

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

$$\begin{aligned}
\min z &= c_1 x_1 + E_{\xi_2} [\min c_2 x_2 + \dots + E_{\xi_t} [\min C_T x_T] \dots] & (1) \\
\text{s.t.} & A_1 x_1 &= b_1, \\
& B_1 x_1 + A_2 x_2 &= \xi_2, \\
& \dots & \vdots \\
& B_{T-1} x_{T-1} + A_T x_T &= \xi_T, \\
& x_t \geq 0, t=1, \dots, T, \quad t^t \quad t, t=2, \dots, T,
\end{aligned}$$

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 , ξ_t is a random m_2 - vector defined on the probability space (E_t, F_t, P_t) for $t=2, \dots, T$, and A_t and B_t are correspondingly 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])

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

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

$$D_t^{\ell,j} x_t^j \geq d_t^{\ell,j}, \ell=1, \dots, r_t^j, \quad (2.2)$$

$$x_t \geq 0,$$

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

Program (2) can then be solved using a relaxed master problem:

$$\min \quad c_t x_t^j + \theta_t^j \quad (3.0)$$

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

$$D_t^{\ell,j} x_t^j \geq d_t^{\ell,j}, \ell=1, \dots, r_t^j \quad (3.2)$$

$$E_t^{\ell,j} x_t^j + \theta_t^j \geq e_z^{\ell,j}, \ell=1, \dots, s_t^j \quad (3.3)$$

$$x_t^j \geq 0. \quad (3.4)$$

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 optimality cut (3.3) is added to (3) and (3) is re-solved. If \bar{x}_t^j forces infeasibility in any future period then a feasibility cut (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 scenarios 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 ancestor 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 , $j=1, \dots, K_2$. 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 , $j=1, \dots, K_t$. Solve (3) for $t+1$ and all $j=1, \dots, K_{t+1}$ 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

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

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}_τ^j , $\tau = 1, \dots, T$ form an optimal solution of (1).

Steps 1 and 2 of NDSPA represent a forward pass to obtain feasibility in each scenario subproblems. Step (3) is a backward pass 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

<u>Variable</u>	<u>Definition</u>
B(i,j,k,1)	Current right-hand side element i in period j and scenario k,1
X(i,j,k,1)	Current value of variable basic in row i at period j and scenario k,1
XLB(i,j,k,1) and XUB(i,j,k,1)	Lower and upper bounds of variable i at j,k,1
XKSI(i,j,k,1)	Current realization of random vector in row i at j,k,1
YPI(i,j,k,1)	Current dual variable value for row i at j,k,1
NROW(j,k,1)	Current number of rows at j,k,1
NCOL(j,k,1)	Current number of columns at j,k,1
NELM(j,k,1)	Current number of nonzero elements at j,k,1
JH(i,j,k,1)	Variable basic in row i at j,k,1
KINBAS(i,j,k,1)	Status (basic, nonbasic) of variable i at j,k,1
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,1)	Probability of scenario j,k,1

