSI(I,J,K,L),I=1,KNR)
      WRITE(6,2) (B(I,J,K,L),I=1,KNR)
      WRITE(6,11) J,K,L,PROB(J,K,L)
 11   FORMAT(' PROB OF',I5,I5,I5,' IS',F7.5)
      WRITE(6,12)
 12   FORMAT(' ABN MATRIX')
      WRITE(6,2)(ABN(I,J,K,L),I=1,KLM)
      WRITE(6,13)
 13   FORMAT(' IBN')
      WRITE(6,5)(IBN(I,J,K,L),I=1,KLM)
      WRITE(6,14)
 14   FORMAT(' LBN')
      WRITE(6,5)(LBN(I,J,K,L),I=1,KNC)
 80   CONTINUE
 99   CONTINUE
 100  CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE FTRAN(IPAR)
C
C        PERFORMS FORWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH. STANFORD UNIVERSITY)
C        ***DESCRIPTION OF PARAMETERS***
C        IPAR = PARAMETER INDICATING WHICH ETA-VECTORS MATRIX COLUMN
C               IS TO BE UPDATED BY (INPUT)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
```

```
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      NLE = NETA
      NFE = 1
      IF (IPAR.EQ.2) NFE = NLETA + 1
      IF (NFE .GT. NLE) GO TO 9000
      DO 1000 IK = NFE,NLE
         LL = LE(IK)
         KK = LE(IK+1) - 1
         IPIV = IE(LL)
         DY = Y(IPIV)
         DY = DY/E(LL)
         Y(IPIV) = DY
         IF (KK .LE. LL) GO TO 1000
         LL = LL + 1
         DO 500 J = LL,KK
            IR = IE(J)
            Y(IR) = Y(IR) - E(J) * DY
500         CONTINUE
1000     CONTINUE
9000  CONTINUE
      RETURN
      END
C------------------------------------------------------------------------
      SUBROUTINE BTRAN
C
C        PERFORMS BACKWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT.
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
```

```
C
       IF (NETA .LE. 0) GO TO 9000
       DO 1000 I = 1,NETA
          IK = NETA - I + 1
          LL = LE(IK)
          KK = LE(IK+1) - 1
          IPIV = IE(LL)
          DP = E(LL)
          DY = Y(IPIV)
          DSUM = 0.
          IF (KK .LE. LL) GO TO 600
          LL = LL + 1
          DO 500 J = LL,KK
             IR = IE(J)
             DE = E(J)
             DPROD = DE * Y(IR)
             DSUM = DSUM + DPROD
500          CONTINUE
C
600       Y(IPIV) = (DY - DSUM) / DP
1000      CONTINUE
       JC=JCUR
       J2=JPER(2)
       J3=JPER(3)
       KNR=NROW(JC,J2,J3)
       DO 700 I=1,KNR
       YPI(I,JC,J2,J3)=Y(I)
 700   CONTINUE
9000   RETURN
       END
C-----------------------------------------------------------------------
       SUBROUTINE FORMC
C
C         FORMS OBJ. FUNCTION VECTOR AND CHECKS FEASIBILITY OF BASIS
C         SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C         BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
       IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
      1  INTEGER*4 (I-N,Q)
       INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
       DOUBLE PRECISION E(3000)
       REAL A(3000,3,3,1)
       COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
      1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
       COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
       COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
      1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
      1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
      2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
      2  ),NETA,
      3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
      3  602,3,3,1),
      4  LE(1002),IA(3000,3,3,1),IE(3000),
      5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
       COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
```

```
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNSI,NST
C
      NINF=0
      MSTAT = QF
      Y(IOBJ) = 0.
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNR=NROW(JC,J2,J3)
      DO 30 I=1,KNR
         IF (I .EQ. IOBJ) GO TO 30
         ICOL = JH(I,JC,J2,J3)
         IF (X(I,JC,J2,J3) .LE. (XLB(ICOL,JC,J2,J3) - ZTOLZE)) GO TO 10
         IF (X(I,JC,J2,J3) .GE. (XUB(ICOL,JC,J2,J3) + ZTOLZE)) GO TO 20
         Y(I) = 0.
         GO TO 30
10       Y(I) = 1.
         MSTAT = QI
      NINF=NINF+1
         GO TO 30
20       Y(I) = -1.
      NINF=NINF+1
         MSTAT = QI
30       CONTINUE
      IF (MSTAT.EQ.QF) Y(IOBJ) = 1.
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE PRICE
C
C        PRICES OUT NONBASIC COLUMNS; CHOOSES PIVOT COLUMN JCOLP FOR
C        CURRENT PRIMAL SIMPLEX ITERATION
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
```

```
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      CMIN = 1.E10
      NOPT=0
      CMAX = -1.E10
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNC=NCOL(JC,J2,J3)
      DO 1000 J=1,KNC
         IF (KINBAS(J,JC,J2,J3) .GT. 0) GO TO 1000
         IF ((XUB(J,JC,J2,J3) - XLB(J,JC,J2,J3)) .LT. ZTOLZE) GO TO 1000
         DSUM = 0.
         LL = LA(J,JC,J2,J3)
         KK = LA(J+1,JC,J2,J3) - 1
         DO 500 I = LL,KK
            IR = IA(I,JC,J2,J3)
            DE = A(I,JC,J2,J3)
            DPROD = DE * Y(IR)
            DSUM = DSUM + DPROD
500         CONTINUE
         IF (KINBAS(J,JC,J2,J3) .EQ. -1) GO TO 600
      IF(DSUM.LT.-ZTOLZE)NOPT=NOPT+1
         IF (DSUM .GE. CMIN) GO TO 1000
         CMIN = DSUM
         JCOL1 = J
         GO TO 1000
 600  IF(DSUM.GT.ZTOLZE)NOPT=NOPT+1
         IF (DSUM .LE. CMAX) GO TO 1000
         CMAX = DSUM
         JCOL2 = J
1000     CONTINUE
C
      IF (CMIN .LE. -ZTCOST) GO TO 1500
      IF (CMAX .GE. ZTCOST) GO TO 2000
      JCOLP = 0
      RETURN
1500  IF (CMAX .GE. ZTCOST) GO TO 2500
1600  JCOLP = JCOL1
      RETURN
2000  JCOLP = JCOL2
      RETURN
2500  IF (ABS(CMIN) - CMAX) 2000,2000,1600
      END
C-----------------------------------------------------------------------
      SUBROUTINE CHUZR
C
C        PERFORMS MIN-RATIO TEST FOR PIVOT COLUMN JCOLP DETERMINED IN
C        SUBROUTINE PRICE.  SELECTS PIVOT ROW IROWP FOR CURRENT PRIMAL
C        SIMPLEX ITERATION.
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
```

```
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,MINF,NOPT
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      IF (KINBAS(JCOLP,JC,J2,J3) .EQ. -1) GO TO 1000
C
C              INCOMING VARIABLE AT LOWER BOUND
C
      DP = 1.E10
      KNR=NROW(JC,J2,J3)
      DO 500 I=1,KNR
         IF (I .EQ. IOBJ) GO TO 500
         ICOL = JH(I,JC,J2,J3)
         IF (Y(I) .GT. ZTOLPV) GO TO 100
         IF (Y(I) .LT. -ZTOLPV) GO TO 200
         GO TO 500
C              POSITIVE COEFFICIENT
100      IF (X(I,JC,J2,J3) .LT. (XLB(ICOL,JC,J2,J3) - ZTOLZE))
     1    GO TO 500
         DE = (X(I,JC,J2,J3) - XLB(ICOL,JC,J2,J3))/Y(I)
         IF (DE .GE. DP) GO TO 500
         IPTYPE = 0
         GO TO 250
C              NEGATIVE COEFFICIENT
200      IF (X(I,JC,J2,J3) .GT. (XUB(ICOL,JC,J2,J3) + ZTOLZE))
     1    GO TO 500
         DE = (X(I,JC,J2,J3) - XUB(ICOL,JC,J2,J3))/Y(I)
         IF (DE .GE. DP) GO TO 500
         IPTYPE = -1
250      DP = DE
      APV=Y(I)
         IROWP = I
500      CONTINUE
      DE = DP + XLB(JCOLP,JC,J2,J3)
      IF (DE .LT. XUB(JCOLP,JC,J2,J3)) GO TO 600
```

```
      DP = XUB(JCOLP,JC,J2,J3) - XLB(JCOLP,JC,J2,J3)
      NPIVOT = 0
      RETURN
600   NPIVOT = 1
      RETURN
C
C   INCOMING VARIABLE AT UPPER BOUND
C
1000  DP = -1.E10
      KNR=NROW(JC,J2,J3)
      DO 1500 I=1,KNR
         IF (I .EQ. IOBJ) GO TO 1500
         ICOL = JH(I,JC,J2,J3)
         IF (Y(I) .GT. ZTOLPV) GO TO 1100
         IF (Y(I) .LT. -ZTOLPV) GO TO 1200
         GO TO 1500
C              POSITIVE COEFFICIENT
1100     IF (X(I,JC,J2,J3) .GT. (XUB(ICOL,JC,J2,J3) + ZTOLZE))
     1    GO TO 1500
         DE = (X(I,JC,J2,J3) - XUB(ICOL,JC,J2,J3))/Y(I)
         IF (DE .LE. DP) GO TO 1500
         IPTYPE = -1
         GO TO 1250
C              NEGATIVE COEFFICIENT
1200     IF (X(I,JC,J2,J3) .LT. (XLB(ICOL,JC,J2,J3) - ZTOLZE))
     1    GO TO 1500
         DE = (X(I,JC,J2,J3) - XLB(ICOL,JC,J2,J3))/Y(I)
         IF (DE .LE. DP) GO TO 1500
         IPTYPE = 0
1250     DP = DE
         IROWP = I
      APV=Y(I)
1500     CONTINUE
      DE = DP + XUB(JCOLP,JC,J2,J3)
      IF (DE .GT. XLB(JCOLP,JC,J2,J3)) GO TO 1600
      DP = XLB(JCOLP,JC,J2,J3) - XUB(JCOLP,JC,J2,J3)
      NPIVOT = 0
      RETURN
1600  NPIVOT = 1
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE WRETA
C
C        FORMS NEW ETA-VECTORS FOR PRODUCT FORM OF BASIS INVERSE
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YFI(350,3,3,1)
```

```
     1   ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2   INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2   ),NETA,
     3   NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3   602,3,3,1),
     4   LE(1002),IA(3000,3,3,1),IE(3000),
     5   ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6   PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7   LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1   PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2   NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      NELEM = NELEM + 1
      IE(NELEM) = IROWP
      E(NELEM) = Y(IROWP)
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNR=NROW(JC,J2,J3)
      DO 1000 I = 1,KNR
         IF (I .EQ. IROWP) GO TO 1000
         IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 1000
         NELEM = NELEM + 1
         IE(NELEM) = I
         E(NELEM) = Y(I)
 1000    CONTINUE
C
      NETA = NETA + 1
      LE(NETA+1) = NELEM + 1
      RETURN
      END
C------------------------------------------------------------------------
      SUBROUTINE SHIFTR(IOLD,INEW)
C
C        REARRANGES DATA STORAGE
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C        ***DESCRIPTION OF PARAMETERS***
C        IOLD,INEW = PARAMETERS INDEXING STORAGE LOCATIONS IN WHICH
C                    DATA IS TO BE TRANSFERRED (INPUT)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1   INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1   NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1   22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1   ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2   INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2   ),NETA,
```

```
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      DIMENSION BARRAY(11900)
      EQUIVALENCE (BARRAY(1),B(1,1,1,1))
      IF(IOLD.EQ.1) IFO=350*(JCUR-1)+1050*(JPER(2)-1)+3150*(JPER(3)-1)
      IF(INEW.EQ.1) IFN=350*(JCUR-1)+1050*(JPER(2)-1)+3150*(JPER(3)-1)
      IF(IOLD.EQ.2) IFO=3150+350*(JCUR-1)+1050*(JPER(2)-1)
     1  +3150*(JPER(3)-1)
      IF(INEW.EQ.2) IFN=3150+350*(JCUR-1)+1050*(JPER(2)-1)
     1  +3150*(JPER(3)-1)
      IF(IOLD.EQ.3) IFO=6300
      IF(INEW.EQ.3) IFN=6300
      IF(IOLD.EQ.4) IFO=6650
      IF(INEW.EQ.4) IFN=6650
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNR=NROW(JC,J2,J3)
      DO 1000 I = 1,KNR
         BARRAY(IFN + I) = BARRAY(IFO + I)
1000     CONTINUE
      RETURN
      END
C----------------------------------------------------------------------
      SUBROUTINE INVERT
C
C        COMPUTES INVERSE OF CURRENT BASIS BY LU DECOMPOSITION
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
```

A2. Listing of input data for test problem SCAGR7.S2

```
   1   1  501000   2
   1
   1
1.000
ROWS                                                                          12000050
 N  FOB00001                                                                  12000060
 E  ROW00001                                                                  12000070
 E  ROW00002                                                                  12000080
 E  ROW00003                                                                  12000090
 E  ROW00004                                                                  12000100
 E  ROW00005                                                                  12000110
 L  ROW00006                                                                  12000120
 E  ROW00007                                                                  12000130
 L  ROW00008                                                                  12000140
 G  ROW00009                                                                  12000150
 E  ROW00010                                                                  12000160
 E  ROW00011                                                                  12000170
 E  ROW00012                                                                  12000180
 E  ROW00013                                                                  12000190
 E  ROW00014                                                                  12000200
 E  ROW00015                                                                  12000210
COLUMNS                                                                       12001360
    COL00001  FOB00001    -35.000000   ROW00001      1.000000                 12001370
    COL00002  FOB00001     54.900000   ROW00001      1.000000                 12001380
    COL00002  ROW00003      1.000000   ROW00005      -.400000                 12001390
    COL00002  ROW00015      -.500000                                          12001400
    COL00003  FOB00001     54.900000   ROW00002      1.000000                 12001410
    COL00003  ROW00004      1.000000   ROW00005      -.400000                 12001420
    COL00003  ROW00015      -.500000                                          12001430
    COL00004  FOB00001    -35.000000   ROW00002      1.000000                 12001440
    COL00005  FOB00001     23.500000   ROW00003     -1.000000                 12001450
    COL00005  ROW00005     -1.700000   ROW00010      1.000000                 12001460
    COL00005  ROW00013      -.245000   ROW00015     -1.400000                 12001470
    COL00006  FOB00001     23.500000   ROW00004     -1.000000                 12001480
    COL00006  ROW00005     -1.700000   ROW00011      1.000000                 12001490
    COL00006  ROW00013      -.245000   ROW00015     -1.400000                 12001500
    COL00007  FOB00001      8.720000   ROW00007      1.000000                 12001510
    COL00007  ROW00013       .200000                                          12001520
    COL00008  FOB00001      9.720000   ROW00005      1.500000                 12001530
    COL00008  ROW00006      1.000000   ROW00007      1.000000                 12001540
    COL00009  FOB00001      6.740000   ROW00007      1.000000                 12001550
    COL00009  ROW00009      1.000000   ROW00014      1.200000                 12001560
    COL00010  FOB00001      6.840000   ROW00007      1.000000                 12001570
    COL00010  ROW00009      1.000000   ROW00015      1.000000                 12001580
    COL00011  FOB00001     15.000000   ROW00005      1.000000                 12001590
    COL00012  FOB00001     22.500000   ROW00015      1.000000                 12001600
    COL00013  ROW00007      1.000000   ROW00008      1.000000                 12001610
    COL00014  FOB00001   -500.000000   ROW00010     -1.000000                 12001620
    COL00015  FOB00001     18.700000   ROW00010     -1.000000                 12001640
    COL00016  FOB00001   -258.300000   ROW00011     -1.000000                 12001670
    COL00017  FOB00001   -662.000000   ROW00012     -1.000000                 12001690
    COL00018  FOB00001      3.000000   ROW00013     -1.000000                 12001740
    COL00019  FOB00001       .390000   ROW00014     -1.000000                 12001760
    COL00020  FOB00001       .470000   ROW00015     -1.000000                 12001780
RHS
    RHS       ROW00001    158.000000   ROW00002    158.000000                 12004290
    RHS       ROW00005   3092.960000   ROW00006   2566.670000                 12004300
    RHS       ROW00007   6900.000000   ROW00008   1600.000000                 12004310
```

```
    RHS       ROW00009    800.000000   ROW00012   -375.200000           12004320
    RHS       ROW00013    -92.120000   ROW00014   -684.000000           12004330
    RHS       ROW00015    -150.0
ENDATA



99999999999999999999999999999999999999999999999999999999999999999999999999999
999999999999999999999999999999999999999999999999999999999999999999999999999999
9999999999999999999999999999999999999999999999999999999999999999999999999999999
ROWS
 N  FOB00001
 E  ROW00016                                                            12000220
 E  ROW00017                                                            12000230
 E  ROW00018                                                            12000240
 E  ROW00019                                                            12000250
 L  ROW00020                                                            12000260
 L  ROW00021                                                            12000270
 L  ROW00022                                                            12000280
 E  ROW00023                                                            12000290
 L  ROW00024                                                            12000300
 E  ROW00025                                                            12000310
 L  ROW00026                                                            12000320
 G  ROW00027                                                            12000330
 E  ROW00028                                                            12000340
 E  ROW00029                                                            12000350
 E  ROW00030                                                            12000360
 E  ROW00031                                                            12000370
 E  ROW00032                                                            12000380
 E  ROW00033                                                            12000390
 L  ROW00034                                                            12000400
 E  ROW00035                                                            12000410
 E  ROW00036                                                            12000420
 E  ROW00037                                                            12000430
 E  ROW00038                                                            12000440
 L  ROW00039                                                            12000450
 L  ROW00040                                                            12000460
 L  ROW00041                                                            12000470
 E  ROW00042                                                            12000480
 L  ROW00043                                                            12000490
 E  ROW00044                                                            12000500
 L  ROW00045                                                            12000510
 G  ROW00046                                                            12000520
 E  ROW00047                                                            12000530
 E  ROW00048                                                            12000540
 E  ROW00049                                                            12000550
 E  ROW00050                                                            12000560
 E  ROW00051                                                            12000570
 E  ROW00052                                                            12000580
 L  ROW00053                                                            12000590
COLUMNS                                                                 12001360
    COL00001  ROW00016     0.0
    COL00002  ROW00017    0.0
    COL00003  ROW00018    0.0
    COL00004  ROW00019  0.0
    COL00005  ROW00020    0.0
    COL00006  ROW00020     0.0
    COL00007  ROW00017    0.00
```

```
          COL00008  ROW00018   0.0
          COL00009  ROW00019    0.0
          COL00010  ROW00020    0.0
          COL00011  ROW00020    0.0
          COL00012  ROW00018   0.0
          COL00013  ROW00021   0.0
          COL00014  ROW00023      -.480000   ROW00031       -.490000          12001630
          COL00015  ROW00023      -.480000   ROW00030       1.000000          12001650
          COL00015  ROW00031      -.490000                                    12001660
          COL00016  ROW00023      -.480000   ROW00031       -.490000          12001680
          COL00017  ROW00016      -.500000   ROW00017       -.500000          12001700
          COL00017  ROW00023     -9.320000   ROW00030        .700000          12001710
          COL00017  ROW00031      -.560000   ROW00032      -1.000000          12001720
          COL00017  ROW00034      1.000000                                    12001730
          COL00018  ROW00020      1.000000   ROW00031       1.000000          12001750
          COL00019  ROW00021      1.000000   ROW00032       1.000000          12001770
          COL00020  ROW00022      1.000000   ROW00033       1.000000          12001790
      ENDATA
      1.000
      ROWS
       N  FOB00001
       E  ROW00016                                                            12000220
       E  ROW00017                                                            12000230
       E  ROW00018                                                            12000240
       E  ROW00019                                                            12000250
       L  ROW00020                                                            12000260
       L  ROW00021                                                            12000270
       L  ROW00022                                                            12000280
       E` ROW00023                                                            12000290
       L  ROW00024                                                            12000300
       E  ROW00025                                                            12000310
       L  ROW00026                                                            12000320
       G  ROW00027                                                            12000330
       E  ROW00028                                                            12000340
       E  ROW00029                                                            12000350
       E  ROW00030                                                            12000360
       E  ROW00031                                                            12000370
       E  ROW00032                                                            12000380
       E  ROW00033                                                            12000390
       L  ROW00034                                                            12000400
       E  ROW00035                                                            12000410
       E  ROW00036                                                            12000420
       E  ROW00037                                                            12000430
       E  ROW00038                                                            12000440
       L  ROW00039                                                            12000450
       L  ROW00040                                                            12000460
       L  ROW00041                                                            12000470
       E  ROW00042                                                            12000480
       L  ROW00043                                                            12000490
       E  ROW00044                                                            12000500
       L  ROW00045                                                            12000510
       G  ROW00046                                                            12000520
       E  ROW00047                                                            12000530
       E  ROW00048                                                            12000540
       E  ROW00049                                                            12000550
       E  ROW00050                                                            12000560
       E  ROW00051                                                            12000570
       E  ROW00052                                                            12000580
```