The important variables in BLOCK 3 are

NND(i)	Number of scenarios in period 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 $-e_T^{l,j} + E_T^{l,j} \bar{x}_T^j$ for checking for optimality
PRBY(i,j)	Probability of <u>jth</u> realization of <u>ith</u> random element in stochastic vector in period T
PRST(i,j,k)	Joint probability of <u>ith</u> realization of first random element element, <u>jth</u> realization of second, and <u>kth</u> realization of third for stochastic vector at T
CBST(i,j)	Value of <u>jth</u> realization of <u>ith</u> random element in stochastic vector at T
NCUR(i)	Current realization of <u>ith</u> random element at T
IBST(i)	Row of <u>ith</u> 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 <u>i</u> th column in A;
BUNPCK(i)	expands <u>i</u> th 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 NDSFA 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 ith random element and NXNF(i) is the realization of the ith random element in the first infeasible basis found by the bunching procedure. NETND(i) keeps the number of eta vectors in the ith 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 in 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 pre-determined 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 decomposition algorithms because of perhaps unavoidable numerical error present in generating cuts. Two truly 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


```

1 C ALGORITHM FOR STOCHASTIC PROGRAMS, WHERE
2 C THE RANDOM VARIABLES CAN BE IN EITHER THE CONSTRAINT
3 C MATRIX OR THE RIGHT HAND SIDE, AS LONG AS A DISCRETE
4 C DISTRIBUTION IS ASSUMED. THE L.P. SECTIONS OF
5 C THIS CODE ARE TAKEN FROM LPM-1 BY J. TOMLIN
6 C AND REVISED BY G. KOCHMAN.
7 C
8 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), RFAL*8 (B,D,X,Y).
9 1 INTEGER*4 (I-N,Q)
10 COMMON/TIMERS/ ITOT
11 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
12 C
13 C MAIN PROGRAM
14 C
15 C START TIMER
16 CALL TIME(O,O,ITOT)
17 CALL INIT
18 C INPUT PROBLEM DATA
19 CALL INPUT(IFPROB)
20 CALL INCHK
21 IF (IFPROB .EQ. O) GO TO 1000
22 CALL TIME(1,O,ITOT)
23 JTOT=ITOT
24 JCUR=1
25 DO 100 I=1,3
26 JPER(I)=1
27 CONTINUE
28 NODE=O
29 NPASS=O
30 JPASS=1
31 CONTINUE
32 ITSINV = 99999
33 NPASS=NPASS+1
34 IF(NPER.EQ.1) GO TO 200
35 IF(JCUR .EQ. NPER) GO TO 101
36 CALL NORMAL(ITSINV)
37 IF(JCUR.EQ.NPER+1) GO TO 201
38 C CHECK FOR CORRECT CUTS
39 CALL STRPRT
40 CALL ACHECK
41 CALL NDCOM
42 GO TO 200
43 CONTINUE
44 CALL PARSFT
45 CONTINUE
46 IF(NPER.EQ.1) JCUR=2
47 IF(JCUR .LE. NPER) GO TO 11
48 CONTINUE
49 CALL TIME(1,O,ITOT)
50 WRITE (6,1) ITOT
51 FORMAT (/' TOTAL SOLUTION TIME =',I6,' MILLISECONDS')
52 KTOT=ITOT-JTOT
53 WRITE(6,2) KTOT
54 FORMAT(' SOLUTION TIME WITHOUT INPUT = ',I6,' MILLISECONDS')
55 C OUTPUT OPTIMAL SOLUTION
56 CALL WRAPUP
57 GO TO 10
58 1000 STOP

```

```

59 END
60 C-----
61 BLOCK DATA
62 C
63 C INITIALIZES GLOBAL PROGRAM CONSTANTS
64 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
65 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
66 C
67 C
68 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
69 C INTEGER*4 (I-N,Q)
70 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
71 C NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QP
72 C DATA ZTOLZ/1.E-7/,ZTOLPV/1.E-5/,ZTCOST/1.E-4/,ZTOLSM/1.E-10/
73 C DATA NRMAX/350/,NTMAX/1000/,NEMAX/3000/,NEGINF/-100000/
74 C DATA QBL/'',QA/'A',QI/'I',QF/'F',QN/'N',
75 C QB/'B',QC/'C',QE/'E',QH/'H',
76 C QL/'L',QO/'O',QR/'R',QM/'M',QG/'G',
77 C OS/'S',QP/'P'
78 C-----
79 SUBROUTINE RHCHK
80 C
81 C THIS SUBROUTINE CHECKS THE ROW RESIDUALS
82 C
83 C
84 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
85 C INTEGER*4 (I-N,Q)
86 C INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
87 C INTEGER ICNAM(602,2,3,3,1),ICN(602,2),NAME(6)
88 C DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
89 C REAL A(3000,3,3,1)
90 C
91 C COMMON/BL5/DRES,ICNAM
92 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
93 C NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QP
94 C COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
95 C 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
96 C 1 A,E,MSTAT,IOBJ,IROWP,ITCNT
97 C 2 INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
98 C ),NETA,
99 C 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
100 C 602,3,3,1),
101 C 4 LE(1002),IA(3000,3,3,1),IE(3000),
102 C 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
103 C 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
104 C 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
105 C COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
106 C COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
107 C 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
108 C 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
109 C JC = JCUR
110 C J2=JPER(2)
111 C J3 = JPER(3)
112 C KNR=NROW(JC,J2,J3)
113 C KNC=NCOL(JC,J2,J3)
114 C DO 90 I=1,KNR
115 C YTEMP(I)=B(I,JC,J2,J3)
116 C YTEMP1(I)=0.