```
 L  ROW00053                                                         12000590
COLUMNS
    COL00021  FOB00001   -35.000000   ROW00016      1.000000         12001800
    COL00022  FOB00001    54.900000   ROW00016      1.000000         12001810
    COL00022  ROW00018     1.000000   ROW00023      -.400000         12001820
    COL00022  ROW00033     -.500000                                  12001830
    COL00023  FOB00001    54.900000   ROW00017      1.000000         12001840
    COL00023  ROW00019     1.000000   ROW00023      -.400000         12001850
    COL00023  ROW00033     -.500000                                  12001860
    COL00024  FOB00001   -35.000000   ROW00017      1.000000         12001870
    COL00025  FOB00001    23.500000   ROW00018     -1.000000         12001880
    COL00025  ROW00023    -1.700000   ROW00028      1.000000         12001890
    COL00025  ROW00031     -.245000   ROW00033     -1.400000         12001900
    COL00026  FOB00001    23.500000   ROW00019     -1.000000         12001910
    COL00026  ROW00023    -1.700000   ROW00029      1.000000         12001920
    COL00026  ROW00031     -.245000   ROW00033     -1.400000         12001930
    COL00027  FOB00001     8.720000   ROW00025      1.000000         12001940
    COL00027  ROW00031      .200000                                  12001950
    COL00028  FOB00001     9.720000   ROW00023      1.500000         12001960
    COL00028  ROW00024     1.000000   ROW00025      1.000000         12001970
    COL00029  FOB00001     6.740000   ROW00025      1.000000         12001980
    COL00029  ROW00027     1.000000   ROW00032      1.200000         12001990
    COL00030  FOB00001     6.840000   ROW00025      1.000000         12002000
    COL00030  ROW00027     1.000000   ROW00033      1.000000         12002010
    COL00031  FOB00001    15.000000   ROW00023      1.000000         12002020
    COL00032  FOB00001    22.500000   ROW00033      1.000000         12002030
    COL00033  ROW00025     1.000000   ROW00026      1.000000         12002040
    COL00034  FOB00001  -500.000000   ROW00028     -1.000000         12002050
    COL00034  ROW00042     -.480000   ROW00050      -.490000         12002060
    COL00035  FOB00001    18.700000   ROW00028     -1.000000         12002070
    COL00035  ROW00042     -.480000   ROW00049      1.000000         12002080
    COL00035  ROW00050     -.490000                                  12002090
    COL00036  FOB00001  -258.300000   ROW00029     -1.000000         12002100
    COL00036  ROW00042     -.480000   ROW00050      -.490000         12002110
    COL00037  FOB00001  -662.000000   ROW00030     -1.000000         12002120
    COL00037  ROW00034    -1.000000   ROW00035      -.500000         12002130
    COL00037  ROW00036     -.500000   ROW00042     -9.320000         12002140
    COL00037  ROW00049      .700000   ROW00050      -.560000         12002150
    COL00037  ROW00051    -1.000000   ROW00053      1.000000         12002160
    COL00038  FOB00001     3.000000   ROW00031     -1.000000         12002170
    COL00038  ROW00039     1.000000   ROW00050      1.000000         12002180
    COL00039  FOB00001      .390000   ROW00032     -1.000000         12002190
    COL00039  ROW00040     1.000000   ROW00051      1.000000         12002200
    COL00040  FOB00001      .470000   ROW00033     -1.000000         12002210
    COL00040  ROW00041     1.000000   ROW00052      1.000000         12002220
    COL00041  FOB00001   -35.000000   ROW00035      1.000000         12002230
    COL00042  FOB00001    54.900000   ROW00035      1.000000         12002240
    COL00042  ROW00037     1.000000   ROW00042      -.400000         12002250
    COL00042  ROW00052     -.500000                                  12002260
    COL00043  FOB00001    54.900000   ROW00036      1.000000         12002270
    COL00043  ROW00038     1.000000   ROW00042      -.400000         12002280
    COL00043  ROW00052     -.500000                                  12002290
    COL00044  FOB00001   -35.000000   ROW00036      1.000000         12002300
    COL00045  FOB00001    23.500000   ROW00037     -1.000000         12002310
    COL00045  ROW00042    -1.700000   ROW00047      1.000000         12002320
    COL00045  ROW00050     -.245000   ROW00052     -1.400000         12002330
    COL00046  FOB00001    23.500000   ROW00038     -1.000000         12002340
    COL00046  ROW00042    -1.700000   ROW00048      1.000000         12002350
```

```
        COL00046  ROW00050       -.245000  ROW00052      -1.400000          12002360
        COL00047  FOB00001       8.720000  ROW00044       1.000000          12002370
        COL00047  ROW00050        .200000                                   12002380
        COL00048  FOB00001       9.720000  ROW00042       1.500000          12002390
        COL00048  ROW00043       1.000000  ROW00044       1.000000          12002400
        COL00049  FOB00001       6.740000  ROW00044       1.000000          12002410
        COL00049  ROW00046       1.000000  ROW00051       1.200000          12002420
        COL00050  FOB00001       6.840000  ROW00044       1.000000          12002430
        COL00050  ROW00046       1.000000  ROW00052       1.000000          12002440
        COL00051  FOB00001      15.000000  ROW00042       1.000000          12002450
        COL00052  FOB00001      22.500000  ROW00052       1.000000          12002460
        COL00053  ROW00044       1.000000  ROW00045       1.000000          12002470
        COL00054  FOB00001    -500.000000  ROW00047      -1.000000          12002480
        COL00055  FOB00001      18.700000  ROW00047      -1.000000          12002500
        COL00056  FOB00001    -258.300000  ROW00048      -1.000000          12002530
        COL00057  FOB00001    -662.000000  ROW00049      -1.000000          12002550
        COL00057  ROW00053  -1.0
        COL00058  FOB00001  3.00
        COL00058  ROW00050  -1.00
        COL00059  FOB00001   .39
        COL00059  ROW00051  -1.00
        COL00060  FOB00001  .47
        COL00060  ROW00052  -1.00
    RHS                                                                     12004280
        RHS       ROW00020   1800.00
        RHS       ROW00021   2400.000000   ROW00022   1200.000000           12004350
        RHS       ROW00024   2566.670000   ROW00025   6900.000000           12004360
        RHS       ROW00026   1600.000000   ROW00027    800.000000           12004370
        RHS       ROW00039   1800.000000   ROW00040   2400.000000           12004380
        RHS       ROW00041   1200.000000   ROW00043   2566.670000           12004390
        RHS       ROW00044   6900.000000   ROW00045   1600.000000           12004400
        RHS       ROW00046    800.0
    ENDATA                                                                  12004560




    999999999999999999999999999999999999999999999999999999999999999999999999999
    99999999999999999999999999999999999999999999999999999999999999999999999999999
    999999999999999999999999999999999999999999999999999999999999999999999999999
    9999999999999999999999999999999999999999999999999999999999999999999999999999999999999
    99999999999999999999999999999999999999999999999999999999999999999999999999
    STOCH
        ROW00020  VAL        1600.00       PROB        0.5
        ROW00020  VAL        1800.00       PROB        0.5
        ROW00045  VAL        1600.0      PROB           0.5
        ROW00045  VAL        1800.0      PROB           0.5
        ROW00046  VAL         600.0      PROB           0.5
        ROW00046  VAL         800.0      PROB           0.5
    ENDATA
```

A3. Listing of solution output for SCAGR7.S2

```
PROBLEM     1
DENSITY OF CONSTRAINT MATRIX IS  0.191
DENSITY OF CONSTRAINT MATRIX IS  0.092

STOCHASTIC VARIABLES

ROW           VALUE      PROB

ROW00020     1600.00  0.500000
ROW00020     1800.00  0.500000
ROW00045     1600.00  0.500000
ROW00045     1800.00  0.500000
ROW00046      600.00  0.500000
ROW00046      800.00  0.500000
CHECK THE A MATRIX OF NODE     1,    1,    1


    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00   -1.00    1.00    1.00    1.00    1.00    1.00
    1.00  -35.00    1.00   54.90    1.00    1.00   -0.40   -0.50   54.90    1.00    1.00   -0.40   -0.50  -35.00    1.00
   23.50   -1.00   -1.70    1.00   -0.24   -1.40   23.50   -1.00   -1.70    1.00   -0.24   -1.40    8.72    1.00    0.20
    9.72    1.50    1.00    1.00    6.74    1.00    1.00    1.20    6.84    1.00    1.00    1.00   15.00    1.00   22.50
    1.00    1.00    1.00-500.00   -1.00   18.70    1.00-258.30   -1.00-662.00   -1.00    3.00   -1.00    0.39   -1.00
    0.47   -1.00


IA
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    16     1     2     1     2     4     6    16     1     3     5     6    16     1     3
     1     4     6    11    14    16     1     5     6    12    14    16     1     8    14
     1     6     7     8     1     8    10    15     1     8    10    16     1     6     1
    16     8     9     1    11     1    11     1    12     1    13     1    11     1    15
     1    16


LA
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    16    17    19    24    29    31    37    43    46    50    54    58    60    62    64
    66    68    70    72    74    76


NROW=   16 NCOL=    36 NELM=    77
KINBAS
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    16     0     0     0     0     0     0     0     0     0     0     0     0     0     0
     0     0     0     0     0     0
JH
     1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
    16
X VALUES
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0
XKSI"S
    0.0  158.00  158.00    0.0     0.0 3092.962566.676900.001600.00  800.00    0.0     0.0 -375.20  -92.12-684.00
 -150.00
    0.0  158.00  158.00    0.0     0.0 3092.962566.676900.001600.00  800.00    0.0     0.0 -375.20  -92.12-684.00
 -150.00
PROB OF     1     1     1 IS1.00000
ABN MATRIX
```

```
   0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
   0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0   -0.48
  -0.49   -0.48    1.00   -0.49   -0.48   -0.49   -0.50   -0.50   -9.32    0.70   -0.56   -1.00    1.00    1.00    1.00
   1.00    1.00    1.00    1.00    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
   0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
   0.0     0.0
  IBN
    1     2     3     4     5     6     7     8     9    10    11    12    13    14    15
   16     2     3     4     5     6     6     3     4     5     6     6     4     7     9
   17     9    16    17     9    17     2     3     9    16    17    18    20     6    17
    7    18     8    19     0     0     0     0     0     0     0     0     0     0     0
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
    0     0
  LBN
    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    29    30
   32    35    37    44    46    48
  CHECK THE A MATRIX OF NODE     2,    1,    1


   1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00   -1.00    1.00    1.00
   1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00
   1.00   -1.00    1.00    1.00    1.00    1.00    1.00    1.00    1.00  -35.00    1.00   54.90    1.00    1.00   -0.40
  -0.50   54.90    1.00    1.00   -0.40   -0.50  -35.00    1.00   23.50   -1.00   -1.70    1.00   -0.24   -1.40   23.50
  -1.00   -1.70    1.00   -0.24   -1.40    8.72    1.00    0.20    9.72    1.50    1.00    1.00    6.74    1.00    1.00
   1.20    6.84    1.00    1.00    1.00   15.00    1.00   22.50    1.00    1.00    1.00-500.00   -1.00   -0.48   -0.49
  18.70   -1.00   -0.48    1.00   -0.49-258.30   -1.00   -0.48   -0.49-662.00   -1.00   -1.00   -0.50   -0.50   -9.32
   0.70   -0.56   -1.00    1.00    3.00   -1.00    1.00    1.00    0.39   -1.00    1.00    1.00    0.47   -1.00    1.00
   1.00  -35.00    1.00   54.90    1.00    1.00   -0.40   -0.50   54.90    1.00    1.00   -0.40   -0.50  -35.00    1.00
  23.50   -1.00   -1.70    1.00   -0.24   -1.40   23.50   -1.00   -1.70    1.00   -0.24   -1.40    8.72    1.00    0.20
   9.72    1.50    1.00    1.00    6.74    1.00    1.00    1.20    6.84    1.00    1.00    1.00   15.00    1.00   22.50
   1.00    1.00    1.00-500.00   -1.00   18.70   -1.00-258.30   -1.00-662.00   -1.00   -1.00    3.00   -1.00    0.39
  -1.00    0.47   -1.00


  IA
    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    29    30
   31    32    33    34    35    36    37    38    39     1     2     1     2     4     9
   19     1     3     5     9    19     1     3     1     4     9    14    17    19     1
    5     9    15    17    19     1    11    17     1     9    10    11     1    11    13
   18     1    11    13    19     1     9     1    19    11    12     1    14    28    36
    1    14    28    35    36     1    15    28    36     1    16    20    21    22    28
   35    36    37    39     1    17    25    36     1    18    26    37     1    19    27
   38     1    21     1    21    23    28    38     1    22    24    28    38     1    22
    1    23    28    33    36    38     1    24    28    34    36    38     1    30    36
    1    28    29    30     1    30    32    37     1    30    32    38     1    28     1
   38    30    31     1    33     1    33     1    34     1    35    39     1    36     1
   37     1    38


  LA
    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    29    30
   31    32    33    34    35    36    37    38    39    40    42    47    52    54    60
   66    69    73    77    81    83    85    87    91    96   100   110   114   118   122
  124   129   134   136   142   148   151   155   159   163   165   167   169   171   173
  175   178   180   182
```

```


 NROW=   39 NCOL=   79 NELM=  183
 KINBAS
    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   29   30
   31   32   33   34   35   36   37   38   39    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0
 JH
    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   29   30
   31   32   33   34   35   36   37   38   39
 X VALUES
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 XKSI"S
    0.0     0.0     0.0     0.0     0.0 1800.002400.001200.00     0.0 2566.676900.001600.00 800.00     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0 1800.002400.001200.00     0.0 2566.676900.00
 1600.00 800.00     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0 1800.002400.001200.00     0.0 2566.676900.001600.00 800.00     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0 1800.002400.001200.00     0.0 2566.676900.00
 1600.00 800.00     0.0     0.0     0.0     0.0     0.0     0.0     0.0
 PROB OF     2     1     1 IS1.00000
 ABN MATRIX
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0     0.0
    0.0     0.0     0.0
 IBN
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0
 LBN
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
    0    0    0    0    0    0    0    0    0    0    0    0    0    0    0
```

```
         0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
         0     0     0     0     0     0     0     0     0     0     0     0     0     0     0
         0     0     0     0
      ITNS  RRES       PIV        IN   OUT  OBJ         CMIN        CMAX         THET    NINF NOPT
         1  0.91E-12  0.15E+01    24    6  -0.20E+05 -0.25E+01 -0.10E+11  0.21E+04     9   14
         2  0.91E-12 -0.10E+01    22    5  -0.20E+05 -0.28E+01 -0.10E+11  0.0          8   14
         3  0.91E-12  0.10E+01    19   12  -0.20E+05 -0.45E+01 -0.10E+11  0.0          8   13
         4  0.91E-12 -0.19E+01    32   16  -0.69E+04 -0.45E+01 -0.10E+11  0.79E+02     8   13
         5  0.18E-11  0.53E+00    26    3   0.52E+04 -0.34E+01 -0.10E+11  0.15E+03     7   11
         6  0.18E-11 -0.10E+01    21    4   0.52E+04 -0.42E+01 -0.10E+11  0.0          6   10
         7  0.18E-11  0.10E+01    18   11   0.52E+04 -0.64E+01 -0.10E+11  0.0          6    9
         8  0.27E-11  0.10E+01    31    2  -0.14E+05 -0.64E+01 -0.10E+11  0.16E+03     6   10
         9  0.24E-11  0.10E+01    36    8  -0.43E+05 -0.20E+01 -0.10E+11  0.39E+04     5    7
        10  0.56E-10 -0.10E+01    33   13   0.21E+06 -0.10E+01 -0.10E+11  0.38E+03     3    3
        11  0.57E-10 -0.10E+01    34   14   0.21E+06 -0.10E+01 -0.10E+11  0.15E+02     2    2
        12  0.57E-10 -0.10E+01    35   15   0.20E+06 -0.10E+01 -0.10E+11  0.68E+03     1    1
        12 ITERATIONS SO FAR


      WE HAVE MADE     1 PASSES
      WE ARE AT PROBLEM     1,     1,     1
      X VALUES
      204891.07     158.00    4395.63     158.00     158.00    2504.37      62.30    3945.23
         1600.00    3595.63     158.00     158.00     375.20      14.70     684.00     158.00
      JH
         1    31    26    21    22    24     7    36     9    10    18    19    33    34    35
        32
      KINBAS
         1     0     0    -1    -1     0     7     0     9    10     0     0    -1    -1    -1
        -1     0    11    12     0     4     5     0     6     0     3     0     0     0     0
         2    16    13    14    15     8
      BS CURRENTLY
             0.0     158.00     158.00        0.0        0.0    3092.96    2566.67    6900.00
         1600.00     800.00        0.0        0.0    -375.20     -92.12    -684.00    -150.00
      CURRENT Y
            1.00        0.0        0.0        0.0        0.0        0.0        0.0        0.0
             0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
      CURRENT PI
             0.0        0.0        0.0        0.0        0.0        0.0        0.0        0.0
             0.0        0.0        0.0        0.0        0.0        0.0       1.00        0.0
      STATUS AT NDCOM=F
      AT BEG OF NDCOM, JCUR=     1
      AT END OF ND, JCUR=     2
      ITNS  RRES       PIV        IN   OUT  OBJ         CMIN        CMAX         THET    NINF NOPT
        13  0.91E-12  0.15E+01    47    9  -0.24E+05 -0.25E+01 -0.10E+11  0.24E+04    13   18
        14  0.91E-12 -0.10E+01    45    5  -0.24E+05 -0.23E+01 -0.10E+11  0.0         12   18
        15  0.91E-12  0.10E+01    42   15  -0.24E+05 -0.41E+01 -0.10E+11  0.0         12   17
        16  0.91E-12 -0.48E+00    55   28  -0.24E+05 -0.41E+01 -0.10E+11  0.0         12   17
        17  0.91E-12 -0.15E+01    67   36  -0.24E+05 -0.14E+02 -0.10E+11  0.0         12   16
        18  0.36E-11  0.29E+01    57   10  -0.81E+04 -0.79E+01 -0.10E+11  0.47E+02    12   18
        19  0.36E-11 -0.50E+00    56   21  -0.81E+04 -0.77E+01 -0.10E+11  0.0         12   19
        20  0.36E-11 -0.10E+01    61   22  -0.81E+04 -0.16E+02 -0.10E+11  0.0         12   19
        21  0.36E-11  0.10E+01    62   23  -0.81E+04 -0.16E+02 -0.10E+11  0.0         12   18
        22  0.36E-11  0.10E+01    64   24  -0.81E+04 -0.17E+02 -0.10E+11  0.0         12   17
        23  0.36E-11  0.10E+01    65   33  -0.81E+04 -0.18E+02 -0.10E+11  0.0         12   16
        24  0.36E-11  0.10E+01    74   34  -0.81E+04 -0.18E+02 -0.10E+11  0.0         12   17
        25  0.36E-11  0.14E+01    75   35  -0.81E+04 -0.18E+02 -0.10E+11  0.0         12   16
        26  0.36E-11 -0.14E+01    76   37  -0.81E+04 -0.13E+02 -0.10E+11  0.0         12   15
```

```
 27  0.36E-11 -0.23E+01   68   38  -0.81E+04 -0.13E+02 -0.10E+11  0.0          12   14
 28  0.36E-11  0.50E+00   63   23  -0.81E+04 -0.93E+01 -0.10E+11  0.0          12   16
 29  0.36E-11  0.60E+00   60   39  -0.81E+04 -0.19E+02 -0.10E+11  0.0          12   15
 30  0.36E-11 -0.10E+01   54   14  -0.81E+04 -0.31E+02 -0.10E+11  0.0          12   15
 31  0.36E-11 -0.10E+01   44    4  -0.81E+04 -0.30E+02 -0.10E+11  0.0          12   15
 32  0.13E-09  0.10E+01   41    5   0.38E+06 -0.31E+02 -0.10E+11  0.96E+02     12   14
 33  0.15E-09  0.10E+02   50   29   0.49E+06 -0.16E+02 -0.10E+11  0.55E+02     12   13
 34  0.16E-09  0.98E+00   53    2   0.51E+06 -0.20E+01 -0.10E+11  0.67E+02     11   16
 35  0.25E-09  0.10E+01   45    3   0.52E+06 -0.21E+01 -0.10E+11  0.19E+03     10   13
 36  0.10E-09  0.83E+00   78   30   0.50E+06 -0.17E+01 -0.10E+11  0.48E+04      9   11
 37  0.89E-10  0.20E+00   46   17   0.46E+06 -0.12E+01 -0.10E+11  0.42E+04      7    9
 38  0.64E-10  0.50E+01   77   11   0.46E+06 -0.50E+01 -0.10E+11  0.20E+02      6    9
 39  0.52E-10 -0.10E+01   58   18   0.46E+06 -0.10E+01 -0.10E+11  0.31E+03      5    4
 40  0.51E-10  0.20E+00   48   11   0.46E+06 -0.10E+01 -0.10E+11  0.10E+03      4    4
 41  0.17E-09  0.10E+01   43    3   0.45E+06 -0.17E+01 -0.10E+11  0.19E+03      4    7
 42  0.27E-09  0.10E+01   40    2   0.43E+06 -0.17E+01 -0.10E+11  0.66E+02      4    6
 43  0.19E-09  0.35E+01   53   13   0.45E+06 -0.18E+01 -0.10E+11  0.56E+02      3    6
 44  0.13E-09  0.11E+00   59   20   0.39E+06 -0.72E+00 -0.10E+11  0.25E+03      3    5
 45  0.12E-09  0.10E+01   79   27   0.38E+06 -0.10E+01 -0.10E+11  0.95E+03      3    4
 46  0.24E-09  0.46E+01   20   22   0.44E+06 -0.25E+01 -0.10E+11  0.29E+02      3    4
 47  0.22E-09  0.28E+00   66    2   0.45E+06 -0.52E+00 -0.10E+11  0.37E+02      3    3
 48  0.17E-09  0.10E+01   45    3   0.46E+06 -0.18E+01 -0.10E+11  0.19E+03      3    2
 49  0.24E-09 -0.11E+00   70   16   0.49E+06 -0.11E+00 -0.10E+11  0.25E+03      3    1
INFEAS AT PROB      1      1      1
MAKING A FEASIBILITY CUT
ITNS  RRES       PIV      IN   OUT  OBJ        CMIN       CMAX        THET    NINF NOPT
 50  0.95E-11  0.10E+01   24   10   0.20E+06 -0.20E+01 -0.10E+11  0.36E+04      1    3
 51  0.95E-11  0.10E+01   31   11   0.28E+06 -0.52E+03 -0.10E+11  0.16E+03      0    5
 52  0.95E-11  0.10E+01   30    9   0.29E+06 -0.93E+01 -0.10E+11  0.16E+04      0    3
 53  0.10E-10  0.20E+01   10   17   0.30E+06 -0.20E+01 -0.10E+11  0.12E+04      0    2
 54  0.16E-10  0.10E+01   26   10   0.30E+06 -0.21E+01 -0.10E+11  0.79E+03      0    1
 55  0.28E-10  0.50E+00   17   16   0.30E+06 -0.51E-01 -0.10E+11  0.31E+04      0    1
 55 ITERATIONS SO FAR


WE HAVE MADE     3 PASSES
WE ARE AT PROBLEM      1,      1,      1
X VALUES
298780.88     158.00     158.00     158.00     158.00    2504.37      62.30     450.40
   1600.00    2345.23     158.00     158.00     375.20      14.70    3498.27    3112.88
   1995.63
JH
    1    19    20    22    23    25     7    27    30    26    31    33    34    35    36
   17    10
KINBAS
    1     0     0    -1    -1     0     7     0     0    17     0     0    -1    -1    -1
   -1    16     0     2     3     0     4     5     0     6    10     8     0     0     9
   11     0    12    13    14    15     0
BS CURRENTLY
       0.0     158.00     158.00        0.0        0.0    3092.96    2566.67    6900.00
   1600.00     800.00        0.0        0.0    -375.20     -92.12    -684.00    -150.00
   2400.00
CURRENT Y
      1.00     419.52     177.82    -474.90    -233.20      -1.67        0.0       7.21
      7.21        0.0    -500.00    -258.30    -662.00       3.00       0.39       0.37
       0.0
CURRENT PI
      1.00     419.52     177.82    -474.90    -233.20      -1.67        0.0      -7.21
```

```
          7.21        0.0      -500.00     -258.30    -662.00       3.00        0.39        0.37
          0.0


 LOOKING FOR CUTS TO DELETE
 ITNS  RRES        PIV         IN    OUT  OBJ         CMIN         CMAX         THET    NINF NOPT
  56  0.25E-09  0.10E+01     49    13   0.32E+06 -0.22E+01 -0.10E+11  0.80E+03     5     8
  57  0.30E-09  0.10E+01     62    22   0.33E+06 -0.21E+01 -0.10E+11  0.13E+03     5     8
  58  0.34E-09  0.10E+01     61    21   0.31E+06 -0.21E+01 -0.10E+11  0.13E+03     5     7
  59  0.34E-09  0.10E+01     29    29   0.26E+06 -0.15E+01 -0.10E+11  0.26E+04     5     6
  60  0.35E-09  0.10E+01     40    14   0.22E+06 -0.19E+01 -0.10E+11  0.11E+03     4     5
  61  0.35E-09  0.37E+01     43    36   0.21E+06 -0.19E+01 -0.10E+11  0.69E+02     4     4
  62  0.36E-09  0.29E+01     77     9   0.20E+06 -0.26E+01 -0.10E+11  0.74E+02     4     5
  63  0.35E-09  0.71E+00     10     3   0.20E+06 -0.14E+01 -0.10E+11  0.24E+02     4     4
  64  0.38E-09  0.20E+00     13     9   0.20E+06 -0.10E+01 -0.10E+11  0.46E+03     4     3
  65  0.37E-09  0.12E+01     63    19   0.19E+06 -0.12E+01 -0.10E+11  0.74E+02     4     6
 INFEAS AT PROB      1      1      1
 MAKING A FEASIBILITY CUT
 ITNS  RRES        PIV         IN    OUT  OBJ         CMIN         CMAX         THET    NINF NOPT
  66  0.35E-10  0.10E+01     25    10   0.29E+06 -0.24E+01 -0.10E+11  0.20E+04     1     3
  67  0.43E-10  0.24E+01     10    18   0.30E+06 -0.21E+01 -0.10E+11  0.12E+04     0     1
  68  0.49E-10  0.10E+01     38    10   0.30E+06 -0.20E+01 -0.10E+11  0.81E+03     0     1
  68 ITERATIONS SO FAR


 WE HAVE MADE     5 PASSES
 WE ARE AT PROBLEM     1,     1,     1
 X VALUES
 298698.68      158.00      158.00      158.00      158.00    2504.37       62.30     1539.43
   1600.00      805.79      158.00      158.00      375.20      14.70     2531.32     1256.19
   1501.29     1995.63
 JH
     1    20    21    23    24    26     7    27    31    38    32    34    35    36    37
    28    17    10
 KINBAS
     1     0     0    -1    -1     0     7     0     0    18     0     0    -1    -1    -1
    -1    17     0     0     2     3     0     4     5     0     6     8    16     0     0
     9    11     0    12    13    14    15    10
 BS CURRENTLY
       0.0      158.00      158.00        0.0         0.0     3092.96     2566.67     6900.00
   1600.00      800.00        0.0         0.0      -375.20     -92.12     -684.00     -150.00
   2400.00     4800.00
 CURRENT Y
       1.00      419.85      178.15     -475.16     -233.46      -1.61        0.0        -7.31
       7.31        0.0      -500.00     -258.30     -662.03       3.00        0.47        0.47
       0.0         0.04
 CURRENT PI
       1.00      419.85      178.15     -475.16     -233.46      -1.61        0.0        -7.31
       7.31        0.0      -500.00     -258.30     -662.03       3.00        0.47        0.47
       0.0         0.04


 LOOKING FOR CUTS TO DELETE


 DELETING ROW     17 VALUE =     0.150E+04
 ITNS  RRES        PIV         IN    OUT  OBJ         CMIN         CMAX         THET    NINF NOPT
  69  0.26E-09  0.14E+01     53    10   0.26E+06 -0.18E+02 -0.10E+11  0.17E+02     5     8
```

```
 70  0.23E-09  0.41E+01   77   24   0.19E+06 -0.12E+02 -0.10E+11  0.15E+03    5    6
 71  0.26E-09  0.12E+01   48   26   0.19E+06 -0.10E+01 -0.10E+11  0.20E+03    3    2
 72  0.30E-09  0.48E+00   50    2   0.22E+06 -0.90E+00 -0.10E+11  0.19E+03    3    1
INFEAS AT PROB     1      1      1
MAKING A FEASIBILITY CUT
ITNS  RRES       PIV       IN   OUT  OBJ        CMIN       CMAX       THET   NINF NOPT
 73  0.56E-10  0.42E+00   17    8   0.30E+06 -0.50E+00 -0.10E+11  0.37E+04    1    2
 74  0.68E-10  0.12E+01   27   18   0.30E+06 -0.10E+00 -0.10E+11  0.13E+04    0    1
 74 ITERATIONS SO FAR


WE HAVE MADE    7 PASSES
WE ARE AT PROBLEM     1,     1,     1
X VALUES
298677.96    158.00    158.00    158.00    158.00   2504.37     62.30    487.76
  1600.00   1995.63    158.00    158.00    375.20     14.70   2287.44   1009.03
  1459.43   1336.20
JH
   1   20   21   23   24   26    7   17   31   10   32   34   35   36   37
  38   28   27
KINBAS
   1    0    0   -1   -1    0    7    0    0   10    0    0   -1   -1   -1
  -1    8    0    0    2    3    0    4    5    0    6   18   17    0    0
   9   11    0   12   13   14   15   16
BS CURRENTLY
      0.0    158.00    158.00       0.0       0.0   3092.96   2566.67   6900.00
  1600.00    800.00       0.0       0.0   -375.20    -92.12   -684.00   -150.00
  4800.00   2400.00
CURRENT Y
     1.00    419.85    178.15   -475.16   -233.46     -1.61       0.0     -7.31
     7.31       0.0   -500.00   -258.30   -661.97      3.00      0.47      0.47
     0.00      0.08
CURRENT PI
     1.00    419.85    178.15   -475.16   -233.46     -1.61       0.0     -7.31
     7.31       0.0   -500.00   -258.30   -661.97      3.00      0.47      0.47
     0.00      0.08


LOOKING FOR CUTS TO DELETE
ITNS  RRES       PIV       IN   OUT  OBJ        CMIN       CMAX       THET   NINF NOPT
INFEAS AT PROB     1      1      1
MAKING A FEASIBILITY CUT
ITNS  RRES       PIV       IN   OUT  OBJ        CMIN       CMAX       THET   NINF NOPT
 75  0.20E-10  0.10E+01   34   11   0.22E+06 -0.10E+01 -0.10E+11  0.16E+03    1    1
 76  0.70E-10  0.10E+01   33   19   0.24E+06 -0.52E+03 -0.10E+11  0.45E+02    0    2
 76 ITERATIONS SO FAR


WE HAVE MADE    9 PASSES
WE ARE AT PROBLEM     1,     1,     1
X VALUES
240302.65    158.00    158.00    158.00    158.00   2504.37     62.30   1365.63
  1600.00   1995.63    112.56    158.00    375.20     14.70   2400.00    915.23
   375.20   1430.00     45.44
JH
   1   21   22   24   25   27    7   29   32   10   34   35   36   37   38
  39   17   28   33
KINBAS
```

```
  1    0    0   -1   -1    0    7    0    0   10    0    0   -1   -1   -1
 -1   17    0    0    0    2    3    0    4    5    0    6   18    8    0
  0    9   19   11   12   13   14   15   16
BS CURRENTLY
     0.0      158.00     158.00      0.0        0.0     3092.96    2566.67    6900.00
  1600.00     800.00       0.0       0.0     -375.20     -92.12    -684.00    -150.00
  4800.00    2400.00      -0.0
CURRENT Y
     1.00     419.85     178.15   -475.16    -233.46      -1.61       0.0        7.31
     7.31       0.0     -500.00   -258.30    -506.39       3.00       0.47       0.47
     0.0        0.08     518.61
CURRENT PI
     1.00     419.85     178.15   -475.16    -233.46      -1.61       0.0       -7.31
     7.31       0.0     -500.00   -258.30    -506.39       3.00       0.47       0.47
     0.0        0.08     518.61


LOOKING FOR CUTS TO DELETE


DELETING ROW     17 VALUE =    0.375E+03
ITNS  RRES        PIV         IN   OUT  OBJ        CMIN        CMAX         THET     NINF NOPT
  77  0.24E-09  0.10E+01    40   14   0.31E+06 -0.11E+01 -0.10E+11  0.75E+02    1    7
  78  0.23E-09  0.10E+01    73   33   0.41E+06 -0.52E+03 -0.10E+11  0.19E+03    0    9
  79  0.23E-09  0.11E+01    39   36   0.41E+06 -0.50E+03 -0.10E+11  0.14E+02    0    7
  80  0.26E-09  0.18E+00    66   14   0.44E+06 -0.88E+02 -0.10E+11  0.34E+03    0    8
  81  0.27E-09  0.10E+01    62   22   0.46E+06 -0.11E+03 -0.10E+11  0.19E+03    0    5
  82  0.32E-09  0.10E+01    42    3   0.48E+06 -0.10E+03 -0.10E+11  0.19E+03    0    4
  83  0.34E-09  0.10E+01    67   29   0.53E+06 -0.20E+02 -0.10E+11  0.26E+04    0    3
  84  0.34E-09  0.20E+00    52   17   0.54E+06 -0.78E+01 -0.10E+11  0.81E+03    0    2
  85  0.36E-09  0.10E+01    27   38   0.55E+06 -0.78E+01 -0.10E+11  0.49E+03    0    3
  86  0.37E-09  0.10E+01    69   13   0.55E+06 -0.77E+01 -0.10E+11  0.23E+02    0    3
  87  0.37E-09  0.10E+01    72   32   0.55E+06 -0.72E+01 -0.10E+11  0.13E+04    0    1
VAR1   VAR2   VAR3   PROB      OBJ         KSIPI       OBJTO1       CURR BAS
    1      1     1 0.125000   0.555E+06   0.551E+05   0.0             1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    1      1     2 0.125000   0.553E+06   0.537E+05   0.693E+05       1-
CHECKING FEAS WITH DCHUZR
OK - FEAS
    1      2     1 0.125000   0.555E+06   0.551E+05   0.138E+06       1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    1      2     2 0.125000   0.553E+06   0.537E+05   0.208E+06       1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      1     1 0.125000   0.555E+06   0.551E+05   0.277E+06       1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      1     2 0.125000   0.553E+06   0.537E+05   0.346E+06       1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      2     1 0.125000   0.555E+06   0.551E+05   0.415E+06       1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      2     2 0.125000   0.553E+06   0.537E+05   0.485E+06       1
Q(X) = -0.554E+06
THETA= -0.915E+03
```

```
MAKING A LOOKAHEAD CUT
ITNS  RRES        PIV        IN   OUT  OBJ          CMIN        CMAX          THET    NINF NOPT
  88  0.73E-10  0.10E+01    18    11   0.83E+06 -0.70E+03 -0.10E+11  0.45E+02     0    2
  89  0.58E-10  0.10E+01    26    16   0.83E+06 -0.75E+01 -0.10E+11  0.88E+03     0    1
  90  0.58E-10  0.83E+00    17    10   0.84E+06 -0.63E+01 -0.10E+11  0.13E+04     0    1
  90 ITERATIONS SO FAR


WE HAVE MADE   11 PASSES
WE ARE AT PROBLEM    1,    1,    1
X VALUES
841177.50     158.00     158.00     158.00     158.00   2504.37     62.30    349.60
  1600.00    1341.92      45.44     158.00     375.20    413.83   1103.52   1995.63
   158.00     450.40  628565.98
JH
    1    21    22    24    25    27     7    28    32    17    18    35    36    37    38
   26    34    29    40
KINBAS
    1     0     0    -1    -1     0     7     0     0     0     0     0    -1    -1    -1
   -1    10    11     0     0     2     3     0     4     5    16     6     8    18     0
    0     9     0    17    12    13    14    15     0    19
BS CURRENTLY
      0.0     158.00     158.00        0.0        0.0   3092.96   2566.67   6900.00
  1600.00     800.00        0.0        0.0    -375.20    -92.12   -684.00   -150.00
  2400.00       -0.0   54420.09
CURRENT Y
     1.00    1074.94     131.10   -1131.62    -187.78     -6.08       0.0     -0.60
     0.60      -7.55   -1173.58    -229.74   -1656.40    -40.60      1.17      1.31
    -0.00      -0.00       1.00
CURRENT PI
     1.00    1074.94     131.10   -1131.62    -187.78     -6.08       0.0     -0.60
     0.60      -7.55   -1173.58    -229.74   -1656.40    -40.60      1.17      1.31
    -0.00      -0.00       1.00


LOOKING FOR CUTS TO DELETE
ITNS  RRES        PIV        IN   OUT  OBJ          CMIN        CMAX          THET    NINF NOPT
  91  0.27E-09  0.50E+01    57    31   0.63E+06 -0.50E+01 -0.10E+11  0.96E+02     4   10
  92  0.27E-09  0.50E+01    77    11   0.60E+06 -0.50E+01 -0.10E+11  0.57E+03     4   10
  93  0.26E-09  0.10E+01    40    39   0.57E+06 -0.38E+01 -0.10E+11  0.61E+02     3    9
  94  0.25E-09  0.10E+01    43     3   0.55E+06 -0.38E+01 -0.10E+11  0.19E+03     3    8
  95  0.25E-09  0.10E+01    13    38   0.55E+06 -0.30E+01 -0.10E+11  0.16E+04     3    7
  96  0.27E-09  0.10E+01    39    39   0.58E+06 -0.50E+03 -0.10E+11  0.61E+02     0    6
  97  0.31E-09  0.10E+01    42     3   0.61E+06 -0.11E+03 -0.10E+11  0.19E+03     0    4
  98  0.30E-09  0.20E+00    52    11   0.61E+06 -0.99E+01 -0.10E+11  0.73E+03     0    3
  99  0.31E-09  0.10E+01    69    32   0.62E+06 -0.77E+01 -0.10E+11  0.60E+03     0    3
 100  0.32E-09  0.83E+00    26    12   0.62E+06 -0.64E+01 -0.10E+11  0.32E+03     0    2
VAR1    VAR2    VAR3   PROB      OBJ         KSIPI       OBJTO1      CURR BAS
    1       1      1 0.125000   0.619E+06   0.252E+05   0.0              1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      1      1      2
CHECKING FEAS WITH DCHUZR
OK - FEAS
    1       2      1 0.125000   0.621E+06   0.266E+05   0.774E+05        1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      1      2      2
```

```
CHECKING FEAS WITH DCHUZR
OK - FEAS
   2      1      1 0.125000   0.619E+06   0.252E+05   0.155E+06      1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO     2
CHECKING FEAS WITH DCHUZR
OK - FEAS
   2      2      1 0.125000   0.621E+06   0.266E+05   0.232E+06      1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO     2     2     2
RETURNING TO SOLVE NEXT BATCH
ITERATING IN DNORML
ITNS  RRES       PIV        IN   OUT  OBJ       CMIN       MXINF      THET    NINF NOPT
101  0.33E-09 -0.10E+01   68   19   0.62E+06  0.0       0.78E+00  0.78E+00    1    0
VAR1   VAR2   VAR3   PROB     OBJ          KSIPI        OBJTOT      CURR BAS
   1      1      2 0.125000   0.619E+06   0.252E+05   0.310E+06      2
CHECKING FEAS WITH DCHUZR
OK - FEAS
   1      2      2 0.125000   0.621E+06   0.266E+05   0.388E+06      2
CHECKING FEAS WITH DCHUZR
OK - FEAS
   2      1      2 0.125000   0.619E+06   0.252E+05   0.465E+06      2
CHECKING FEAS WITH DCHUZR
OK - FEAS
   2      2      2 0.125000   0.621E+06   0.266E+05   0.543E+06      2
Q(X) = -0.620E+06
THETA= -0.629E+06
LOWER BOUND= -0.841E+06

UPPER BOUND= -0.833E+06
MAKING A LOOKAHEAD CUT
ITNS  RRES       PIV        IN   OUT  OBJ       CMIN       CMAX       THET    NINF NOPT
102  0.73E-10  0.12E+01    10   17   0.83E+06 -0.77E+01 -0.10E+11  0.11E+04    1    3
103  0.87E-10  0.64E+01    17   20   0.83E+06 -0.63E+01 -0.10E+11  0.31E+02    0    1
103 ITERATIONS SO FAR


WE HAVE MADE   13 PASSES
WE ARE AT PROBLEM     1,    1,    1
X VALUES
832935.41     158.00     158.00     158.00     158.00   2504.37     62.30     903.38
   1600.00    1441.85     158.00     158.00     375.20    195.38   2414.21     450.40
   1092.25      45.44 618017.08      31.23
JH
    1   22   23   25   26   28    7   27   33   29   35   36   37   38   39
   30   10   18   41   17
KINBAS
    1    0    0   -1   -1    0    7    0    0   17    0    0   -1   -1   -1
   -1   20   18    0    0    0    2    3    0    4    5    8    6   10   16
    0    0    9    0   11   12   13   14   15    0   19
BS CURRENTLY
       0.0     158.00     158.00        0.0        0.0   3092.96   2566.67   6900.00
   1600.00     800.00        0.0        0.0    -375.20    -92.12   -684.00   -150.00
   2400.00       -0.0   54420.09   25898.61
CURRENT Y
      1.00    1141.21     169.38   -1195.88    -224.05     -1.05       0.0      -8.14
```

```
     8.14       0.00   -1220.05    -248.22   -1723.39      -2.88       1.17       1.30
    -0.00       0.0        0.02       0.98
CURRENT PI
     1.00    1141.21     169.38   -1195.88    -224.05      -1.05       0.0       -8.14
     8.14       0.00   -1220.05    -248.22   -1723.39      -2.88       1.17       1.30
    -0.00       0.0        0.02       0.98


LOOKING FOR CUTS TO DELETE


DELETING ROW     18 VALUE =    0.454E+02
ITNS  RRES       PIV        IN   OUT  OBJ         CMIN        CMAX         THET    NINF NOPT
104  0.26E-09  0.15E+01    60   26   0.60E+06 -0.19E+01 -0.10E+11  0.55E+02    2    5
105  0.27E-09  0.41E+01    77   21   0.54E+06 -0.78E+01 -0.10E+11  0.38E+02    2    4
INFEAS AT PROB      1      1      1
MAKING A FEASIBILITY CUT
ITNS  RRES       PIV        IN   OUT  OBJ         CMIN        CMAX         THET    NINF NOPT
106  0.58E-10  0.10E+01    40   19   0.83E+06 -0.12E+01 -0.10E+11  0.14E+04    1    2
107  0.62E-04  0.12E+01    29   20   0.83E+06 -0.20E-02 -0.10E+11  0.14E+04    0    1
ITNS  RRES       PIV        IN   OUT  OBJ         CMIN        CMAX         THET    NINF NOPT
107 ITERATIONS SO FAR