```

```

117 90 CONTINUE
118 C WRITE(6,91)
119 C1 FORMAT(' CHECKING ROW RESIDUAL')
120 DMAX=0.
121 DO 200 J=1,KNC
122 CALL UNPACK(J)
123 KB=KINBAS(J,JC,J2,J3)
124 IF(KB.GT.O) GO TO 102
125 DO 100 I=1,KNR
126 IF(KB.LT.O) GO TO 101
127 YTEMP(I)=YTEMP(I) - Y(I)*XLB(J,JC,J2,J3)
128 GO TO 100
129 CONTINUE
130 YTEMP(I) = YTEMP(I) - Y(I)*XUB(J,JC,J2,J3)
131 CONTINUE
132 GO TO 200
133 CONTINUE
134 DO 103 I=1,KNR
135 YTEMP1(I) = YTEMP1(I) + Y(I)*X(KB,JC,J2,J3)
136 CONTINUE
137 CONTINUE
138 DO 300 I=1,KNR
139 DF = DABS(YTEMP(I)-YTEMP1(I))
140 IF(DF.GT.DMAX) DMAX = DF
141 IF(DF.LE.ZTOLZE) GO TO 300
142 CONTINUE
143 WRITE(6,302) DMAX
144 FORMAT(' DMAX = ',E9.2)
145 DRES = DMAX
146 RETURN
147 END
148
149 C-----
150 C SUBROUTINE INPUT(IFPROB)
151 C
152 C INPUTS PROBLEM DATA
153 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
154 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
155 C ***DESCRIPTION OF PARAMETERS***
156 C IFPROB = NONZERO PROBLEM ID NUMBER(OUTPUT)
157 C
158 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
159 C1 INTEGER*4 (I-N,Q)
160 C2 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
161 C3 INTEGER ICNAM(602,2,3,3,1),ICN(602,2),NAME(6)
162 C4 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
163 C5 REAL A(3000,3,3,1)
164 C
165 C COMMON/BL5/DRES,ICNAM
166 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
167 C1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QQ
168 C2 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
169 C3 22),XLB(602,3,3,1),XUB(602,3,3,1),XKST(350,3,3,1),YPT(350,3,3,1)
170 C4 ,A,E,MSTAT,IOBJ,IROWP,ITCNT
171 C5 2 INVFRQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
172 C6 ),NETA,
173 C7 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
174 C8 602,3,3,1),
175 C9 LE(1002),IA(3000,3,3,1),IE(3000),