WE HAVE MADE    15 PASSES
WE ARE AT PROBLEM     1,     1,     1
X VALUES
832935.32     158.00     158.00     158.00     158.00    2504.37      62.30     903.38
  1600.00    1092.25     158.00     158.00     375.20     195.38    2400.01      11.85
    45.43  618018.26    1430.01     462.25
JH
    1    22    23    25    26    28     7    27    33    10    35    36    37    38    39
   40    17    41    29    30
KINBAS
    1     0     0    -1    -1     0     7     0     0    10     0     0    -1    -1    -1
   -1    17     0     0     0     0     2     3     0     4     5     8     6    19    20
    0     0     9     0    11    12    13    14    15    16    18
BS CURRENTLY
      0.0     158.00     158.00       0.0        0.0     3092.96    2566.67    6900.00
  1600.00     800.00       0.0        0.0     -375.20     -92.12    -684.00    -150.00
  2400.00   54420.09   25898.61    2400.00
CURRENT Y
     1.00    1141.24     169.40   -1195.90    -224.07      -1.05       0.0       -8.15
     8.15       0.0    -1220.06    -248.22   -1723.41      -2.87       1.17       1.31
     0.0        0.02       0.98       0.00
CURRENT PI
     1.00    1141.24     169.40   -1195.90    -224.07      -1.05       0.0       -8.15
     8.15       0.0    -1220.06    -248.22   -1723.41      -2.87       1.17       1.31
     0.0        0.02       0.98       0.00


LOOKING FOR CUTS TO DELETE


DELETING ROW     17 VALUE =    0.454E+02
ITNS  RRES       PIV        IN   OUT  OBJ         CMIN        CMAX         THET    NINF NOPT
INFEAS AT PROB      1      1      1
MAKING A FEASIBILITY CUT
```

```
ITNS  RRES       PIV        IN   OUT  OBJ        CMIN       CMAX        THET    NINF NOPT
108  0.40E-01  0.83E+00    19   18   0.83E+06 -0.10E+01 -0.10E+11  0.17E+04    1    2
ITNS  RRES       PIV        IN   OUT  OBJ        CMIN       CMAX        THET    NINF NOPT
109  0.73E-10  0.12E+01    29   19   0.83E+06 -0.20E-02 -0.10E+11  0.14E+04    0    1
109 ITERATIONS SO FAR


WE HAVE MADE   17 PASSES
WE ARE AT PROBLEM    1,     1,     1
X VALUES
832935.39    158.00    158.00    158.00    158.00   2504.37     62.30    462.25
  1600.00   1092.25    158.00    158.00    375.20    195.38   2400.00     11.85
618018.26    903.38   1430.00      0.00
JH
   1   22   23   25   26   28    7   30   33   10   35   36   37   38   39
  40   41   27   29   20
KINBAS
   1    0    0   -1   -1    0    7    0    0   10    0    0   -1   -1   -1
  -1    0    0    0   20    0    2    3    0    4    5   18    6   19    8
   0    0    9    0   11   12   13   14   15   16   17
BS CURRENTLY
      0.0    158.00    158.00      0.0       0.0    3092.96   2566.67   6900.00
  1600.00    800.00      0.0       0.0    -375.20    -92.12   -684.00   -150.00
 54420.09  25898.61   2400.00   2400.00
CURRENT Y
     1.00   1141.23    169.40  -1195.90   -224.07     -1.05      0.0      -8.15
     8.15      0.0   -1220.06   -248.22  -1723.41     -2.87      1.17      1.31
     0.02      0.98      0.00      0.0
CURRENT PI
     1.00   1141.23    169.40  -1195.90   -224.07     -1.05      0.0      -8.15
     8.15      0.0   -1220.06   -248.22  -1723.41     -2.87      1.17      1.31
     0.02      0.98      0.00      0.0


LOOKING FOR CUTS TO DELETE
ITNS  RRES       PIV        IN   OUT  OBJ        CMIN       CMAX        THET    NINF NOPT
110  0.29E-09  0.19E+01    61   32   0.58E+06 -0.36E+03 -0.10E+11  0.11E+03    0    3
111  0.29E-09  0.10E+01    63   21   0.61E+06 -0.24E+03 -0.10E+11  0.10E+03    0    4
112  0.29E-09  0.33E+00    59   36   0.61E+06 -0.63E+02 -0.10E+11  0.37E+02    0    3
113  0.34E-09  0.18E+01    43   21   0.61E+06 -0.97E+02 -0.10E+11  0.48E+02    0    8
114  0.35E-09  0.15E+00    26   21   0.62E+06 -0.15E+02 -0.10E+11  0.32E+03    0    3
VAR1   VAR2   VAR3   PROB     OBJ         KSIPI      OBJTOT      CURR BAS
    1      1      1 0.125000   0.617E+06   0.252E+05   0.0            1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      1      1      2
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      1      2      1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      1      2      2
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      1      1 0.125000   0.617E+06   0.252E+05   0.772E+05      1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      2      1      2
```

```
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      2      2      1
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      2      2      2
RETURNING TO SOLVE NEXT BATCH
ITERATING IN DNORML
ITNS  RRES       PIV        IN   OUT  OBJ        CMIN        MXINF       THET    NINF NOPT
115  0.35E-09 -0.10E+01    31    13   0.62E+06  0.0        0.53E+02  0.53E+02    2    0
116  0.35E-09 -0.10E+01    68    36   0.62E+06  0.0        0.78E+00  0.78E+00    1    0
VAR1   VAR2   VAR3   PROB     OBJ          KSIPI       OBJTOT      CURR BAS
    1      1      2 0.125000   0.617E+06   0.252E+05   0.154E+06       2
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      1      2      1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    1      2      2 0.125000   0.617E+06   0.266E+05   0.231E+06       2
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      1      2 0.125000   0.617E+06   0.252E+05   0.309E+06       2
CHECKING FEAS WITH DCHUZR
TRYING TO FIND ENTERING IN DCHUZC
ENT VAR - WILL RETURN TO      2      2      1
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      2      2 0.125000   0.617E+06   0.266E+05   0.386E+06       2
RETURNING TO SOLVE NEXT BATCH
ITERATING IN DNORML
ITNS  RRES       PIV        IN   OUT  OBJ        CMIN        MXINF       THET    NINF NOPT
117  0.36E-09 -0.10E+01    59    36   0.62E+06  0.0        0.20E+03  0.20E+03    1    0
118  0.35E-09 -0.10E+01    13    18   0.62E+06  0.0        0.10E+03  0.10E+03    1    0
VAR1   VAR2   VAR3   PROB     OBJ          KSIPI       OBJTOT      CURR BAS
    1      2      1 0.125000   0.618E+06   0.266E+05   0.463E+06       3
CHECKING FEAS WITH DCHUZR
OK - FEAS
    2      2      1 0.125000   0.618E+06   0.266E+05   0.540E+06       3
Q(X) = -0.617E+06
THETA= -0.618E+06
LOWER BOUND= -0.833E+06

UPPER BOUND= -0.832E+06

TOTAL SOLUTION TIME =   1658 MILLISECONDS
SOLUTION TIME WITHOUT INPUT =    1374 MILLISECONDS


LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO     1     1     1

ROW        VALUE
FOBOOOO1    617323.7260
ROWOOO16         187.6000
ROWOOO17         187.6000
ROWOOO18         187.6000
ROWOOO19         187.6000
ROWOOO20        1404.6237
ROWOOO21           0.0000
```

```
ROW00022      1188.1548
ROW00023       586.4588
ROW00024      2566.6700
ROW00025      1786.4842
ROW00026      1600.0000
ROW00027       146.8458
ROW00028       187.6000
ROW00029       187.6000
ROW00030       420.6400
ROW00031        95.7972
ROW00032        46.5954
ROW00033       900.2505
ROW00034        45.4400
ROW00035       319.2856
ROW00036       210.3200
ROW00037       210.3200
ROW00038       210.3200
ROW00039      1704.2028
ROW00040      2080.7144
ROW00041      1000.7842
ROW00042      1133.7995
ROW00043      2566.6700
ROW00044      2133.3300
ROW00045      1600.0000
ROW00046       210.3200
ROW00047       210.3200
ROW00048       210.3200
ROW00049       482.0480
ROW00050       199.2158
ROW00051      1660.0744
ROW00052       600.0000
ROW00053        61.4080


 LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO     1      1      2

 ROW          VALUE
FOB00001    616921.6054
ROW00016       187.6000
ROW00017       187.6000
ROW00018       187.6000
ROW00019       187.6000
ROW00020      1404.6237
ROW00021         0.0000
ROW00022      1188.1548
ROW00023       586.4588
ROW00024      2566.6700
ROW00025      1933.3300
ROW00026      1600.0000
ROW00027        53.1542
ROW00028       187.6000
ROW00029       187.6000
ROW00030       420.6400
ROW00031       125.1663
ROW00032        98.9654
ROW00033       701.0346
ROW00034        45.4400
ROW00035       256.4416
```

```
ROW00036        210.3200
ROW00037        210.3200
ROW00038        210.3200
ROW00039       1674.8337
ROW00040       2143.5584
ROW00041       1200.0000
ROW00042       1133.7995
ROW00043       2566.6700
ROW00044       1986.4842
ROW00045       1546.8458
ROW00046        210.3200
ROW00047        210.3200
ROW00048        210.3200
ROW00049        482.0480
ROW00050          0.7842
ROW00051       1723.8594
ROW00052        799.2158
ROW00053         61.4080


 LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO      1      2      1

 ROW          VALUE
FOB00001     617653.4347
ROW00016        187.6000
ROW00017        187.6000
ROW00018        187.6000
ROW00019        187.6000
ROW00020       1404.6237
ROW00021          0.0000
ROW00022       1188.1548
ROW00023        586.4588
ROW00024       2566.6700
ROW00025       1833.0795
ROW00026       1600.0000
ROW00027        153.4046
ROW00028        187.6000
ROW00029        187.6000
ROW00030        420.6400
ROW00031        105.1162
ROW00032        100.2505
ROW00033        900.2505
ROW00034         45.4400
ROW00035        375.2000
ROW00036        210.3200
ROW00037        210.3200
ROW00038        210.3200
ROW00039       1694.8838
ROW00040       2024.8000
ROW00041       1000.7842
ROW00042       1133.7995
ROW00043       2566.6700
ROW00044       2086.7346
ROW00045       1646.5954
ROW00046        210.3200
ROW00047        210.3200
ROW00048        210.3200
ROW00049        482.0480
```

```
      ROW00050       199.2158
      ROW00051      1604.1600
      ROW00052       600.0000
      ROW00053        61.4080


       LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO     1      2      2

       ROW        VALUE
      FOB00001    616921.6054
      ROW00016       187.6000
      ROW00017       187.6000
      ROW00018       187.6000
      ROW00019       187.6000
      ROW00020      1404.6237
      ROW00021         0.0000
      ROW00022      1188.1548
      ROW00023       586.4588
      ROW00024      2566.6700
      ROW00025      1933.3300
      ROW00026      1600.0000
      ROW00027       253.1542
      ROW00028       187.6000
      ROW00029       187.6000
      ROW00030       420.6400
      ROW00031       125.1663
      ROW00032        98.9654
      ROW00033       701.0346
      ROW00034        45.4400
      ROW00035       256.4416
      ROW00036       210.3200
      ROW00037       210.3200
      ROW00038       210.3200
      ROW00039      1674.8337
      ROW00040      2143.5584
      ROW00041      1200.0000
      ROW00042      1133.7995
      ROW00043      2566.6700
      ROW00044      1986.4842
      ROW00045      1546.8458
      ROW00046       210.3200
      ROW00047       210.3200
      ROW00048       210.3200
      ROW00049       482.0480
      ROW00050         0.7842
      ROW00051      1723.8594
      ROW00052       799.2158
      ROW00053        61.4080


       LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO     2      1      1

       ROW        VALUE
      FOB00001    617323.7260
      ROW00016       187.6000
      ROW00017       187.6000
      ROW00018       187.6000
      ROW00019       187.6000
```

```
ROW00020      1604.6237
ROW00021         0.0000
ROW00022      1188.1548
ROW00023       586.4588
ROW00024      2566.6700
ROW00025      1786.4842
ROW00026      1600.0000
ROW00027       146.8458
ROW00028       187.6000
ROW00029       187.6000
ROW00030       420.6400
ROW00031        95.7972
ROW00032        46.5954
ROW00033       900.2505
ROW00034        45.4400
ROW00035       319.2856
ROW00036       210.3200
ROW00037       210.3200
ROW00038       210.3200
ROW00039      1704.2028
ROW00040      2080.7144
ROW00041      1000.7842
ROW00042      1133.7995
ROW00043      2566.6700
ROW00044      2133.3300
ROW00045      1600.0000
ROW00046       210.3200
ROW00047       210.3200
ROW00048       210.3200
ROW00049       482.0480
ROW00050       199.2158
ROW00051      1660.0744
ROW00052       600.0000
ROW00053        61.4080


 LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO      2      1      2

 ROW          VALUE
FOB00001    616921.6054
ROW00016       187.6000
ROW00017       187.6000
ROW00018       187.6000
ROW00019       187.6000
ROW00020      1604.6237
ROW00021         0.0000
ROW00022      1188.1548
ROW00023       586.4588
ROW00024      2566.6700
ROW00025      1933.3300
ROW00026      1600.0000
ROW00027        53.1542
ROW00028       187.6000
ROW00029       187.6000
ROW00030       420.6400
ROW00031       125.1663
ROW00032        98.9654
ROW00033       701.0346
```

```
      ROW00034           45.4400
      ROW00035          256.4416
      ROW00036          210.3200
      ROW00037          210.3200
      ROW00038          210.3200
      ROW00039         1674.8337
      ROW00040         2143.5584
      ROW00041         1200.0000
      ROW00042         1133.7995
      ROW00043         2566.6700
      ROW00044         1986.4842
      ROW00045         1546.8458
      ROW00046          210.3200
      ROW00047          210.3200
      ROW00048          210.3200
      ROW00049          482.0480
      ROW00050            0.7842
      ROW00051         1723.8594
      ROW00052          799.2158
      ROW00053           61.4080


       LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO      2      2      1

       ROW         VALUE
      FOB00001   617653.4347
      ROW00016          187.6000
      ROW00017          187.6000
      ROW00018          187.6000
      ROW00019          187.6000
      ROW00020         1604.6237
      ROW00021            0.0000
      ROW00022         1188.1548
      ROW00023          586.4588
      ROW00024         2566.6700
      ROW00025         1833.0795
      ROW00026         1600.0000
      ROW00027          153.4046
      ROW00028          187.6000
      ROW00029          187.6000
      ROW00030          420.6400
      ROW00031          105.1162
      ROW00032          100.2505
      ROW00033          900.2505
      ROW00034           45.4400
      ROW00035          375.2000
      ROW00036          210.3200
      ROW00037          210.3200
      ROW00038          210.3200
      ROW00039         1694.8838
      ROW00040         2024.8000
      ROW00041         1000.7842
      ROW00042         1133.7995
      ROW00043         2566.6700
      ROW00044         2086.7346
      ROW00045         1646.5954
      ROW00046          210.3200
      ROW00047          210.3200
```

```
ROW00048      210.3200
ROW00049      482.0480
ROW00050      199.2158
ROW00051     1604.1600
ROW00052      600.0000
ROW00053       61.4080


 LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO     2     2     2

 ROW         VALUE
FOB00001   616921.6054
ROW00016      187.6000
ROW00017      187.6000
ROW00018      187.6000
ROW00019      187.6000
ROW00020     1604.6237
ROW00021        0.0000
ROW00022     1188.1548
ROW00023      586.4588
ROW00024     2566.6700
ROW00025     1933.3300
ROW00026     1600.0000
ROW00027      253.1542
ROW00028      187.6000
ROW00029      187.6000
ROW00030      420.6400
ROW00031      125.1663
ROW00032       98.9654
ROW00033      701.0346
ROW00034       45.4400
ROW00035      256.4416
ROW00036      210.3200
ROW00037      210.3200
ROW00038      210.3200
ROW00039     1674.8337
ROW00040     2143.5584
ROW00041     1200.0000
ROW00042     1133.7995
ROW00043     2566.6700
ROW00044     1986.4842
ROW00045     1546.8458
ROW00046      210.3200
ROW00047      210.3200
ROW00048      210.3200
ROW00049      482.0480
ROW00050        0.7842
ROW00051     1723.8594
ROW00052      799.2158
ROW00053       61.4080


 PROBLEM NODE =     1-    1-   1


 OPTIMUM LP OBJECTIVE VALUE = 832935.39
 AFTER   18 PASSES FROM NODE TO NODE
 AND AFTER  118 ITERATIONS
```

```
      OPTIMAL VALUE REACHED AT THE POINT

      VARIABLE           STATUS      VALUE
     FOBOOOO1                1      832935.3907
     ROWOOOO1                0           0.0
     ROWOOOO2                0           0.0
     ROWOOOO3               -1           0.0
     ROWOOOO4               -1           0.0
     ROWOOOO5                0           0.0
     ROWOOOO6                7          62.2967
     ROWOOOO7                0           0.0
     ROWOOOO8                0           0.0
     ROWOOOO9               10        1092.2453
     ROWOOO10                0           0.0
     ROWOOO11                0           0.0
     ROWOOO12               -1           0.0
     ROWOOO13               -1           0.0
     ROWOOO14               -1           0.0
     ROWOOO15               -1           0.0
     COLOOOO1                0           0.0
     COLOOOO2                2         158.0000
     COLOOOO3                3         158.0000
     COLOOOO4                0           0.0
     COLOOOO5                4         158.0000
     COLOOOO6                5         158.0000
     COLOOOO7               18         903.3814
     COLOOOO8                6        2504.3733
     COLOOOO9               19        1430.0002
     COLOOO10                8         462.2451
     COLOOO11                0           0.0
     COLOOO12                0           0.0
     COLOOO13                9        1600.0000
     COLOOO14                0           0.0
     COLOOO15               11         158.0000
     COLOOO16               12         158.0000
     COLOOO17               13         375.2000
     COLOOO18               14         195.3763
     COLOOO19               15        2400.0000
     COLOOO20               16          11.8452
      OPTIMAL DUAL VALUES
          1.00   1141.23    169.40 -1195.90  -224.07     -1.05      0.0     -8.15
          8.15      0.0   -1220.06  -248.22 -1723.41     -2.87      1.17     1.31
          0.02      0.98      0.00      0.0


      PROBLEM NODE =     2-    1-   1


      OPTIMUM LP OBJECTIVE VALUE = 617627.43
      AFTER    18 PASSES FROM NODE TO NODE
      AND AFTER  118 ITERATIONS
      OPTIMAL VALUE REACHED AT THE POINT

      VARIABLE           STATUS      VALUE
     FOBOOOO1                1      617627.4347
     ROWOOO16                0           0.0
     ROWOOO17                0           0.0
     ROWOOO18               -1           0.0
```

1161	ROW00019	-1	0.0
1162	ROW00020	6	1604.6237
1163	ROW00021	7	0.0000
1164	ROW00022	8	1188.1548
1165	ROW00023	0	0.0
1166	ROW00024	0	0.0
1167	ROW00025	0	0.0
1168	ROW00026	0	0.0
1169	ROW00027	13	-99.7495
1170	ROW00028	-1	0.0
1171	ROW00029	0	0.0
1172	ROW00030	-1	0.0
1173	ROW00031	0	0.0
1174	ROW00032	-1	0.0
1175	ROW00033	0	0.0
1176	ROW00034	20	45.4400
1177	ROW00035	-1	0.0
1178	ROW00036	-1	0.0
1179	ROW00037	0	0.0
1180	ROW00038	0	0.0
1181	ROW00039	25	1651.8838
1182	ROW00040	26	375.2000
1183	ROW00041	27	1200.7842
1184	ROW00042	-1	0.0
1185	ROW00043	0	0.0
1186	ROW00044	0	0.0
1187	ROW00045	31	153.4046
1188	ROW00046	0	0.0
1189	ROW00047	0	0.0
1190	ROW00048	0	0.0
1191	ROW00049	0	0.0
1192	ROW00050	-1	0.0
1193	ROW00051	-1	0.0
1194	ROW00052	-1	0.0
1195	ROW00053	39	61.4080
1196	COL00021	0	0.0
1197	COL00022	2	187.6000
1198	COL00023	3	187.6000
1199	COL00024	0	0.0
1200	COL00025	4	187.6000
1201	COL00026	5	187.6000
1202	COL00027	11	2033.0795
1203	COL00028	10	2566.6700
1204	COL00029	0	0.0
1205	COL00030	19	700.2505
1206	COL00031	9	586.4588
1207	COL00032	0	0.0
1208	COL00033	12	1600.0000
1209	COL00034	0	0.0
1210	COL00035	14	187.6000
1211	COL00036	15	187.6000
1212	COL00037	16	420.6400
1213	COL00038	17	145.1162
1214	COL00039	18	2024.8000
1215	COL00040	38	-0.7842
1216	COL00041	0	0.0
1217	COL00042	21	210.3200
1218	COL00043	22	210.3200

```
COL00044              0         0.0
COL00045             23       210.3200
COL00046             24       210.3200
COL00047             36      1886.7346
COL00048             29      2566.6700
COL00049              0         0.0
COL00050             32       800.0000
COL00051             28      1133.7995
COL00052              0         0.0
COL00053             30      1646.5954
COL00054             33       210.3200
COL00055              0         0.0
COL00056             34       210.3200
COL00057             35       482.0480
COL00058              0         0.0
COL00059             37      1604.1600
COL00060              0         0.0
 OPTIMAL DUAL VALUES
     0.0      1.90      1.90     -1.40     -1.40      0.0       0.0       0.0
     0.0      0.0       0.0       0.0      -1.00      0.0       0.0      -3.80
     0.0      0.0       1.00      0.0       3.80      3.80     -2.80     -2.80
     0.0      0.0       0.0       0.0       0.0       0.0       0.0      -2.00
     0.0      0.0       0.0       0.0       0.0       2.00      0.0
```