```

```

175      5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
176
177      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
178      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
179      COMMON/BLOCK3/ NND(5),NPASS,UPER(5),JCUR,JPASS,NPER
180      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
181      1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),HOUR(3),
182      2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
183
184      C
185      C ***DESCRIPTIONS OF SOME IMPORTANT VARIABLES IN BLANK COMMON***
186      C B(I) = RIGHT HAND SIDE OF ROW I
187      C X(I) = LP VALUE FOR JH(I), WHICH IS THE VARIABLE BASIC IN ROW I
188      C A CONTAINS THE NONZERO ELEMENTS OF THE CONSTRAINT MATRIX, INCL.
189      C THE OBJECTIVE ROW IOBJ. LA(J) = LOCATION IN A OF THE FIRST
190      C ELEMENT OF COL J. IA(I) = ROW IN WHICH ELEMENT I OF A BELONGS.
191      C E CONTAINS THE NONZERO ELEMENTS OF THE CURRENT LP BASIS INVERSE
192      C IN ETA VECTOR FORM. LE, IE ARE TO E AS LA, IA ARE TO A.
193      C MSTAT FLAGS FEASIBILITY OF CURRENT LP
194      C ITCNT = NO. OF SIMPLEX ITERATIONS SO FAR; IF > ITRFRQ, STOP
195      C INVFRQ = NUMBER OF SIMPLEX ITERATIONS BEFORE E IS REINVERTED
196      C NROW = NO. OF ROWS; NCOL = NO. OF COLUMNS
197      C KINBAS(J) = { I IF J IS BASIC IN ROW I, I.E. J = JH(I)
198      C               O IF J IS NONBASIC AT ITS LOWER BOUND XLR(J)
199      C               -1 IF J IS NONBASIC AT ITS UPPER BOUND XUB(J) }
200
201      DO 21 K=1,3
202      DO 21 K1=1,2
203      DO 21 K2=1,2
204      DO 10 I=1,NRMAX
205      XKSI(I,K,K1,K2)=O.O
206      B(I,K,K1,K2) = O.
207      XKSI(I,K,K1,K2) = O.O
208      NROW(K,K1,K2) = O
209      L = 500
210      DO 20 J=1,L
211      KINBAS(J,K,K1,K2) = O
212      CONTINUE
213      ITCNT = O
214      ICS1=O
215      ICS2=O
216      C SET FOR MXNST POSSIBILITIES
217      MXNST=5
218      READ (5,7000,END=9999) IFPROB,IOBJ,INVFRQ,ITRFRQ,NPER
219      FORMAT (5I4)
220      DO 7002 I=1,NPER
221      READ(5,7001,END=9999) NND(I)
222      FORMAT(I4)
223      CONTINUE
224      IF (IFPROB .EQ. O) RETURN
225      IF (IOBJ .EQ. O)IOBJ = 1
226      IF (INVFRQ .EQ. O)INVFRQ = 99999
227      IF (ITRFRQ .EQ. O)ITRFRQ = 99999
228      WRITE(6,8010) IFPROB
229      FORMAT(' PROBLEM ',I4)
230
231      C
232      C START TO ITERATE
233      JCUR=1
234      DO 19 I=1,3

```

```

233 JPER(I)=1
234 CONTINUE
235 JPER(1)=0
236 CONTINUE
237 NR=0
238 JPER(JCUR)=JPER(JCUR)+1
239 IF(JPER(JCUR) .LE. NND(JCUR)) GO TO 197
240 K=JCUR-1
241 CONTINUE
242 IF(K .LE. 1) GO TO 188
243 IF(JPER(K) .GE. NND(K)) K=K-1
244 IF(JPER(K) .GE. NND(K)) GO TO 187
245 JPER(K)=JPER(K)+1
246 GO TO 186
247 CONTINUE
248 IF(JCUR .EQ. NPER) GO TO 9998
249 JCUR=JCUR+1
250 CONTINUE
251 KK=K+1
252 IF(K .EQ. 0) K=1
253 DO 189 I=KK,JCUR
254 JPER(I)=1
255 CONTINUE
256 CONTINUE
257 JC=JCUR
258 J1=JPER(1)
259 J2=JPER(2)
260 J3=JPER(3)
261 NROW(JC,J2,J3)=0
262 NCOL(JC,J2,J3)=0
263 NELM(JC,J2,J3)=0
264 READ(5,222) PROB(JC,J2,J3)
265 FORMAT(F5.3)
266 READ(5,101) K1,K2,K3,K4,(NAME(I),I=1,4),ATEMP1,NAME(5),NAME(6),
267 1 ATEMP2
268 FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
269 IF(K1 .EQ. QE) GO TO 600
270 IF(K1 .EQ. QBL) GO TO 50
271 IF(K1 .EQ. QN) GO TO 5
272 IF(K1 .EQ. QR .AND. K2 .EQ. QO) L=1
273 IF(K1 .EQ. QR .AND. K2 .EQ. QO) GO TO 5
274 IF(K1 .EQ. QC) L=2
275 IF(K1 .EQ. QC) GO TO 5
276 IF(K1 .EQ. QB .AND. K2 .EQ. QA) L=3
277 IF(K1 .EQ. QB .AND. K2 .EQ. QA) GO TO 5
278 IF(K1 .EQ. QR .AND. K2 .EQ. QH) L=4
279 IF(K1 .EQ. QR .AND. K2 .EQ. QH) GO TO 5
280 GO TO(210,320,410,500),L
281 C
282 C
283 NROW(JC,J2,J3)=NROW(JC,J2,J3)+1
284 NCOL(JC,J2,J3)=NCOL(JC,J2,J3)
285 ICNAM(NROW(JC,J2,J3),1,JC,J2,J3) = NAME(1)
286 ICNAM(NROW(JC,J2,J3),2,JC,J2,J3) = NAME(2)
287 C
288 C
289 C
290 TEST ROW TYPE
IF(K2.EQ.QL .OR. K3.EQ.QL) GO TO 220
IF(K2.EQ.QE .OR. K3.EQ.QE) GO TO 230

```

```

291 IF(K2.EQ.QG .OR. K3.EQ.QG) GO TO 240
292 IF(K2.EQ.QN .OR. K3.EQ.QN) GO TO 250
293 GO TO 230
294
295 XLB(NROW(JC,J2,J3),JC,J2,J3) = O.
296 XUB(NROW(JC,J2,J3),JC,J2,J3) = 1.E7
297 GO TO 250
298 XLB(NROW(JC,J2,J3),JC,J2,J3) = O.
299 XUB(NROW(JC,J2,J3),JC,J2,J3) = O.
300 GO TO 250
301 XLB(NROW(JC,J2,J3),JC,J2,J3) = O.
302 XUB(NROW(JC,J2,J3),JC,J2,J3) = 1.E7
303 A(NROW(JC,J2,J3),JC,J2,J3) = -1.
304 GO TO 260
305 A(NROW(JC,J2,J3),JC,J2,J3) = 1.
306 IA(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
307 LA(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
308 JH(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
309 KINBAS(NROW(JC,J2,J3),JC,J2,J3) = NROW(JC,J2,J3)
310 NELEM=NROW(JC,J2,J3)
311 GO TO 5
312
313 C
314 C
315 C
316 C
317 C
318 C
319 C
320 C
321 C
322 C
323 C
324 C
325 C
326 C
327 C
328 C
329 C
330 C
331 C
332 C
333 C
334 C
335 C
336 C
337 C
338 C
339 C
340 C
341 C
342 C
343 C
344 C
345 C
346 C
347 C
348 C
220
230
240
250
260
320
321
324
330
335
340
8300
MATRIX ELEMENTS
J = 3
K = 4
IF (DABS(ATEMP1) .LE. ZTOLZE) GO TO 321
GO TO 324
J=5
K=6
IF (DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
ATEMP1=ATEMP2
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
TEST FOR ROW MATCH
KNR=NROW(JC,J2,J3)
DO 340 I=1,KNR
IF(NAME(I) .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
IF(K .GT. 5) GO TO 5
IF(DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
J = 5
K = 6
ATEMP1 = ATEMP2
GO TO 330
CONTINUE
WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
FORMAT(17HNO MATCH FOR ROW ,2A4,10HAT COLUMN ,2A4)

```

```

349      STOP
350      C
351      C      BASIS CARDS
352      C
353      410      KNC=NCOL(JC,J2,J3)
354      DO 420 I=1,KNC
355      IF(NAME(I).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(2).NE.
356      1 ICNAM(I,2,JC,J2,J3)) GO TO 420
357      IBVEC = I
358      GO TO 425
359      CONTINUE
360      WRITE(6,8400) NAME(1),NAME(2)
361      FORMAT(20HNO MATCH FOR VECTOR ,2A4)
362      GO TO 5
363      425      KNR=NROW(JC,J2,J3)
364      DO 430 I=1,KNR
365      IF(NAME(3).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(4).NE.
366      1 ICNAM(I,2,JC,J2,J3)) GO TO 430
367      IBROW=I
368      GO TO 440
369      CONTINUE
370      WRITE(6,8300) NAME(3),NAME(4)
371      GO TO 5
372      440      JH(IBROW,JC,J2,J3) = IBVEC
373      KINBAS(IBROW,JC,J2,J3) = 0
374      KINBAS(IBVEC,JC,J2,J3) = IBROW
375      GO TO 5
376      C
377      C      RHS
378      C
379      500      J = 3
380      K = 4
381      IF (DABS(ATEMP1) .LE. ZTOLZE) GO TO 521
382      GO TO 530
383      J=5
384      K=6
385      IF (DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
386      ATEMP1=ATEMP2
387      C
388      C      TEST FOR ROW MATCH
389      C
390      530      KNR=NROW(JC,J2,J3)
391      DO 540 I=1,KNR
392      IF(NAME(I).NE.ICNAM(I,1,JC,J2,J3).OR.NAME(K).NE.
393      1 ICNAM(I,2,JC,J2,J3)) GO TO 540
394      B(I,JC,J2,J3) = ATEMP1
395      XKSI(I,JC,J2,J3) = ATEMP1
396      IF(K.GT. 5) GO TO 5
397      IF(DABS(ATEMP2) .LE. ZTOLZE) GO TO 5
398      J = 5
399      K = 6
400      ATEMP1 = ATEMP2
401      GO TO 530
402      CONTINUE
403      540      WRITE(6,8300) NAME(J),NAME(K)
404      STOP
405      C
406      C      END OF INPUT