```
BASIS
    FOB00001   FOB00001
    ROW00006   ROW00006
    ROW00009   ROW00009
    COL00002   ROW00001
    COL00003   ROW00002
    COL00005   ROW00003
    COL00006   ROW00004
    COL00008   ROW00005
    COL00010   ROW00007
    COL00013   ROW00008
    COL00015   ROW00010
    COL00016   ROW00011
    COL00017   ROW00012
    COL00018   ROW00013
    COL00019   ROW00014
    COL00020   ROW00015
BASIS
    FOB00001   FOB00001
    ROW00020   ROW00020
    ROW00021   ROW00021
    ROW00022   ROW00022
    ROW00027   ROW00027
    ROW00034   ROW00034
    ROW00039   ROW00039
    ROW00040   ROW00040
    ROW00041   ROW00041
    ROW00045   ROW00045
    ROW00053   ROW00053
    COL00022   ROW00016
    COL00023   ROW00017
    COL00025   ROW00018
    COL00026   ROW00019
    COL00027   ROW00025
    COL00028   ROW00024
    COL00030   ROW00033
    COL00031   ROW00023
    COL00033   ROW00026
    COL00035   ROW00028
    COL00036   ROW00029
    COL00037   ROW00030
    COL00038   ROW00031
    COL00039   ROW00032
    COL00040   ROW00052
    COL00042   ROW00035
    COL00043   ROW00036
    COL00045   ROW00037
    COL00046   ROW00038
    COL00047   ROW00050
    COL00048   ROW00043
    COL00050   ROW00046
    COL00051   ROW00042
    COL00053   ROW00044
    COL00054   ROW00047
    COL00056   ROW00048
    COL00057   ROW00049
    COL00059   ROW00051
```

```

      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
       COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
       COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
      1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
      2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
       INTEGER MREG,HREG,VREG
       DIMENSION MREG(350),HREG(350),VREG(350)
       EQUIVALENCE (MREG(1),YTEMP(1)),(HREG(1),YTEMP(301))
      1 ,(VREG(1),YTEMP1(6))
C
C    SET PARAMETERS
C
       NETA = 0
       NLETA = 0
       NUETA = 0
       NELEM = 0
       NLELEM = 0
       NUELEM = 0
       NABOVE = 0
       LE(1) = 1
       LR1 = 1
       KR1 = 0
       JC=JCUR
       J2=JPER(2)
       J3=JPER(3)
       LR4 = NROW(JC,J2,J3) + 1
C
C    PUT SLACKS AND ARTIFICIALS IN PART 4 AND REST IN PART 1
C
       KNR=NROW(JC,J2,J3)
       DO 100 I = 1,KNR
          IF (JH(I,JC,J2,J3) .GT. NROW(JC,J2,J3)) GO TO 50
          LR4 = LR4 - 1
          MREG(LR4) = JH(I,JC,J2,J3)
          VREG(LR4) = JH(I,JC,J2,J3)
          GO TO 90
50        KR1 = KR1 + 1
          VREG(KR1) = JH(I,JC,J2,J3)
90        HREG(I) = -1
          JH(I,JC,J2,J3) = 0
100       CONTINUE
C
       KR3 = LR4 -1
       LR3 = LR4
C
       KNR=NROW(JC,J2,J3)
       DO 200 I = LR4,KNR
          IR = MREG(I)
          HREG(IR)= 0
          JH(IR,JC,J2,J3) = IR
          KINBAS(IR,JC,J2,J3) = IR
200       CONTINUE
C
C    PULL OUT VECTORS BELOW BUMP AND GET ROW COUNTS
C
```

```
          NBNONZ = NROW(JC,J2,J3) - LR4 + 1
          IF (KR1 .EQ. 0) GO TO 1190
          J = LR1
 210      IV = VREG(J)
          LL = LA(IV,JC,J2,J3)
          KK = LA(IV+1,JC,J2,J3) -1
          IRCNT = 0
          DO 220 I = LL,KK
             NBNONZ = NBNONZ + 1
             IR = IA(I,JC,J2,J3)
             IF (HREG(IR) .GE. 0) GO TO 220
             IRCNT = IRCNT + 1
             HREG(IR) = HREG(IR) - 1
             IRP = IR
 220         CONTINUE
          IF (IRCNT - 1) 230,250,300
 230      WRITE(6,8000)
 8000     FORMAT(16HMATRIX SINGULAR )
          KINBAS(IV,JC,J2,J3) = 0
          VREG(J) = VREG(KR1)
          KR1 = KR1 - 1
          IF (J .GT. KR1) GO TO 310
          GO TO 210
 C
 250      VREG(J) = VREG(KR1)
          KR1 = KR1 - 1
          LR3 = LR3 - 1
          VREG(LR3) = IV
          MREG(LR3) = IRP
          HREG(IRP) = 0
          JH(IRP,JC,J2,J3) = IV
          KINBAS(IV,JC,J2,J3) = IRP
          IF (J .GT. KR1) GO TO 310
          GO TO 210
 300      IF (J .GE. KR1) GO TO 310
          J = J+1
          GO TO 210
 C
 C                     PULL OUT REMAINING VECTORS ABOVE AND BELOW THE
 C                     BUMP AND ESTABLISH MERIT COUNTS OF COLUMNS
 C
 310      NVREM = 0
          IF(KR1 .EQ. 0) GO TO 1190
          J = LR1
 320      IV = VREG(J)
          LL = LA(IV,JC,J2,J3)
          KK = LA(IV+1,JC,J2,J3) - 1
          IRCNT = 0
          DO 800 I = LL,KK
             IR = IA(I,JC,J2,J3)
             IF(HREG(IR) .NE. -2) GO TO 400
 C
 C                PIVOT ABOVE BUMP (PART OF L)
 C
             NABOVE = NABOVE + 1
             IROWP = IR
             CALL UNPACK(IV)
             CALL WRETA
```

```
            NLETA = NETA
            JH(IR,JC,J2,J3) = IV
            KINBAS(IV,JC,J2,J3) = IR
            VREG(J) = VREG(KR1)
            KR1 = KR1 - 1
            NVREM = NVREM + 1
            HREG(IR) = IV
            GO TO 940
C
400         IF (HREG(IR) .GE. 0) GO TO 800
            IRCNT = IRCNT + 1
            IRP = IR
800         CONTINUE
C
         IF (IRCNT - 1) 810,900,1000
810      WRITE(6,8000)
         KINBAS(IV,JC,J2,J3) = 0
         VREG(J) = VREG(KR1)
         NVREM = NVREM + 1
         KR1 = KR1 - 1
         IF (J .GT. KR1) GO TO 1010
         GO TO 320
C
C   PUT VECTOR BELOW BUMP
C
900      VREG(J) = VREG(KR1)
         NVREM = NVREM + 1
         KR1 = KR1 - 1
         LR3 = LR3 - 1
         VREG(LR3) = IV
         MREG(LR3) = IRP
         HREG(IRP) = 0
         JH(IRP,JC,J2,J3) = IV
         KINBAS(IV,JC,J2,J3) = IRP
C
C   CHANGE ROW COUNTS
C
940      DO 950 II = LL,KK
            IIR = IA(II,JC,J2,J3)
            IF (HREG(IIR) .GE. 0) GO TO 950
            HREG(IIR) = HREG(IIR) + 1
950         CONTINUE
         IF (J .GT. KR1) GO TO 1010
         GO TO 320
1000     IF (J .GE. KR1) GO TO 1010
         J = J+1
         GO TO 320
1010     IF(NVREM .GT. 0) GO TO 310
C
C   GET MERIT COUNTS
C
1020     IF (KR1 .EQ. 0) GO TO 1190
         DO 1100 J = LR1,KR1
            IV = VREG(J)
            LL = LA(IV,JC,J2,J3)
            KK = LA(IV+1,JC,J2,J3) - 1
            IMCNT = 0
            DO 1050 I = LL,KK
```

```
                IR = IA(I,JC,J2,J3)
                IF (HREG(IR) .GE. O) GO TO 1050
                IMCNT = IMCNT - (HREG(IR) +1)
 1050           CONTINUE
            MREG(J) = IMCNT
 1100       CONTINUE
C
C    SORT COLUMNS INTO MERIT ORDER USING SHELL SORT
C
      ISD = 1
 1106 IF (KR1 .LT. 2*ISD) GO TO 1108
      ISD = 2*ISD
      GO TO 1106
 1108 ISD = ISD - 1
C           END OF INITIALIZATION
 1101 IF (ISD .LE. O) GO TO 1107
      ISK = 1
 1102 ISJ = ISK
      ISL = ISK + ISD
      ISY = MREG(ISL)
      ISZ = VREG(ISL)
 1103 IF (ISY .LT. MREG(ISJ)) GO TO 1104
 1105 ISL = ISJ + ISD
      MREG(ISL) = ISY
      VREG(ISL) = ISZ
      ISK = ISK + 1
      IF ((ISK + ISD) .LE. KR1) GO TO 1102
      ISD = (ISD - 1) / 2
      GO TO 1101
 1104 ISL = ISJ + ISD
      MREG(ISL) = MREG(ISJ)
      VREG(ISL) = VREG(ISJ)
      ISJ = ISJ - ISD
      IF (ISJ .GT. O) GO TO 1103
      GO TO 1105
 1107 CONTINUE
C
C    END OF SORT ROUTINE
C    PUT OUT BELOW BUMP ETAS (PART OF U)
C
 1190 NSLCK = O
      NBELOW = O
      NELAST = NEMAX
      NTLAST = NTMAX
      LE(NTLAST + 1) = NELAST + 1
C
      LR = LR3
      IF (LR3 .GE. LR4) LR = LR4
      IF (LR .GT. NROW(JC,J2,J3)) GO TO 2050
      JK = NROW(JC,J2,J3) + 1
      KNR=NROW(JC,J2,J3)
      DO 2000 JJ= LR,KNR
         JK = JK - 1
         IV = VREG(JK)
         I = MREG(JK)
         NBELOW = NBELOW + 1
         IF (IV .GT. NROW(JC,J2,J3)) GO TO 1200
         NSLCK = NSLCK + 1
```

```
1200      LL = LA(IV,JC,J2,J3)
          KK = LA(IV+1,JC,J2,J3) -1
          IF (KK .GT. LL) GO TO 1300
1250      IF (ABS(A(LL,JC,J2,J3) - 1.) .LE. ZTOLZE) GO TO 2000
C
1300      NUETA = NUETA + 1
          DO 1400 J = LL,KK
             IR = IA(J,JC,J2,J3)
             IF (IR .EQ. I)GO TO 1390
             IE(NELAST) = IR
             E(NELAST) = A(J,JC,J2,J3)
             NELAST = NELAST - 1
             NUELEM = NUELEM + 1
             GO TO 1400
1390         EP = A(J,JC,J2,J3)
1400         CONTINUE
          IE(NELAST) = I
          E(NELAST) = EP
          LE(NTLAST) = NELAST
          NELAST = NELAST - 1
          NTLAST = NTLAST - 1
          NUELEM = NUELEM + 1
2000      CONTINUE
2050   IF(KR1 .EQ. 0) GO TO 3500
C
C    DO L-U DECOMPOSITION OF BUMP
C
       DO 3000 J = LR1,KR1
          IV = VREG(J)
          CALL UNPACK(IV)
          CALL FTRAN(2)
          IROWP = 0
          IRCMIN = -999999
       KNR=NROW(JC,J2,J3)
          DO 2100 I = 1,KNR
             IF (DABS(Y(I)) .LE. ZTOLPV) GO TO 2100
             IF (HREG(I) .GE.0) GO TO 2100
             IF (HREG(I) .LE. IRCMIN) GO TO 2100
             IRCMIN = HREG(I)
             IROWP = I
2100         CONTINUE
          IF (IROWP .GT. 0) GO TO 2150
          WRITE(6,8000)
          KINBAS(IV,JC,J2,J3) = 0
          GO TO 3000
C
2150      INCR = HREG(IROWP) + 3
C
C    WRITE L AND U ETAS
C
          IF (J .EQ. KR1) GO TO 2160
          NELEM = NELEM + 1
          IE(NELEM) = IROWP
          E(NELEM) = Y(IROWP)
       KNR=NROW(JC,J2,J3)
2160      DO 2300 I = 1,KNR
             IF (I .EQ. IROWP) GO TO 2300
             IF(DABS(Y(I)) .LE. ZTOLZE) GO TO 2300
```

```
                 IF (HREG(I) .GE. 0) GO TO 2200
C
C   L ETA ELEMENTS
C
                 NELEM = NELEM + 1
                 IE(NELEM) = I
                 E(NELEM) = Y(I)
                 GO TO 2300
C
C   U ETA ELEMENTS
C
2200             IE(NELAST) = I
                 E(NELAST) = Y(I)
                 NELAST = NELAST - 1
                 NUELEM = NUELEM + 1
2300             CONTINUE
C
              JH(IROWP,JC,J2,J3) = IV
              KINBAS(IV,JC,J2,J3) = IROWP
              NUETA = NUETA + 1
              IE(NELAST) = IROWP
              IF (J .NE. KR1) GO TO 2330
              E(NELAST) = Y(IROWP)
              GO TO 2340
2330          E(NELAST) = 1.
              NETA = NETA + 1
              LE(NETA+1) = NELEM + 1
2340          NUELEM = NUELEM + 1
              LE(NTLAST) = NELAST
              NELAST = NELAST - 1
              NTLAST = NTLAST - 1
C
C   UPDATE ROW COUNTS
C
           KNR=NROW(JC,J2,J3)
              DO 2350 I = 1,KNR
                 IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2350
                 IF (HREG(I) .GE. 0) GO TO 2350
                 HREG(I) = HREG(I) - INCR
                 IF (HREG(I) .GE. 0) HREG(I) = -1
2350             CONTINUE
              HREG(IROWP) = 0
3000          CONTINUE
C
C   MERGE L AND U ETAS
C
3500       NLETA = NETA
           NETA = NLETA + NUETA
           NLELEM = NELEM
           NELEM = NLELEM + NUELEM
           IF (NUELEM .EQ. 0) GO TO 3550
           CALL SHFTE
C
C   INSERT SLACKS FOR DELETED COLUMNS
C
           KNR=NROW(JC,J2,J3)
3550       DO 3600 I = 1,KNR
              IF (JH(I,JC,J2,J3) .NE. 0) GO TO 3600
```

```
                JH(I,JC,J2,J3) = I
                IROWP = I
                CALL UNPACK(IROWP)
                CALL FTRAN(1)
                CALL WRETA
      3600      CONTINUE
      C
      C   UPDATE X
      C
            CALL SHIFTR(1,3)
            KNC=NCOL(JC,J2,J3)
            DO 9000 J=1,KNC
                IF (KINBAS(J,JC,J2,J3)) 8600,8700,9000
      8600      DE = XUB(J,JC,J2,J3)
                GO TO 8750
      8700      DE = XLB(J,JC,J2,J3)
      8750      LL = LA(J,JC,J2,J3)
                KK = LA(J+1,JC,J2,J3) - 1
                DO 8800 I=LL,KK
                   IR = IA(I,JC,J2,J3)
      8800         Y(IR) = Y(IR) - A(I,JC,J2,J3)*DE
      9000      CONTINUE
            CALL FTRAN(1)
            CALL SHIFTR(3,2)
            RETURN
            END
      C-----------------------------------------------------------------------
            SUBROUTINE UNPACK(IV)
      C
      C         EXPANDS COMPRESSED MATRIX COLUMNS
      C         SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
      C         BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
      C         ***DESCRIPTION OF PARAMETERS***
      C         IV = PARAMETER INDEXING COLUMN TO BE EXPANDED (INPUT)
      C
            IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
           1  INTEGER*4 (I-N,Q)
            INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
            DOUBLE PRECISION E(3000)
            REAL A(3000,3,3,1)
            COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
           1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
           1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
           2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
           2  ),NETA,
           3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
           3  602,3,3,1),
           4  LE(1002),IA(3000,3,3,1),IE(3000),
           5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

           6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
           7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
            COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
            COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
           1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
           2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      C
            JC=JCUR
```

```
      J2=JPER(2)
      J3=JPER(3)
      IF(JC.EQ.NPER)J2=1
      KNR=NROW(JC,J2,J3)
      DO 100 I = 1,KNR
         Y(I) = 0.
100      CONTINUE
C
      LL = LA(IV,JC,J2,J3)
      KK = LA(IV+1,JC,J2,J3) - 1
      DO 200 I = LL,KK
         IR = IA(I,JC,J2,J3)
         Y(IR) = A(I,JC,J2,J3)
200      CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE BUNPCK(IV)
C
C        EXPANDS COMPRESSED MATRIX COLUMNS
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C        ***DESCRIPTION OF PARAMETERS***
C        IV = PARAMETER INDEXING COLUMN TO BE EXPANDED (INPUT)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNRO=NROW(JC+1,1,1)
      DO 100 I = 1,KNRO
         Y(I) = 0.
100      CONTINUE
C
      LL = LBN(IV,JC,J2,J3)
      KK = LBN(IV+1,JC,J2,J3) - 1
      DO 200 I = LL,KK
```

```
              IR = IBN(I,JC,J2,J3)
              Y(IR) = ABN(I,JC,J2,J3)
  200         CONTINUE
          RETURN
          END
C------------------------------------------------------------------------
          SUBROUTINE SHFTE
C
C             SUBROUTINE FOR INVERT
C             SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
C             BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
          IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
         1  INTEGER*4 (I-N,Q)
          INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
          DOUBLE PRECISION E(3000)
          REAL A(3000,3,3,1)
C
          COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
         1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
          COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
         1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
         1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
         2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
         2  ),NETA,
         3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
         3  602,3,3,1),
         4  LE(1002),IA(3000,3,3,1),IE(3000),
         5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

         6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
         7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
          COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
          COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
         1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
         2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C                   SHIFT IE AND E OF U ELEMENTS
C
          NF = NEMAX - NUELEM + 1
          INCR = 0
          DO 1000 I = NF,NEMAX
              INCR = INCR + 1
              IE(NLELEM + INCR) = IE(I)
              E(NLELEM + INCR) = E(I)
 1000         CONTINUE
C
          IDIF = NEMAX - NLELEM - NUELEM
          NF = NTMAX - NUETA + 1
          INCR = 0
          DO 2000 I = NF,NTMAX
              INCR = INCR + 1
              LE(NLETA + INCR) = LE(I) - IDIF
 2000         CONTINUE
          LE(NETA+1) = NELEM + 1
          RETURN
          END
C------------------------------------------------------------------------
```

```
      SUBROUTINE UPBETA
C
C        UPDATES RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
C        CURRENT SIMPLEX PIVOT
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),,
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNR=NROW(JC,J2,J3)
      DO 1000 I=1,KNR
 1000    X(I,JC,J2,J3) = X(I,JC,J2,J3) - Y(I)*DP
      IF (NPIVOT .EQ. 1) GO TO 2000
      KINBAS(JCOLP,JC,J2,J3) = -(KINBAS(JCOLP,JC,J2,J3) + 1)
      RETURN
 2000 X(IROWP,JC,J2,J3) = DE
      IVOUT = JH(IROWP,JC,J2,J3)
      KINBAS(JCOLP,JC,J2,J3) = IROWP
      KINBAS(IVOUT,JC,J2,J3) = IPTYPE
      JH(IROWP,JC,J2,J3) = JCOLP
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE NORMAL(ITSINV)
C
C        SERVES AS MASTER PROGRAM FOR LINEAR PROGRAMMING COMPONENT
C        (REVISED,PRIMAL-SIMPLEX METHOD) OF BRANCH-AND-BOUND ROUTINE BB
C        SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
C        BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
C        ***DESCRIPTION OF PARAMETERS***
C        ITSINV = NUMBER OF SIMPLEX ITERATIONS SINCE LAST BASIS
C                 INVERSION (INPUT/OUTPUT)
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
```

```
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
C
      COMMON/BL5/DRES,ICNAM
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPI
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      NELEM=NELM(JC,J2,J3)
      IF (ITSINV .LT. INVFRQ) GO TO 1500
 1000 CALL INVERT
      WRITE(6,1001)
 1001 FORMAT(' ITNS',2X,'RRES',5X,' PIV',6X,' IN',3X,'OUT',2X,
     1 'OBJ',7X,'CMIN',7X,'CMAX',7X,'THET',3X,'NINF',1X,'NOPT')
      ITSINV = 0
C
C                 SIMPLEX CYCLE
C
 1500 CALL FORMC
C                 TRANSFER AFTER PHASE 1 FOR FIRST PASS
      IF((MSTAT .EQ. QF) .AND. (JPASS .EQ. 1)) GO TO 6000
      CALL BTRAN
      CALL PRICE
      IF (JCOLP .GT. 0) GO TO 3000
      IF (MSTAT .EQ. QI) GO TO 2000
      MSTAT = QBL
      GO TO 6000
 2000 MSTAT = QN
      GO TO 6000
 3000 CALL UNPACK(JCOLP)
      CALL FTRAN(1)
      CALL CHUZR
      CALL UPBETA
      ITCNT = ITCNT + 1
      ITSINV = ITSINV + 1
      IF (NPIVOT .EQ. 0) GO TO 4010
      IF (NELEM .GT. (NEMAX-NROW(JC,J2,J3))) GO TO 1000
```

```
      CALL WRETA
      CALL RHCHCK
      IF(ITCNT.GE.ITRFRQ) WRITE(6,4011) ITCNT
4011  FORMAT(' MORE THAN',I7,' ITERATIONS')
      IF(ITCNT.GE.ITRFRQ) JCUR=NPER+1
      IF (ITCNT .GE. ITRFRQ) GO TO 6000
      WRITE(6,1002)ITCNT,DRES,APV,JCOLP,IROWP,X(1,JC,J2,J3),CMIN,CMAX
     1 ,DP,NINF,NOPT
 1002 FORMAT(I4,1X,E9.2,1X,E9.2,1X,I4,1X,I4,1X,E10.2,1X,E9.2,1X,E9.2,1X,
     1 E9.2,1X,I4,1X,I4)
      IF(DRES.GT.10*ZTOLZE) GO TO 1000
4010  IF (ITSINV .GE. INVFRQ) GO TO 1000
      GO TO 1500
6000  RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE NDCOM
C
C  THIS SUBROUTINE PERFORMS ALL THE NECESSARY STOCHASTIC DECOMP
C  OSITION STEPS.
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C        FOR ALL FEASIBLE, PASSBACK
C
      IF(JPASS .EQ. 2) GO TO 70
C
C        IF CURRENT FEASIBLE GO ON
      WRITE(6,300) MSTAT
300   FORMAT(' STATUS AT NDCOM=',A4)
      WRITE(6,301) JCUR
301   FORMAT(' AT BEG OF NDCOM, JCUR=',I6)
C
      IF(MSTAT.NE. QN) GO TO 20
C
```