```

```

407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464

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)
    FORMAT (9F8.0)
650 NEM(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 INPUT FOR THE B MATRIX FOLLOWS
C
15 NELEM=0
    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. QO) L=1
    IF (K1 .EQ. QR .AND. K2 .EQ. QO) GO TO 15
    IF (K1 .EQ. QC) L=2
    IF (K1 .EQ. QC) GO TO 15
    GO TO (115,1320),L
150
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

```



```

465 ATEMP1=ATEMP2
466 IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 1330
467 NC = NC + 1
468 ICS1 = NAME(1)
469 ICS2 = NAME(2)
470 LBN(NC,JC,J2,J3) = NELEM + 1
471
472 C TEST FOR ROW MATCH
473 C
474
475 DO 1340 I = 1,NR
476 IF (NAME(J) .NE. ICN(I,1) .OR. NAME(K) .NE. ICN(I,2)) GO TO 1340
477 NELEM = NELEM + 1
478 IBN(NELEM,JC,J2,J3) = I
479 ABN(NELEM,JC,J2,J3) = ATEMP1
480 LBN(NC+1,JC,J2,J3)=NELEM+1
481 IF (K .GT. 5) GO TO 15
482 IF (DABS(ATEMP2) .LE. ZTOLZE) GO TO 15
483 J = 5
484 K = 6
485 ATEMP1 = ATEMP2
486 GO TO 1330
487 CONTINUE
488 WRITE(6,8300) NAME(J),NAME(K),NAME(1),NAME(2)
489 CONTINUE
490 GO TO 22
491 CONTINUE
492
493 C HERE WE HAVE NEW POSSIBILITIES
494 C INITIALIZE
495 C
496
497 NST=0
498 DO 2102 I=1, MXNST
499 INST(I)=0
500 CONTINUE
501
502 C READ IN VALUES
503
504 READ(5,2101) K1,K2,K3,K4,(NAME(I), I=1,4),ATEMP1,
505 1 NAME(5),NAME(6),ATEMP2
506 FORMAT(4A1,2A4,2X,2A4,2X,F12.4,3X,2A4,2X,F12.4)
507 IF (K1 .EQ. QS) L=1
508 IF (K1 .EQ. QS) GO TO 2505
509 IF (K1 .EQ. QE) GO TO 2600
510
511 C CHECK FOR NEW VALUES
512
513 IF (NAME(1) .EQ. ICS1 .AND. NAME(2) .EQ. ICS2) GO TO 2330
514 ICS1=NAME(1)
515 ICS2=NAME(2)
516 NST=NST+1
517 DO 2340 I=1,KNR
518 IF (NAME(1) .NE. ICNAM(I,1,JC,J2,J3) .OR. NAME(2) .NE.
519 1 ICNAM(I,2,JC,J2,J3)) GO TO 2340
520 IBST(NST)=I
521 INST(NST)=INST(NST) + 1
522 CBST(NST,INST(NST))=ATEMP1
523 PRBV(NST,INST(NST))=ATEMP2