```
C        HERE TOTAL PROBLEM IS INFEASIBLE
C
      IF(JCUR .EQ. 1) JCUR=NPER+1
      IF(JCUR .EQ. NPER+1) RETURN
      WRITE(6,303) JCUR
303   FORMAT(' INFEASIBLE AT JCUR = ',I6)
C
C        ADD CUT AND RETURN
C
      WRITE(6,201)
 201  FORMAT('MAKING A FEASIBILTIY CUT')
      CALL FEASCT
      JPER(JCUR)=1
      JCUR=JCUR-1
      RETURN
 20   CONTINUE
C
C        FEASIBLE, SO CONTINUE DOWN
C
      IF(JPER(JCUR) .LT. NND(JCUR)) GO TO 30
      K=JCUR
C
C        ITERATE TO FIND THE PROPER INDEX TO CHANGE
C
 25   CONTINUE
      K=K-1
      IF(K .LE. 1) GO TO 28
      IF(JPER(K) .EQ. NND(K)) GO TO 25
      JPER(K)=JPER(K)+1
      KK=K+1
      DO 26 I=KK,JCUR
      JPER(I)=1
 26   CONTINUE
      RETURN
C
C        THERE, YOU'VE FINISHED THE CURRENT PERIOD
C
   28 CONTINUE
C
C        MULTIPLY OUT THE CURRENT PERIOD, CARRY TO NEXT
C
      IF(JCUR .GT. 1) GO TO 285
      CALL BPRODX
      CALL FRMRHS
      GO TO 290
  285 CONTINUE
      IF(JCUR .EQ. NPER) GO TO 290
      NNDD=NND(JCUR)
      DO 280 L=1,NNDD
      JPER(JCUR)=L
      CALL BPRODX
      CALL FRMRHS
  280 CONTINUE
  290 CONTINUE
      JCUR=JCUR+1
      DO 29 I=1,JCUR
      JPER(I)=1
   29 CONTINUE
```

```
      WRITE(6,302) JCUR
  302 FORMAT(' AT END OF ND, JCUR='.I6)
      IF(JCUR .LE. NPER) RETURN
      JPASS=2
      JCUR=NPER
      RETURN
   30 CONTINUE
      JPER(JCUR)=JPER(JCUR)+1
      RETURN
C
C        NOW, FOR THE BACK PASS
C
   70 CONTINUE
C
C        THIS SAYS SET THE OPTIMAL FLAG AT FALSE
C
      NFLG = 0
C
C        CHECK IF FEASIBLE
C
      IF(MSTAT .NE. QN) GO TO 80
C
C        IF FIRST PERIOD INFEASIBLE
C
      IF(JCUR .EQ. 1) WRITE(6,72)
   72 FORMAT('INFEASIBLE ON PASS 2')
      IF(JCUR .EQ. 1) STOP
C
C        FEASIBLE CUT
C
      CALL FEASCT
      JPER(JCUR)=1
      JCUR=JCUR-1
      RETURN
C
C        HERE IT'S FEASIBLE
C
   80 CONTINUE

C
C        GET RID OF CUTS YOU DON'T NEED
C
      CALL DLETCT
      IF((JCUR .EQ. NPER-1).AND.(NFLG.EQ.1)) GO TO 85
C
C        THIS PART SAYS MOVE ON DOWN TO NEXT PERIOD
C
      CALL BPRODX
      CALL FRMRHS
      JCUR=JCUR+1
      DO 82 I=JCUR,NPER
      JPER(I)=1
   82 CONTINUE
   81 CONTINUE
      JJC=JCUR+1
      DO 83 I=JJC,NPER
      JPER(I)=1
   83 CONTINUE
```

```
      NFLG=0
      RETURN
   85 CONTINUE
C
C         CHECK FOR PERIOD CURRENTLY OPTIMAL
C
      IF(JPER(NPER-1) .EQ. NND(NPER-1)) GO TO 90
      JPER(NPER-1)=JPER(NPER-1)+1
      NFLG=0
      RETURN
   90 CONTINUE
C
C         CHECK LAST PERIOD OPTIMALITY
C
      NFLG=0
      IF(NTH(1,1,1).EQ.1) CALL OPTCHK(2)
      IF(NFLG .EQ. 1) GO TO 95
C
C         THERE, IT'S NOT OPTIMAL, GO BACK
C
      CALL LKHDCT(NPER-1)
      JCUR=JCUR-1
      GO TO 81
   95 CONTINUE
      K=JCUR
C
C         ITERATE BACK THROUGH EACH PERIOD TO CHECK OPTIMALITY
C
   96 CONTINUE
      NFLG=0
      K=K-1
C
C         IF K=1, YOU'RE DONE
C
      IF(K .EQ. 1) JCUR=NPER+1
      IF(K .EQ. 1) RETURN
      JPER(K)=JPER(K)+1
      IF(JPER(K) .LE. NND(K)) GO TO 101
      JPER(K)=JPER(K)-1
C
C         CHECK IF OPTIMAL
C
      J2=JPER(2)
      J3=JPER(3)
      IF((K .EQ. 3) .AND. (NTH(2,J2,1) .EQ. 1)) CALL OPTCHK(K)
      IF((K .EQ. 2) .AND. (NTH(1,1,1) .EQ. 1)) CALL OPTCHK(K)
C
C         YES, THEN GO BACK
C
      IF(NFLG .EQ. 1) GO TO 96
C
C         NO, THEN PLACE A CUT
C
      CALL LKHDCT(K)
      JCUR=K-1
      GO TO 81
  101 CONTINUE
      KK=K+1
```

```
      DO 102 I=KK,JCUR
      JPER(I)=1
  102 CONTINUE
      JCUR=K
      RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE STRPRT
C
C  REPORTS ON THE CURRENT STATUS
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      KNR=NROW(JC,J2,J3)
      KNC=NCOL(JC,J2,J3)
      WRITE(6,11) ITCNT
      WRITE(6,10) NPASS
      WRITE(6,1) JC,J2,J3
      WRITE(6,2)
      WRITE(6,3)(X(I,JC,J2,J3),I=1,KNR)
      WRITE(6,4)
      WRITE(6,5)(JH(I,JC,J2,J3),I=1,KNR)
      WRITE(6,6)
      WRITE(6,5)(KINBAS(I,JC,J2,J3),I=1,KNC)
      WRITE(6,7)
      WRITE(6,3)(B(I,JC,J2,J3),I=1,KNR)
      WRITE(6,8)
      WRITE(6,3)(Y(I),I=1,KNR)
      WRITE(6,9)
      WRITE(6,3)(YPI(I,JC,J2,J3),I=1,KNR)
    1 FORMAT(' WE ARE AT PROBLEM',I5,',',I5,',',I5)
    2 FORMAT(' X VALUES')
    3 FORMAT(8F10.2)
    4 FORMAT(' JH')
```

```
5       FORMAT(15I5)
6       FORMAT(' KINBAS')
7       FORMAT(' BS CURRENTLY')
8       FORMAT(' CURRENT Y')
9       FORMAT(' CURRENT PI')
10      FORMAT(//,' WE HAVE MADE',I5,' PASSES')
11       FORMAT(I4,' ITERATIONS SO FAR')
        RETURN
        END
C-------------------------------------------------------
        SUBROUTINE ACHECK
C
C LOOK AT THE VALUES IN A AT THE MOMENT
C
        IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
      1  INTEGER*4 (I-N,Q)
        INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
        INTEGER ICNAM(602,2),NAME(6)
        DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
        REAL A(3000,3,3,1)
C
        COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
      1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
      1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
      2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
      2  ),NETA,
      3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
      3  602,3,3,1),
      4  LE(1002),IA(3000,3,3,1),IE(3000),
      5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
        COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
        COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
      1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
      2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
        JC=JCUR
        J2=JPER(2)
        J3=JPER(3)
        WRITE(6,1) JC,J2,J3,NPASS
 1      FORMAT(//,' A MATRIX OF',I5,I5,I5,' AT PASS',I5)
        KNRO=NROW(JC,J2,J3)+1
        KNC=NCOL(JC,J2,J3)
        KNR=NROW(JC,J2,J3)
        DO 10 I=KNRO,KNC
        IND=I
        CALL UNPACK(IND)
        WRITE(6,2) IND
2       FORMAT(' COLUMN',I5)
        WRITE(6,3)(Y(J),J=1,KNR)
3       FORMAT(15F7.2)
10      CONTINUE
        RETURN
        END
C-------------------------------------------------------
        SUBROUTINE DLETCT
C
```

```
C DELETES CUTS THAT ARE SLACK.
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1 INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2 INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2 ),NETA,
     3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3 602,3,3,1),
     4 LE(1002),IA(3000,3,3,1),IE(3000),
     5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C CHECK FOR BASIC SLACKS
C
      WRITE(6,20)
20    FORMAT(//,' LOOKING FOR CUTS TO DELETE')
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      NRX=NROW(JC,J2,J3)
      IF(NRX.EQ.NROWP(JC,J2,J3)) GO TO 10
      KNRPO=NROWP(JC,J2,J3)+1
      DO 10 I=KNRPO,NRX
      IF (KINBAS(I,JC,J2,J3).NE.I) GO TO 10
C SLACK ON I-M IS BASIC
      IF(X(I,JC,J2,J3).LT.0.5) GO TO 10
C LARGE POSITIVE SLACK TO DELETE
      WRITE(6,21) I,X(I,JC,J2,J3)
21    FORMAT(//,' DELETING ROW',I6,' VALUE = ',E11.3)
C UP COLUMNS
      KNR=NROW(JC,J2,J3)
      DO 13  J=1,KNR
      IF(JH(J,JC,J2,J3).GT.I) JH(J,JC,J2,J3)=JH(J,JC,J2,J3)-1
 13   CONTINUE
      IF(I.EQ.NRX) GO TO 15
      KNRM=NROW(JC,J2,J3)-1
      DO 14 J=I,KNRM
C MOVE DOWN ROW INFO
      X(J,JC,J2,J3)=X(J+1,JC,J2,J3)
      B(J,JC,J2,J3)=B(J+1,JC,J2,J3)
      JH(J,JC,J2,J3)=JH(J+1,JC,J2,J3)
 14   CONTINUE
 15   CONTINUE
      KNCM=NCOL(JC,J2,J3)-1
```

```
            DO 3 J=1,KNCM
            IF(J.LT.I) GO TO 115
      C MOVE DOWN COL INFO
            XLB(J,JC,J2,J3)=XLB(J+1,JC,J2,J3)
            XUB(J,JC,J2,J3)=XUB(J+1,JC,J2,J3)
            KINBAS(J,JC,J2,J3)=KINBAS(J+1,JC,J2,J3)
       115    CONTINUE
            IF(KINBAS(J,JC,J2,J3).GT.I)
           1 KINBAS(J,JC,J2,J3)=KINBAS(J,JC,J2,J3)-1
        3   CONTINUE
      C  SHIFT UP LISTS
            KNC=NCOL(JC,J2,J3)
            LNEW=LA(I-1,JC,J2,J3)
            II=I+1
            DO 200 J=II,KNC
                 K1=LA(J,JC,J2,J3)
                 K2=LA(J+1,JC,J2,J3)-1
                       LA(J,JC,J2,J3)=LNEW+1
                 DO 30 K=K1,K2
                      IF(IA(K,JC,J2,J3).EQ.I)GO TO 30
                      LNEW=LNEW+1
                      A(LNEW,JC,J2,J3)=A(K,JC,J2,J3)
                      IA(LNEW,JC,J2,J3)=IA(K,JC,J2,J3)
                      IF(IA(K,JC,J2,J3).GT.I)IA(LNEW,JC,J2,J3)=IA(LNEW,JC,J2,J3)-1
        30       CONTINUE
        200   CONTINUE
            LA(KNC+1,JC,J2,J3)=LNEW+1
      C SHIFT UP LA
            DO 35 J=I,KNC
            LA(J,JC,J2,J3)=LA(J+1,JC,J2,J3)
        35  CONTINUE
            NELM(JC,J2,J3)=LNEW
            NCOL(JC,J2,J3)=NCOL(JC,J2,J3)-1
            NROW(JC,J2,J3)=NROW(JC,J2,J3)-1
      C     CALL STRPRT
      C     CALL ACHECK
        10  CONTINUE
       11   CONTINUE
            RETURN
            END
      C----------------------------------------------------------------
            SUBROUTINE FEASCT
      C
      C  FINDS AND ADDS A FEASIBILITY CUT TO NODE I
      C
            IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
           1  INTEGER*4 (I-N,Q)
            INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
            INTEGER ICNAM(602,2),NAME(6)
            DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
            REAL A(3000,3,3,1)
      C
            COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
           1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
           1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
           2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
           2  ),NETA,
           3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
```

```
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C FIRST, FORM THE ROW
C
      WRITE(6,200)
200   FORMAT(' MAKING A FEASIBILITY CUT')
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      JCT=JC
      J2T=J2
      J2L=J2
      J3T=J3
      IF(JC .EQ. 2) J2=1
      IF(JC .EQ. 3) J3=1
      KNCO=NCOL(JC-1,J2,J3)+1
      DO 10 J=1,KNCO
      Y(J)=0.
      YTEMP(J)=0.
 10   CONTINUE
C
C COLUMN BY COLUMN IN TEMP STOR
      IF(JCUR.EQ.NPER) J2L=1
C
      IF(JCT.EQ.NPER) NROWP(JCT,J2L,J3T)=NROWP(JCT,1,1)
      JND=NROW(JC-1,J2,J3)+1
      IF(NROWP(JC-1,J2,J3).GE.NROWP(JCT,J2L,J3T)) NRO=NROWP(JCT,J2L,J3T)
      IF(NROWP(JC-1,J2,J3).LT.NROWP(JCT,J2L,J3T)) NRO=NROWP(JC-1,J2,J3)
      JNX = NROW(JC-1,J2,J3) - NRO
      JCUR=JC-1
      JPER(2)=J2
      JPER(3)=J3
      KNC=NCOL(JC-1,J2,J3)
      DO 30 J=JND,KNC
      JNL=J-JNX
      CALL BUNPCK(JNL)
      KNR=NROW(JCT,J2L,J3T)
         DO 20 K=1,KNR
         YTEMP(J)=YTEMP(J)+YPI(K,JCT,J2L,J3T)*Y(K)
20       CONTINUE
30    CONTINUE
C
C FIND RHS TOO
C
      KNR=NROW(JCT,J2L,J3T)
      DO 40 K=1,KNR
      YTEMP(NCOL(JC-1,J2,J3)+1)=YTEMP(NCOL(JC-1,J2,J3)+1)+
     1 YPI(K,JCT,J2L,J3T)*XKSI(K,JCT,J2L,J3T)
 40   CONTINUE
```

```
C CHANGE RHS FOR UPPER BOUNDED VARIABLES
      KNC=NCOL(JCT,J2L,J3T)
       DO 41 K=1,KNC
      IF(KINBAS(K,JCT,J2L,J3T).NE.-1) GO TO 41
      JCUR=JCUR+1
      CALL UNPACK(K)
      JCUR=JCUR-1
      CBAR=0
      DO 42 J=1,KNR
      CBAR=CBAR + Y(J)*YPI(J,JCT,J2L,J3T)
 42   CONTINUE
      YTEMP(NCOL(JC-1,J2,J3)+1)=YTEMP(NCOL(JC-1,J2,J3)+1)-CBAR*
     1 XUB(K,JCT,J2L,J3T)
 41   CONTINUE
C
C SHIFT A(*,JC-1,J2,J3) DOWN
C     FIRST, COPY INTO TEMPS
C
      KLM=NELM(JC-1,J2,J3)
      DO 50 I=1,KLM
      ATMP(I)=A(I,JC-1,J2,J3)
      ITMP(I)=IA(I,JC-1,J2,J3)
 50   CONTINUE
C
C ADD COL AND ROW FOR SLACK
C
      NCOL(JC-1,J2,J3)=NCOL(JC-1,J2,J3)+1
      NROW(JC-1,J2,J3)=NROW(JC-1,J2,J3)+1
      NELM(JC-1,J2,J3)=NELM(JC-1,J2,J3)+NCOL(JC-1,J2,J3)-
     1 NROW(JC-1,J2,J3)+1
      JH(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
C
C COPY COL VALUES
C
      KNCM=NCOL(JC-1,J2,J3)-1
      DO 60 I=1,KNCM
      KBTMP(I)=KINBAS(I,JC-1,J2,J3)
      LTMP(I)=LA(I,JC-1,J2,J3)
      XLTMP(I)=XLB(I,JC-1,J2,J3)
      XUTMP(I)=XUB(I,JC-1,J2,J3)
 60   CONTINUE
C
C UPDATE COLUMNS
C
      KINBAS(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
      LA(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
      XLB(NROW(JC-1,J2,J3),JC-1,J2,J3)=0.
      XUB(NROW(JC-1,J2,J3),JC-1,J2,J3)=1.0E+7
      JND=0
C
      KNRO=NROW(JC-1,J2,J3)+1
      KNC=NCOL(JC-1,J2,J3)
      DO 70 I=KNRO,KNC
      JND=JND+1
      LA(I,JC-1,J2,J3)=LTMP(I-1)+JND
      KINBAS(I,JC-1,J2,J3)=KBTMP(I-1)
      XLB(I,JC-1,J2,J3)=XLTMP(I-1)
      XUB(I,JC-1,J2,J3)=XUTMP(I-1)
```

```
 70   CONTINUE
      LA(NCOL(JC-1,J2,J3)+1,JC-1,J2,J3)=NELM(JC-1,J2,J3)+1
C
      A(NROW(JC-1,J2,J3),JC-1,J2,J3)=1.
      IA(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
C
C UPDATE A AND IA
C
      DO 80 I=KNRO,KNC
      IND=I
      KND=LA(IND,JC-1,J2,J3)-1
      IF(KND.EQ.NROW(JC-1,J2,J3)) GO TO 81
      IF(DABS(YTEMP(IND-2)) .GE. ZTOLZE) GO TO 75
C
C             SHIFT ALL DOWN
C
      NELM(JC,J2,J3)=NELM(JC,J2,J3)-1
      KKNC=KNC+1
      DO 72 II=IND,KKNC
      LA(II,JC,J2,J3)=LA(II,JC,J2,J3)-1
 72   CONTINUE
      KLM=NELM(JC,J2,J3)+1
      DO 73 II=KND,KLM
      A(II,JC,J2,J3)=A(II+1,JC,J2,J3)
      IA(II,JC,J2,J3)=IA(II+1,JC,J2,J3)
 73   CONTINUE
      GO TO 81
 75   CONTINUE
      A(KND,JC-1,J2,J3)=YTEMP(IND-2)
      IA(KND,JC-1,J2,J3)=NROW(JC-1,J2,J3)
 81   CONTINUE
      MND=LA(IND+1,JC-1,J2,J3)-2
      IF(IND.EQ.NCOL(JC-1,J2,J3)) MND=NELM(JC-1,J2,J3)-1
      KKND=KND+1
      DO 90 J=KKND,MND
      JNX=J-IND+NROW(JC-1,J2,J3)
      A(J,JC-1,J2,J3)=ATMP(JNX)
      IA(J,JC-1,J2,J3)=ITMP(JNX)
 90   CONTINUE
 80   CONTINUE
C
C SPECIAL FOR NO LOOK CUTS
      IF(NTH(JC-1,J2,J3).EQ.1) GO TO 82
      A(NELM(JC-1,J2,J3),JC-1,J2,J3)=YTEMP(NCOL(JC-1,J2,J3)+1)
      IA(NELM(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
      GO TO 83
 82   CONTINUE
      A(NELM(JC-1,J2,J3),JC-1,J2,J3)=0.
      IA(NELM(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
 83    CONTINUE
C UPDATE THE JH TOO
      KNRM=NROW(JC-1,J2,J3)-1
      DO 85 I=1,KNRM
      IF (JH(I,JC-1,J2,J3).GE.NROW(JC-1,J2,J3))
     1 JH(I,JC-1,J2,J3)=JH(I,JC-1,J2,J3)+1
 85   CONTINUE
C
C NEW RHS
```

```
C
      XKSI(NROW(JC-1,J2,J3),JC-1,J2,J3)=YTEMP(NCOL(JC-1,J2,J3))
      B(NROW(JC-1,J2,J3),JC-1,J2,J3)=YTEMP(NCOL(JC-1,J2,J3))
      JCUR=JCT
      JPER(2)=J2T
      JPER(3)=J3T
      DO 3001 I=1,3000
      IF(IA(I,1,1,1).GT.300) STOP
3001  CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE LKHDCT(NODE)
C
C ADDS A LOOK-AHEAD CUT TO THE PREVIOUS NODE.
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C INITIALIZE
C
      WRITE(6,200)
200   FORMAT(' MAKING A LOOKAHEAD CUT')
      JC=NODE
      JCT=JCUR
      J2=JPER(2)
      J2T=JPER(2)
      J2L=JPER(2)
      IF(JC.EQ.NPER-1) J2L=1
      J3=1
      J3T=JPER(3)
      IF(NODE .EQ. 2) J2=1
      KNC=NCOL(JC,J2,J3)+1
      DO 10 J=1,300
      Y(J)=0.
      YTEMP(J)=0.
      YTEMP1(J)=0.
```