```

```

523 GO TO 2505
524 CONTINUE
525 GO TO 2349
526 CONTINUE
527 INST(NST)=INST(NST)+1
528 CBST(NST,INST(NST))=ATEMP1
529 PRBV(NST,INST(NST))=ATEMP2
530 GO TO 2505
531 C
532 C CHECK FOR ROW MATCH
533 C
534 C
535 C
536 C
537 C
538 WRITE(6,2350) NAME(3),NAME(4),NAME(1),NAME(2)
539 FORMAT(17HNO MATHCH FOR ROW,2A4,18HAT STOCH VARIABLE,2A4)
540 STOP
541 C
542 C SET UP RHS
543 C
544 C
545 C
546 C
547 C
548 C
549 C CHECK IF MXNST>NST
550 C
551 IF(NST.EQ.3) GO TO 2376
552 MNST=3
553 IF(NST.LT.3) MNST=NST+1
554 DO 2375 I=NNST,3
555 INST(I)=1
556 PRBV(I,1)=1.0
557 CONTINUE
558 CONTINUE
559 INS=INST(1)
560 JNS=INST(2)
561 KNS=INST(3)
562 DO 2380 I=1,INS
563 DO 2380 J=1,JNS
564 DO 2380 K=1,KNS
565 PRST(I,J,K)=PRBV(1,I)*PRBV(2,J)*PRBV(3,K)
566 CONTINUE
567 WRITE(6,2400)
568 FORMAT(/,' STOCHASTIC VARIABLES')
569 WRITE(6,2401)
570 FORMAT(/,' ROW',8X,' VALUE',4X,' PROB',/)
571 DO 2410 I=1,3
572 IF(I.GT.NST) GO TO 2410
573 J1 = INST(I)
574 DO 2410 J=1,J1
575 I1=IBST(I)
576 A1=CBST(I,J)
577 A2=PRBV(I,J)
578 WRITE(6,2402) ICNAM(I1,1,NPER,1,1),ICNAM(I1,2,NPER,1,1),A1,A2
579 FORMAT(1X,2A4,3X,F8.2,2X,F8.6)
580 CONTINUE

```

```

581 9998 CONTINUE
582 C SET FOR NO THETA CUTS TO START
583 DO 1601 I=1,NPER
584 NNDD=NND(I)
585 DO 1602 J=1,NNDD
586 NTH(I,J)=O
587 CONTINUE
588 1601 CONTINUE
589 RETURN
590 IFPROB = O
591 END
592
593 -----
594 SUBROUTINE INCHK
595
596 C CHECKS OUT THE CURRENT VALUES
597
598 C
599 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
600 1 INTEGER*4 (I-N,Q)
601 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
602 INTEGER ICNAM(602,2),NAME(6)
603 DOUBLE PRECISION E(3000).ATEMP1,ATEMP2
604 REAL A(3000,3,3,1)
605
606 C
607 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
608 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YF1(350,3,3,1)
609 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
610 2 INVFRQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NEIM(3,3,1)
611 2 ),NETA,
612 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
613 3 602,3,3,1),
614 4 LE(1002),IA(3000,3,3,1),IE(3000),
615 5 ATPM(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1).
616
617 C
618 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XITMP(602),
619 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
620 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
621 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIRAP(602),YBX(350),IBST(3),
622 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
623 2 NXNF(3),INFLG,NEIND(10),INST(5),MXNST,NST
624 L=1
625 DO 100 J=1,NPER
626 NNDD=NND(J)
627 DO 99 K=1,NNDD
628 IF((J.EQ. 1) .AND. (K.GT. 1)) GO TO 99
629 NNPP=NND(3)
630 DO 80 L=1, NNPP
631 IF((J.NE. 3) .AND. (L.GT.1)) GO TO 80
632 KLM=NELM(J,K,L)
633 KNC=NCOL(J,K,L)
634 KNR=NROW(J,K,L)
635 WRITE (6,1) J,K,L
636 FORMAT(' CHECK THE A MATRIX OF NODE ',I4,' ',I4,' ',I4,' //')
637 WRITE (6,2) (A(I,J,K,L),I=1,KLM)
638 FORMAT(15F7.2)
639 WRITE(6,3)
640 WRITE(//,' IA')
641 WRITE(6,5) (IA(I,J,K,L),I=1,KLM)