```
   10 CONTINUE
C
C COL BY COL INTO YTEMP
C
      IF(JC+1.GE.NPER) NROWP(JC+1,J2L,J3) = NROWP(JC+1,1,1)
      JC=JC-1
      JND=NROW(JC,J2,J3)+1
      IF(NROWP(JC,J2,J3).GE.NROWP(JC+1,J2L,J3)) NRO = NROWP(JC+1,J2L,J3)
      IF(NROWP(JC,J2,J3).LT.NROWP(JC+1,J2L,J3)) NRO = NROWP(JC,J2,J3)
      JNX = NROW(JC,J2,J3) - NRO
      KNC=NCOL(JC,J2,J3)
      IF(NTH(JC,J2,J3).EQ.1) KNC=KNC-1
      DO 30 J=JND,KNC
      JNL=J-JNX
      JCUR=JC
      JPER(2)=J2
      JPER(3)=J3
      CALL BUNPCK(JNL)
      IF(NODE .LT. NPER) GO TO 19
      KNR=NROW(NPER,1,1)
      DO 18 L=2,KNR
      YTEMP(J)=YTEMP(J)+Y(L)*YPIBAR(L)
   18 CONTINUE
      GO TO 30
   19 CONTINUE
      NNDD=NND(JC+1)
      DO 25 L=1,NNDD
        YTEMP1(J)=0.
      IF(JC+1 .EQ. 2) KNR=NROW(JC+1,L,J3)
      IF(JC+1 .EQ. 3) KNR=NROW(JC+1,J2L,L)
              DO 20 K=1,KNR
                IF(JC+1 .EQ. 2) YTEMP1(J)=YTEMP1(J)+YPI(K,JC+1,L,J3)*Y(K)
                IF(JC+1 .EQ. 3) YTEMP1(J)=YTEMP1(J)+YPI(K,JC+1,J2L,L)*Y(K)
 20           CONTINUE
          IF(JC+1 .EQ. 2) YTEMP(J)=YTEMP(J)+PROB(JC+1,L,J3)*YTEMP1(J)
          IF(JC+1 .EQ. 3) YTEMP(J)=YTEMP(J)+PROB(JC+1,J2L,L)*YTEMP1(J)
 25       CONTINUE
 30   CONTINUE
C
C RHS GOES IN TOO
C
      IF(NODE .LT. NPER) GO TO 31
      YTEMP(NCOL(JC,J2,J3)+1)=XRHO
      GO TO 39
   31 CONTINUE
      DO 35 L=1,NNDD
      YTEMP1(NCOL(JC,J2,J3)+1)=0.
          IF(JC+1 .EQ. 2) KNR=NROW(JC+1,L,J3)
          IF(JC+1 .EQ. 3) KNR=NROW(JC+1,J2L,L)
          DO 40 K=1,KNR
          IF(JC+1.EQ. 2) YTEMP1(NCOL(JC,J2,J3)+1)=
     1     YTEMP1(NCOL(JC,J2,J3)+1)+YPI(K,JC+1,L,J3)*XKSI(K,JC+1,L,J3)
          IF(JC+1.EQ. 3) YTEMP1(NCOL(JC,J2,J3)+1)=
     1     YTEMP1(NCOL(JC,J2,J3)+1)+YPI(K,JC+1,J2L,L)*XKSI(K,JC+1,J2L,L)

 40       CONTINUE
      IF(JC+1 .EQ. 2) YTEMP(NCOL(JC,J2,J3)+1)=
     1 YTEMP(NCOL(JC,J2,J3)+1)+PROB(JC+1,L,J3)*YTEMP1(NCOL(JC,J2,J3)+1)
```

```
          IF(JC+1 .EQ. 3) YTEMP(NCOL(JC,J2,J3)+1)=
         1 YTEMP(NCOL(JC,J2,J3)+1)+PROB(JC+1,J2L,L)*YTEMP1(NCOL(JC,J2,J3)+1)
   35     CONTINUE
C
C SHIFT A(*,1) DOWN
C    FIRST, COPY INTO TEMPS
C
   39 CONTINUE
      KLM=NELM(JC,J2,J3)
       DO 50 I=1,KLM
       ATMP(I)=A(I,JC,J2,J3)
       ITMP(I)=IA(I,JC,J2,J3)
  50   CONTINUE
C
C ADD COL AND ROW FOR SLACK
C
       NCOL(JC,J2,J3)=NCOL(JC,J2,J3)+1
       NROW(JC,J2,J3)=NROW(JC,J2,J3)+1
       NELM(JC,J2,J3)=NELM(JC,J2,J3)+NCOL(JC,J2,J3)-NROW(JC,J2,J3)+1
       JH(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
C
C COPY COL VALUES
C
      KNCM=NCOL(JC,J2,J3)-1
       DO 60 I=1,KNCM
       KBTMP(I)=KINBAS(I,JC,J2,J3)
       LTMP(I)=LA(I,JC,J2,J3)
       XLTMP(I)=XLB(I,JC,J2,J3)
       XUTMP(I)=XUB(I,JC,J2,J3)
  60   CONTINUE
C
C UPDATE COLUMNS
C
       KINBAS(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
       LA(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
       XLB(NROW(JC,J2,J3),JC,J2,J3)=0.
       XUB(NROW(JC,J2,J3),JC,J2,J3)=1.E7
       JND=0
C
      KNRO=NROW(JC,J2,J3)+1
      KNC=NCOL(JC,J2,J3)
      DO 70 I=KNRO,KNC
        JND=JND+1
        LA(I,JC,J2,J3)=LTMP(I-1)+JND
       KINBAS(I,JC,J2,J3)=KBTMP(I-1)
       XLB(I,JC,J2,J3)=XLTMP(I-1)
       XUB(I,JC,J2,J3)=XUTMP(I-1)
  70   CONTINUE
      LA(NCOL(JC,J2,J3)+1,JC,J2,J3)=NELM(JC,J2,J3)+1
C
       A(NROW(JC,J2,J3),JC,J2,J3)=1.
       IA(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
C
C UPDATE A AND IA
C
      DO 80 I=KNRO,KNC
        IND=I
       KND=LA(IND,JC,J2,J3)-1
```

```
         IF(KND.EQ.NROW(JC,J2,J3)) GO TO 81
        IF(DABS(YTEMP(IND-2)) .GE. ZTOLZE) GO TO 75
C
C              SHIFT ALL DOWN
C
        NELM(JC,J2,J3)=NELM(JC,J2,J3)-1
        KKNC=KNC+1
        DO 72 II=IND,KKNC
        LA(II,JC,J2,J3)=LA(II,JC,J2,J3)-1
72      CONTINUE
        KLM=NELM(JC,J2,J3)+1
        DO 73 II=KND,KLM
        A(II,JC,J2,J3)=A(II+1,JC,J2,J3)
        IA(II,JC,J2,J3)=IA(II+1,JC,J2,J3)
73      CONTINUE
        GO TO 81
75      CONTINUE
        A(KND,JC,J2,J3)=YTEMP(IND-2)
         IA(KND,JC,J2,J3)=NROW(JC,J2,J3)
 81      CONTINUE
         MND=LA(IND+1,JC,J2,J3)-2
         IF(IND.EQ.NCOL(JC,J2,J3)) MND=NELM(JC,J2,J3)-1
        KKND=KND+1
         DO 90 J=KKND,MND
         JNX=J-IND+NROW(JC,J2,J3)
         A(J,JC,J2,J3)=ATMP(JNX)
         IA(J,JC,J2,J3)=ITMP(JNX)
  90     CONTINUE
  80     CONTINUE
C
C CHECK FOR FIRST CUT
         IF (NTH(JC,J2,J3).EQ.1) GO TO 811
         A(NELM(JC,J2,J3),JC,J2,J3)=YTEMP(NCOL(JC,J2,J3)-1)
         IA(NELM(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
 811      CONTINUE
C
C NEW RHS
C
        XKSI(NROW(JC,J2,J3),JC,J2,J3)=YTEMP(NCOL(JC,J2,J3))
         B(NROW(JC,J2,J3),JC,J2,J3)=YTEMP(NCOL(JC,J2,J3))
C CHECK IF THIS IS THE FIRST TIME
        IF(NTH(JC,J2,J3).EQ.1) GO TO 100
        NTH(JC,J2,J3)=1
        NCOL(JC,J2,J3)=NCOL(JC,J2,J3)+1
        NELM(JC,J2,J3)=NELM(JC,J2,J3)+1
        A(NELM(JC,J2,J3),JC,J2,J3)=-1
        IA(NELM(JC,J2,J3),JC,J2,J3)=1
        NELM(JC,J2,J3)=NELM(JC,J2,J3)+1
C ADD THETA
        LA(NCOL(JC,J2,J3)+1,JC,J2,J3)=NELM(JC,J2,J3)+1
        XUB(NCOL(JC,J2,J3),JC,J2,J3)=1.0E7
        XLB(NCOL(JC,J2,J3),JC,J2,J3)=-1.0E7
C MAKE THETA BASIC NOT THE SLACK
        KINBAS(NROW(JC,J2,J3),JC,J2,J3)=0
        KINBAS(NCOL(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
        JH(NROW(JC,J2,J3),JC,J2,J3)=NCOL(JC,J2,J3)
C UPDATE THE JH TOO
100     KNRM=NROW(JC,J2,J3)-1
```

```
      DO 85 I=1,KNRM
      IF (JH(I,JC,J2,J3).GE.NROW(JC,J2,J3))
     1  JH(I,JC,J2,J3)=JH(I,JC,J2,J3)+1
 85   CONTINUE
C NOW, A "1" FOR THETA.
C
        A(NELM(JC,J2,J3),JC,J2,J3)=1.
       IA(NELM(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
       JCUR=JCT
       JPER(2)=J2T
       JPER(3)=J3T
       RETURN
       END
C-----------------------------------------------------------
      SUBROUTINE FRMRHS
C
C TAKES B.X AND B TO MAKE A NEW RHS
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      JC=JCUR+1
      J2=JPER(2)
      J3=JPER(3)
      IF(JCUR.EQ.NPER-1) J2=1
      IF(JCUR.EQ.NPER-1) J3=1
      NNDD=NND(JC)
      DO 200 J=1,NNDD
      KNR=NROW(JC,J2,J3)
      DO 100 I=2,KNR
      IF(JC .EQ. 3) B(I,JC,J2,J)=XKSI(I,JC,J2,J)-Y(I)
      IF(JC .EQ. 2) B(I,JC,J,1)=XKSI(I,JC,J,1)-Y(I)
 100  CONTINUE
 200  CONTINUE
      RETURN
      END
C------------------------------------------------------------------
      SUBROUTINE BPRODX
```

```
C
C FINDS TH VALUE OF BX FOR THE NEXT PERIOD INVENTORIES
C AND PUTS IT INTO Y
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C SET Y'S
C
      JCT=JCUR
      J2=JPER(2)
      J3=JPER(3)
      JC=JCUR+1
      IF(JC.EQ.NPER)NROWP(JC,J2,J3)=NROWP(JC,1,1)
      KNRO=NROW(JC,J2,J3)
      DO 10 I=1,KNRO
      YTEMP(I)=O.
      Y(I)=O.
   10 CONTINUE
C
C MULTIPLY BY THE BASIC COLS' VALUES
      IF(NROWP(JCT,J2,J3).GE.NROWP(JC,J2,J3)) NRO = NROWP(JC,J2,J3)
      IF(NROWP(JCT,J2,J3).LT.NROWP(JC,J2,J3))  NRO = NROWP(JCT,J2,J3)
      KNRO=NROW(JCT,J2,J3)+1
      KNC=NCOL(JCT,J2,J3)
      DO 20 J=KNRO,KNC
      JND=J-NROW(JCT,J2,J3)+NRO
      IF(JND-NRO.GT.NCOLP(JCT,J2,J3)-NROWP(JCT,J2,J3)) GO TO 20
   13 CONTINUE
      CALL BUNPCK(JND)
      IF(KINBAS(J,JCT,J2,J3).EQ.O) XIN=XLB(J,JCT,J2,J3)
      IF(KINBAS(J,JCT,J2,J3).EQ.O) GO TO 114
      IF(KINBAS(J,JCT,J2,J3).EQ.-1) XIN=XUB(J,JCT,J2,J3)
      IF(KINBAS(J,JCT,J2,J3).EQ.-1) GO TO 114
      XIN=X(KINBAS(J,JCT,J2,J3),JCT,J2,J3)
  114 CONTINUE
      KNRD=NROW(JC,J2,J3)
```

```
          DO 15 I=1,KNRD
   14     Y(I)=Y(I)*XIN
          YTEMP(I)=Y(I)+YTEMP(I)
   15     CONTINUE
   20     CONTINUE
C
          DO 25 I=1,KNRD
          Y(I)=YTEMP(I)
   25     CONTINUE
          RETURN
          END
C-----------------------------------------------------------------
          SUBROUTINE OPTCHK(NODE)
C
C CHECKS FOR MASTER-SUB OPTIMALITY
C
          IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
         1  INTEGER*4 (I-N,Q)
          INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
          INTEGER ICNAM(602,2),NAME(6)
          DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
          REAL A(3000,3,3,1)
C
          COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
         1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
         1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
         2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
         2  ),NETA,
         3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
         3  602,3,3,1),
         4  LE(1002),IA(3000,3,3,1),IE(3000),
         5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

         6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
         7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
          COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
          COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
         1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),ICUR(3),
         2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
C FIRST, GET THE EXPECTED VALUE
C
            XT=0.
            J2=JPER(2)
            NNDD=NND(NODE)
            DO 10 J=1,NNDD
            IF(NODE .EQ. 2) XT=XT+PROB(2,J,1)*X(1,2,J,1)
            IF(NODE .EQ. 3) XT=XT+PROB(3,J2,J)*X(1,3,J2,J)
   10       CONTINUE
C THEN, CHECK
            IF(NODE .EQ. 2) JN=KINBAS(NCOL(1,1,1),1,1,1)
            IF(NODE .EQ. 3) JN=KINBAS(NCOL(2,J2,1),2,J2,1)
            IF(JN.LE.0) RETURN
            IF((NODE .EQ. 2) .AND. (XT .GE. X(JN,1,1,1))) NFLG=1
            IF((NODE .EQ. 3) .AND. (XT .GE. X(JN,2,J2,1))) NFLG=1
            RETURN
            END
C-----------------------------------------------------------------
```

```
      SUBROUTINE WRAPUP
C
C   OUTPUT OPTIMAL SOLUTION
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2,3,3,1)
      DOUBLE PRECISION E(3000)
      REAL A(3000,3,3,1)
      DIMENSION XTEMP(602)
      EQUIVALENCE (XTEMP(1),Y(1))
C
      COMMON/BL5/ DRES,ICNAM
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      IF (ITCNT .LT. ITRFRQ) GO TO 20
      WRITE (6,1) ITCNT
1     FORMAT (' SIMPLEX ITERATIONS =',I8,':  COMPUTATIONS TERMINATED.')
20    IF (MSTAT .EQ. QN) GO TO 1000
      JPER(2)=1
      JPER(3)=1
      JCUR=NPER
      IN1=INST(1)
      IN2=INST(2)
      IN3=INST(3)
      DO 400 I=1,IN1
      DO 400 J=1,IN2
      DO 400 K=1,IN3
      WRITE(6,401) I,J,K
401   FORMAT(//,' LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO',3I6)
      WRITE(6,403)
403      FORMAT(/,' ROW',6X,' VALUE')
      IBASE=JSTCH(I,J,K)
      NETA=NETND(IBASE)
      XKSI(IBST(1),NPER,1,1)=CBST(1,I)
      XKSI(IBST(2),NPER,1,1)=CBST(2,J)
      XKSI(IBST(3),NPER,1,1)=CBST(3,K)
      JCUR=NPER-1
      JPER(NPER-1)=1
      CALL BPRODX
      CALL FRMRHS
      JCUR = NPER
```

```
      KNR = NROW(NPER,1,1)
      DO 420 IL=1,KNR
      Y(IL)=B(IL,NPER,1,1)
420    CONTINUE
      CALL FTRAN(1)
      DO 402 JK=1,KNR
      WRITE(6,412) ICNAM(JK,1,NPER,1,1),ICNAM(JK,2,NPER,1,1),Y(JK)
412    FORMAT(2A4,2X,F13.4)
402   CONTINUE
400   CONTINUE
      DO 310 J=1,NPER
      NNDD=NND(2)
      IF(J.EQ.NPER)NNDD=1
      DO 300 K=1,NNDD
      NNDDD=NND(3)
      IF(J.EQ.NPER)NNDDD=1
      DO 295 L=1,NNDDD
       NODE=K
      JCUR=J
      JPER(2)=K
      JPER(3)=L
      IF((J.EQ.1).AND.((K.NE.1).OR.(L.NE.1))) GO TO 295
      IF(J.EQ.2.AND.L.NE.1) GO TO 295
      CALL INVERT
      WRITE(7,117)
117   FORMAT(' BASIS')
      WRITE (6,17) J,K,L
 17   FORMAT(//,' PROBLEM NODE = ',I4,'-',I4,'-',I3,//)
      WRITE (6,2) X(IOBJ,J,K,L)
2     FORMAT(' OPTIMUM LP OBJECTIVE VALUE =',F10.2)
      WRITE(6,11) NPASS
11    FORMAT(' AFTER',I5,' PASSES FROM NODE TO NODE')
      WRITE(6,111) ITCNT
111   FORMAT(' AND AFTER',I5,' ITERATIONS')
      WRITE (6,3)
3     FORMAT(' OPTIMAL VALUE REACHED AT THE POINT')
      KNR=NROW(J,K,L)
      DO 100 I=1,KNR
100      XTEMP(JH(I,J,K,L)) = X(I,J,K,L)
      KNC=NCOL(J,K,L)
      WRITE(6,101)
101   FORMAT(/,' VARIABLE',10X,'STATUS',5X,'VALUE')
      DO 200 JJ=1,KNC
      JL=JJ
      NRDF=NROW(J,K,L)-NROWP(J,K,L)
      IF(JJ.GT.NROWP(J,K,L)) JL=JJ-NRDF
      IF(JL.GT.NCOLP(J,K,L)) GO TO 200
      IF((JL.LT.JJ).AND.(JL.LE.NROWP(J,K,L))) GO TO 200
         IF (KINBAS(JJ,J,K,L).GT.0) GO TO 151
         IF (KINBAS(JJ,J,K,L).EQ.0) GO TO 150
         XTEMP(JJ) = XUB(JJ,J,K,L)
         GO TO 151
150      XTEMP(JJ) = XLB(JJ,J,K,L)
151     CONTINUE
      KB=KINBAS(JJ,J,K,L)
      WRITE(6,102) ICNAM(JL,1,J,K,L),ICNAM(JL,2,J,K,L),KB,XTEMP(JJ)
102   FORMAT(2A4,12X,I4,5X,F12.4)
      IF(KB.LE.0) GO TO 200
```

```
      IF(KB.GT.NROWP(J,K,L)) GO TO 200
      WRITE(7,118) ICNAM(JL,1,J,K,L),ICNAM(JL,2,J,K,L),ICNAM(KB,1,J,K,L)
     1  ,ICNAM(KB,2,J,K,L)
118   FORMAT(4X,2A4,2X,2A4)
200      CONTINUE
      CALL FORMC
      CALL BTRAN
      WRITE(6,6)
6     FORMAT(' OPTIMAL DUAL VALUES')
      WRITE(6,4) (Y(JJ),JJ=1,KNR)
  4     FORMAT(8F9.2)
295   CONTINUE
300   CONTINUE
310   CONTINUE
      RETURN
1000  WRITE (6,5)
5     FORMAT (' NO FEASIBLE SOLUTION FOUND.')
      JCUR=1
      JPER(2)=1
      JPER(3)=1
      CALL STRPRT
      RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE INIT
C
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      DO 40 I=1,3000
      DO 30 J=1,3
      DO 20 K=1,2
      DO 10 L=1,2
      A(I,J,K,L)=0.0
```

```
      IA(I,J,K,L)=0
      IF(I .GT. 500) GO TO 10
      ABN(I,J,K,L)=0.0
      IBN(I,J,K,L)=0.0
      IF(I .GT. 522) GO TO 10
      XLB(I,J,K,L)=0.0
      XUB(I,J,K,L)=0.0
      KINBAS(I,J,K,L)=0
      LA(I,J,K,L)=0
      LBN(I,J,K,L)=0
      IF(I .GT. 300) GO TO 10
      XKSI(I,J,K,L)=0.0
      YPI(I,J,K,L)=0.0
      JH(I,J,K,L)=0
      X(I,J,K,L)=0.0
      B(I,J,K,L)=0.0
10    CONTINUE
20    CONTINUE
30    CONTINUE
40    CONTINUE
      DO 50 I=1,600
      ATMP(I)=0.0
      ITMP(I)=0
      IF(I .GT. 122 ) GO TO 50
      YTEMP(I)=0.0
      YTEMP1(I)=0.0
      KBTMP(I)=0
      XUTMP(I)=0.0
      XLTMP(I)=0.0
      LTMP(I)=0
      IF (I .GT. 60) GO TO 50
      Y(350)=0.0
50    CONTINUE
      DO 80 I=1,3
      DO 70 J=1,2
      DO 60 L=1,2
      NROW(I,J,L)=0
      NCOL(I,J,L)=0
      NELM(I,J,L)=0
      PROB(I,J,L)=0.0
      NROWP(I,J,L)=0
      NCOLP(I,J,L)=0
      NTH(I,J,L)=0
60    CONTINUE
70    CONTINUE
80    CONTINUE
      DO 11 I=1,3000
      IE(I)=0
      E(I)=0.0
11    CONTINUE
      DO 12 I=1,1002
      LE(I)=0
12    CONTINUE
      DO 17 I=1,5
      NND(I)=0
      JPER(I)=0
17    CONTINUE
      DE=0.0
```

```
      DP=0.0
      MSTAT=0
      IOBJ=0
      IROWP=0
      ITCNT=0
      INVFRQ=0
      ITRFRQ=0
      JCOLP=0
      NETA=0
      NLELEM=0
      NLETA=0
      NUELEM=0
      NUETA=0
      NELEM=0
      NR=0
      NPASS=0
      JCUR=0
      JPASS=0
      NPER=0
      NFLG=0
      ATEMP1=0.0
      ATEMP2=0.0
      NODE=0
      DO 13 I=1,6
      NAME(I)=0
13    CONTINUE
      DO 14 I=1,122
      DO 15 J=1,2
      ICNAM(I,J)=0
15    CONTINUE
14    CONTINUE
      RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE XOPTCK
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
```

```
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      J2=JPER(2)
      J3=JPER(3)
      JN=KINBAS(NCOL(NPER-1,J2,J3),NPER-1,J2,J3)
      IF(XTOPT.LT.(X(JN,NPER-1,J2,J3)*(1/1.01))) NFLG=0
C CHANGE IF NEGATIVE
       IF(XTOPT.GT.0) GO TO 12
      NFLG = 1
       IF(XTOPT.LT.(X(JN,NPER-1,J2,J3)*(1/.99))) NFLG = 0
 12    CONTINUE
      XQX=-XTOPT
      WRITE(6,10) XQX
10    FORMAT(' Q(X) =',E11.3)
      XTH=-X(JN,NPER-1,J2,J3)
      WRITE(6,11) XTH
11    FORMAT(' THETA=',E11.3)
C CHECK IF THIS IS THE FIRST TIME
      IF(NTH(NPER-1,J2,J3).EQ.0) NFLG=0
      IF(NTH(NPER-1,J2,J3).EQ.0) RETURN
C
C  FIND BOUNDS
C
      IF(NPER.NE.2) RETURN
      X00=-X(1,1,1,1)
      WRITE(6,1) X00
1     FORMAT(' LOWER BOUND=',E11.3)
      XC=-X(1,1,1,1)+X(JN,1,1,1)-XTOPT
      WRITE(6,2) XC
2     FORMAT(/,' UPPER BOUND=',E11.3)
      RETURN
      END
C-------------------------------------------------------------------
      SUBROUTINE DNORML
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
      COMMON/BL5/DRES,ICNAM
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
```

```
     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      COMMON/BLS/ ITSIN
C
      JC=JCUR
      J2=JPER(2)
      J3=JPER(3)
      CMIN=0.
      NINF=0
      NOPT=0
      MSTAT=QF
      WRITE(6,1001)
 1001 FORMAT(' ITNS',2X,'RRES',5X,' PIV',6X,' IN',3X,'OUT',2X,
     1 'OBJ',7X,'CMIN',7X,'MXINF',7X,'THET',3X,'NINF',1X,'NOPT')
      IF(ITSIN .LT. INVFRQ) GO TO 1500
 1000  CONTINUE
C      CALL INVERT
       WRITE(6,1003) ITSIN
1003   FORMAT(' WARNING -- TOO MANY ETAS IN DNORML',1X,' ITERS =',I6)
C     ITSIN=0
C
C             DUAL SIMPLEX CYCLE
C
 1500 CALL DCHUZR
C
C               IROWP=0 IF OPTIMAL
C
      IF(IROWP .EQ. 0) GO TO 6000
C
C               FIND PIVOT COL. - JCOLP=0 IF INFEASIBLE
C
      CALL DCHUZC
      IF(JCOLP .EQ. 0) MSTAT=QN
      IF(JCOLP .EQ. 0) GO TO 6000
      CALL UPBETA
      ITCNT=ITCNT+1
      ITSIN=ITSIN+1
      IF(NPIVOT .EQ. 0) GO TO 4010
      IF(NELEM .GT.(NEMAX-NROW(JCUR,JPER(2),JPER(3)))) GO TO 1000
      CMAX=YTEMP1(1)
      APV=YTEMP1(2)
      CALL WRETA
      CALL RHCHCK
      WRITE(6,1002)ITCNT,DRES,APV,JCOLP,IROWP,X(1,JC,J2,J3),CMIN,CMAX
     1 ,DP,NINF,NOPT
 1002 FORMAT(I4,1X,E9.2,1X,E9.2,1X,I4,1X,I4,1X,E10.2,1X,E9.2,1X,E9.2,1X,
     1 E9.2,1X,I4,1X,I4)
 4010 IF(ITSIN .GE. INVFRQ) GO TO 1000
      IF(ITCNT .GE. ITRFRQ) GO TO 6000
      GO TO 1500
 6000 RETURN
      END
C-----------------------------------------------------------------
      SUBROUTINE PARSFT
```

```
C
C        THIS ROUTINE FINDS ALL THE BASES THAT WILL MAKE
C        THE LAST PERIOD OPTIMAL.
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y);
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
      COMMON/BLS/ITSIN
C
C        SET JPASS
C
      JPASS=2
C
C        SET ALL THE PROPER VALUES
C
C       'IBASE' IS THE CURRENT BASIS
C       'ITSIN' NUMBER OF ITERATIONS FOR DN
C
      IBASE=1
      ITSIN=0
C
C SET JPER FOR CHECKING
      JPOLD=JPER(NPER-1)
      JPER(NPER-1)=1
C        'XRHO' IS THE CURRENT EX VAL OF THE RHS
C        'XTOPT' IS EX VAL OF Z
C
      XRHO=0
      XTOPT=0
C
C        'YPIBAR' KEEPS THE CURRENT EX VAL OF THE PI VECTOR
C
      KNR=NROW(NPER,1,1)
      DO 10 I=1,KNR
      YPIBAR(I)=0
   10 CONTINUE
```

```
C
C        'JSTCH(I,J,K,L)' HAS O IF ALTERNATIVE (I,J,K,L)
C           HAS NOT YET BEEN ASSIGNED A BASIS.
C
      DO 11 I=1,MXNST
      DO 11 J=1,MXNST
      DO 11 K=1,MXNST
      JSTCH(I,J,K)=O
   11 CONTINUE
C
C        'NCUR' KEEPS THE CURRENT ALTERNATIVE
C        'NXNF' KEEPS THE INDEX OF THE FIRST INFEASIBLE ALTERNATIVE.
C
      DO 12 I=1,3
      NCUR(I)=1
      NXNF(I)=1
   12 CONTINUE
C
C        'NETND(I)' KEEPS THE INDEX OF THE ETA VECTOR FOR
C             BASIS 'I'.
C
      DO 13 I=1,3
      NETND(I)=NETA
   13 CONTINUE
C
C        'INFLG' EQUAL O SAYS THERE ARE NO INFEASIBILITIES CAUGHT YET.
      INFLG=O
C
C         'BND(I)' HAS THE BASELINE VALUES FOR THE RHS'S IN NPER.
C
      KNR=NROW(NPER,1,1)
      DO 14 I=1,KNR
      XKSI(I,NPER,1,1)=BND(I)
   14 CONTINUE
C SET FOR INITIAL RHS
      DO 1014 I=1,NST
      XKSI(IBST(I),NPER,1,1)=CBST(I,NCUR(I))
1014  CONTINUE
C
C          ADD FOR RHS
C
      JCUR=JCUR-1
      JPER(NPER-1)=JPOLD
      CALL BPRODX
      JPER(NPER-1)=1
C
C          YBX KEEPS INVENTORIES
C
      DO 27 I=2,KNR
      YBX(I)=Y(I)
   27 CONTINUE
C
C           START OF CYCLE
C
   20 CONTINUE
      DO 28 I=2,KNR
      Y(I)=YBX(I)
   28 CONTINUE
```

```
           JCUR = NPER-1
          CALL FRMRHS
          JCUR = NPER
C
C             IF 'IBASE'=1 FIND THE FIRST OPTIMAL
C
          IF(IBASE .GT. 1) GO TO 21
C     SET JCUR BACK TO NPER FOR CHECKING
          JCUR = NPER
          ITSINU=9999
          CALL NORMAL(ITSINU)
          IF(JCUR.EQ.NPER+1) RETURN
          GO TO 211
   21 CONTINUE
          WRITE(6,2201)
2201  FORMAT(' ITERATING IN DNORML')
          DO 1053 I=1,KNR
          Y(I)=B(I,NPER,1,1)
    1053  CONTINUE
          CALL FTRAN(1)
C
C                COPY INTO X.
C
          DO 1054 I=1,KNR
          X(I,NPER,1,1)=Y(I)
    1054  CONTINUE
          CALL DNORML
   211 CONTINUE
C         CALL STRPRT
          IF(MSTAT .NE. QN) GO TO 22
C
C             IF INFEASIBLE ADD A FEASCT AND RETURN.
C
          WRITE(6,2022) (NCUR(I),I=1,NST)
2022  FORMAT(' INFEAS AT PROB',3I6)
          JPER(NPER-1)=JPOLD
          CALL FEASCT
          JCUR=NPER-1
          RETURN
   22 CONTINUE
C
C             HERE IT'S FEASIBLE, SET FOR FIRST OF NEW BASIS.
C
          NETND(IBASE)=NETA
          JSTCH(NCUR(1),NCUR(2),NCUR(3))=IBASE
C
C             UPDATE EX VAL OF RHS
C
          KNR=NROW(NPER,1,1)
          XXR=0.
          DO 25 I=2,KNR
          XXR=YPI(I,NPER,1,1)*XKSI(I,NPER,1,1)+XXR
   25 CONTINUE
          WRITE(6,2024)
2024  FORMAT(' VAR1',2X,' VAR2',2X,' VAR3',2X,' PROB',3X,' OBJ',10X,
         1 'KSIPI',6X,'OBJTOT',5X,'CURR BAS')
          WRITE(6,2025) (NCUR(I),I=1,3),PRST(NCUR(1),NCUR(2),NCUR(3)),
         1 X(1,NPER,1,1),XXR,XTOPT,IBASE
```

```
 2025  FORMAT(3I6,1X,F8.6,1X,E11.3,1X,E11.3,1X,E11.3,1X,I6)
       XRHO=XXR*PRST(NCUR(1),NCUR(2),NCUR(3))+XRHO
C
C                UPDATE EX VAL OF PI'S
C
       DO 26 I=2,KNR
       YPIBAR(I)=YPI(I,NPER,1,1)*PRST(NCUR(1),NCUR(2),NCUR(3))+YPIBAR(I)
   26  CONTINUE
C
C              UPDATE E.V. OF OBJECTIVE
C
       XTOPT=XTOPT+PRST(NCUR(1),NCUR(2),NCUR(3))*X(1,NPER,1,1)
C
C              BEGIN LOOP TO CHECK FEASIBILITY M IS
C              THE CURRENT STOCHASTIC VARIABLE.
C
       M=NST
   29  CONTINUE
       IF(NCUR(M) .LT. INST(M)) GO TO 30
       M=M-1
       IF(M .GT. 0) GO TO 29
       IF(INFLG .NE. 0) GO TO 31
C
C               HERE, ALL NODES ARE COVERED.
C
       NFLG=1
       JPER(NPER-1)=JPOLD
       CALL XOPTCK
       IF(NFLG .EQ. 1) GO TO 32
C  PUT YBAR IN FOR PI
       DO 319 I=2,KNR
       YPI(I,NPER,JPER(2),1)=YPIBAR(I)
  319  CONTINUE
C      CALL STRPRT
       CALL LKHDCT(NPER)
   32  CONTINUE
       JCUR=NPER-1
       IF((NPER.EQ.2).AND.(NFLG.EQ.1)) JCUR= NPER + 1
       RETURN
C
C               THERE ARE STILL INFEAS'S LEFT.
C
   31  CONTINUE
C
C               RETURN TO SOLVE NEXT PROBLEM.
C
       WRITE(6,2301)
 2301  FORMAT(' RETURNING TO SOLVE NEXT BATCH')
       DO 40 I=1,NST
       XKSI(IBST(I),NPER,1,1)=CBST(I,NXNF(I))
   40  CONTINUE
       IBASE=IBASE+1
       INFLG=0
       DO 41 I=1,NST
       NCUR(I)=NXNF(I)
   41  CONTINUE
       GO TO 20
C
```

```
C             HERE, WE CHECK OUT THE NEXT POSSIBILITY.
C
   30   NCUR(M)=NCUR(M)+1
        MM=M+1
        IF(MM.GT.NST) GO TO 43
        DO 42 I=MM,NST
        NCUR(I)=1
     42 CONTINUE
 43     CONTINUE
C SET M BACK TO THE LAST PLAC
        M=NST
C
C             CHECK IF THIS IS COVERED.
C
        IF(JSTCH(NCUR(1),NCUR(2),NCUR(3)) .EQ. 0) GO TO 50
C
C             THERE, JSTCH WAS COVERED.
C
        GO TO 29
C
C             HERE, NOT COVERED, CHECK FEASIBILITY.
C
   50   CONTINUE
        DO 51 I=1,NST
        XKSI(IBST(I),NPER,1,1)=CBST(I,NCUR(I))
   51   CONTINUE
        DO 52 I=1,KNR
        Y(I)=YBX(I)
   52   CONTINUE
C
C             SET-UP RHS FOR CHECK.
C
C SET JCUR FOR FRMRHS
        JCUR=NPER-1
        CALL FRMRHS
        JCUR=NPER
        DO 53 I=1,KNR
        Y(I)=B(I,NPER,1,1)
   53   CONTINUE
        CALL FTRAN(1)
C
C             COPY INTO X.
C
        DO 54 I=1,KNR
        X(I,NPER,1,1)=Y(I)
   54   CONTINUE
        WRITE(6,2054)
 2054   FORMAT(' CHECKING FEAS WITH DCHUZR')
        CALL DCHUZR
        IF(IROWP .EQ. 0) GO TO 60
C
C            HERE, THE NODE IS OPTIMAL FOR THE GIVEN BASIS.
C
C            MUST CHECK FEASIBILITY.
C
        WRITE(6,2055)
 2055   FORMAT(' TRYING TO FIND ENTERING IN DCHUZC')
        CALL DCHUZC
```

```
      IF(JCOLP. NE. 0) GO TO 59
C
C            HERE INFEASIBLE.
C
      WRITE(6,2056)
 2056 FORMAT(' NON ENTERING VAR - INFEAS')
      JPER(NPER-1)=JPOLD
      CALL STRPRT
      CALL FEASCT
      JCUR=NPER-1
      RETURN
C
C           THERE, FEASIBLE
C
   59 CONTINUE
      WRITE(6,2059) (NCUR(I),I=1,3)
 2059 FORMAT(' ENT VAR - WILL RETURN TO',3I6)
      IF(INFLG .NE. 0) GO TO 29
C
C           FIRST INFEASIBLE FOUND.
C
      INFLG=1
      DO 58 I=1,NST
      NXNF(I)=NCUR(I)
   58 CONTINUE
      GO TO 29
C
C           THERE, IT IS FEASIBLE
C
   60 CONTINUE
      WRITE(6,2060)
 2060 FORMAT(' OK - FEAS')
C
C           UPDATE
C
      KNR=NROW(NPER,1,1)
      XXR=0.
      DO 255 I=2,KNR
      XXR=YPI(I,NPER,1,1)*XKSI(I,NPER,1,1)+XXR
  255  CONTINUE
      WRITE(6,2025) (NCUR(I),I=1,3),PRST(NCUR(1),NCUR(2),NCUR(3)),
     1 X(1,NPER,1,1),XXR,XTOPT,IBASE
      XRHO=XXR*PRST(NCUR(1),NCUR(2),NCUR(3))+XRHO
      XTOPT=XTOPT+PRST(NCUR(1),NCUR(2),NCUR(3))*X(1,NPER,1,1)
C
C           UPDATE PRICES.
C
      DO 61 I=1,KNR
      YPIBAR(I)=YPI(I,NPER,1,1)*PRST(NCUR(1),NCUR(2),NCUR(3))+YPIBAR(I)
   61 CONTINUE
C
C            SET BASIS INDEX.
C
      JSTCH(NCUR(1),NCUR(2),NCUR(3))=IBASE
      GO TO 29
      END
C-----------------------------------------------------------------------
      SUBROUTINE DCHUZR
```

```
C
C          SELECTS PIVOT ROW IROWP FOR CURRENT DUAL-SIMPLEX ITERATION.
C          SETS IROWP=0 IF CURRENT BASIS IS OPTIMAL.  OTHERWISE, IROWP IS
C          CHOSEN TO BE THE ROW WITH GREATEST PRIMAL INFEASIBILITY.
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      NINF=0
      IROWP = 0
      DP = -1.E10
      KNR=NROW(NPER,1,1)
      DO 1000 I=1,KNR
         IF (I .EQ. IOBJ) GO TO 1000
         ICOL = JH(I,NPER,1,1)
         IF (X(I,NPER,1,1) .LT. (XLB(ICOL,NPER,1,1) - ZTOLZE)) GO TO 100
         IF (X(I,NPER,1,1) .GT. (XUB(ICOL,NPER,1,1) + ZTOLZE)) GO TO 200
         GO TO 1000
C
C        BASIC VARIABLE ON ROW I FALLS BELOW ITS LOWER BOUND
100      DE = XLB(ICOL,NPER,1,1) - X(I,NPER,1,1)
      NINF=NINF+1
         IF (DE .LE. DP) GO TO 1000
         IPTYPE = 0
         GO TO 250
C
C        BASIC VARIABLE ON ROW I EXCEEDS ITS UPPER BOUND
200      DE = X(I,NPER,1,1) - XUB(ICOL,NPER,1,1)
      NINF=NINF+1
         IF (DE .LE. DP) GO TO 1000
         IPTYPE = -1
C
250      IROWP = I
         DP = DE
      YTEMP1(1)=DP
```

```
1000     CONTINUE
      RETURN
      END
C-----------------------------------------------------------------------
      SUBROUTINE DCHUZC
C
C         SELECTS PIVOT COLUMN JCOLP FOR CURRENT DUAL-SIMPLEX ITERATION.
C         SETS JCOLP=0 IF LP-PROBLEM AT CURRENT NODE IS INFEASIBLE.
C         OTHERWISE CHOOSES JCOLP TO MAINTAIN PRIMAL-OPTIMALITY.
C
      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
     1  INTEGER*4 (I-N,Q)
      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
      INTEGER ICNAM(602,2),NAME(6)
      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
      REAL A(3000,3,3,1)
C
      COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
     1  NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
     1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
     2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
     2  ),NETA,
     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
     3  602,3,3,1),
     4  LE(1002),IA(3000,3,3,1),IE(3000),
     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
C
      JCOLP = 0
      IF (IPTYPE .EQ. -1) GO TO 1000
C
C              LEAVING VARIABLE FALLS BELOW ITS LOWER BOUND
C
      DP = -1.E10
      KNC=NCOL(NPER,1,1)
      DO 500 J=1,KNC
         IF (KINBAS(J,NPER,1,1) .GT. 0) GO TO 500
         IF ((XUB(J,NPER,1,1) - XLB(J,NPER,1,1)) .LE. ZTOLZE) GO TO 500
         K = J
         CALL UNPACK(K)
         CALL FTRAN(1)
         IF (KINBAS(J,NPER,1,1) .EQ. -1) GO TO 200
         IF (Y(IROWP) + ZTOLPV) 225,225,500
200      IF (Y(IROWP) - ZTOLPV) 500,225,225
C
225      DE = Y(IOBJ)/Y(IROWP)
         IF (DE - DP) 500,500,250
250      JCOLP = J
         DP = DE
```

```
          YTEMP1(2)=Y(IROWP)
  500        CONTINUE
C
          IF (JCOLP .EQ. 0) RETURN
          CALL UNPACK(JCOLP)
          CALL FTRAN(1)
          ICOL = JH(IROWP,NPER,1,1)
          DP = (X(IROWP,NPER,1,1) - XLB(ICOL,NPER,1,1))/Y(IROWP)
          GO TO 2000
C
C               LEAVING VARIABLE EXCEEDS ITS UPPER BOUND
C
 1000     DP = 1.E10
          KNC=NCOL(NPER,1,1)
          DO 1500 J=1,KNC
             IF (KINBAS(J,NPER,1,1) .GT. 0) GO TO 1500
             IF ((XUB(J,NPER,1,1) - XLB(J,NPER,1,1)) .LE. ZTOLZE) GO TO 1500
             K = J
             CALL UNPACK(K)
             CALL FTRAN(1)
             IF (KINBAS(J,NPER,1,1) .EQ. -1) GO TO 1200
             IF (Y(IROWP) - ZTOLPV) 1500,1225,1225
 1200        IF (Y(IROWP) + ZTOLPV) 1225,1225,1500
C
 1225        DE = Y(IOBJ)/Y(IROWP)
             IF (DE - DP) 1250,1500,1500
 1250        JCOLP = J
             DP = DE
          YTEMP1(2)=Y(IROWP)
 1500        CONTINUE
C
          IF (JCOLP .EQ. 0) RETURN
          CALL UNPACK(JCOLP)
          CALL FTRAN(1)
          ICOL = JH(IROWP,NPER,1,1)
          DP = (X(IROWP,NPER,1,1) - XUB(ICOL,NPER,1,1))/Y(IROWP)
C
 2000     IF (KINBAS(JCOLP,NPER,1,1) .EQ. 0) DE = DP + XLB(JCOLP,NPER,1,1)
          IF (KINBAS(JCOLP,NPER,1,1) .EQ. -1) DE = DP + XUB(JCOLP,NPER,1,1)
          NPIVOT = 1
          RETURN
          END
```