```

```

639 5 FORMAT(15I5)
640 WRITE(6,4)
641 FORMAT(//, ' LA')
642 WRITE(6,5) (LA(I,J,K,L), I=1,KNC)
643 WRITE(6,6) KNR, KNC, KLM
644 FORMAT(//, ' NROW=', I5, ' NCOL=', I5, ' NELM=', I5)
645 WRITE(6,7)
646 FORMAT(' KINBAS')
647 WRITE(6,5) (KINBAS(I,J,K,L), I=1,KNC)
648 WRITE(6,8)
649 FORMAT(' JH')
650 WRITE(6,5) (JH(I,J,K,L), I=1,KNR)
651 WRITE(6,9)
652 FORMAT(' X VALUES')
653 WRITE(6,2) (X(I,J,K,L), I=1,KNR)
654 WRITE(6,10)
655 FORMAT(' XKSI "S ')
656 WRITE(6,2) (XKSI(I,J,K,L), I=1,KNR)
657 WRITE(6,2) (B(I,J,K,L), I=1,KNR)
658 WRITE(6,11) J,K,L, PROB(J,K,L)
659 FORMAT(' PROB OF', I5, I5, I5, ' IS', F7.5)
660 WRITE(6,12)
661 FORMAT(' ABN MATRIX')
662 WRITE(6,2) (ABN(I,J,K,L), I=1,KIM)
663 WRITE(6,13)
664 FORMAT(' IBN')
665 WRITE(6,5) (IBN(I,J,K,L), I=1,KLM)
666 WRITE(6,14)
667 FORMAT(' LBN')
668 WRITE(6,5) (LBN(I,J,K,L), I=1,KNC)
669 CONTINUE
670 CONTINUE
671 100 CONTINUE
672 RETURN
673 END
674 C-----
675 SUBROUTINE FTRAN(IPAR)
676 C
677 C PERFORMS FORWARD TRANSFORMATION ON COLUMN STORED VECTOR Y
678 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
679 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
680 C ***DESCRIPTION OF PARAMETERS***
681 C IPAR = PARAMETER INDICATING WHICH ETA-VECTORS MATRIX COLUMN
682 C IS TO BE UPDATED BY (INPUT)
683 C
684 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y)
685 1 INTEGER*4 (I-N,Q)
686 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
687 DOUBLE PRECISION E(3000)
688 REAL A(3000,3,3,1)
689 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
690 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
691 1 ,A,E,MSTAT,I0BJ,IROWP,ITCNT
692 2 INVFREQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
693 2 ) ,NETA
694 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
695 3 602,3,3,1)
696 4 LE(1002),IA(3000,3,3,1),IE(3000)

```

```

697      5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
698
699      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XUTMP(602),
700      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
701      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
702      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
703      1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
704      2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
705
706      NLE = NETA
707      NFE = 1
708      IF (IPAR.EQ.2) NFE = NLETA + 1
709      IF (NFE.GT. NLE) GO TO 9000
710      DO 1000 IK = NFE,NLE
711      LL = LE(IK)
712      KK = LE(IK+1) - 1
713      IPIV = IE(LL)
714      DY = Y(IPIV)
715      DY = DY/E(LL)
716      Y(IPIV) = DY
717      IF (KK.LE. LL) GO TO 1000
718      LL = LL + 1
719      DO 500 J = LL,KK
720      IR = IE(J)
721      Y(IR) = Y(IR) - E(J) * DY
722      CONTINUE
723      500 CONTINUE
724      1000 CONTINUE
725      RETURN
726      END
727
728      C-----
729      SUBROUTINE BTRAN
730
731      C PERFORMS BACKWARD TRANSFORMATION ON COLUMN STORED IN VECTOR Y
732      C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
733      C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
734
735      C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y)
736      1  INTEGER*4 (I-N,Q)
737      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
738      DOUBLE PRECISION E(3000)
739      REAL A(3000,3,3,1)
740      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YIEMP1(6
741      1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YF1(350,3,3,1)
742      1  ,A,E,MSTAT,IOBJ,IROWP,ITCNT
743      2  INVFRO,ITFRFRO,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFILM(3,3,1
744      2  ),NETA,
745      3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
746      3  602,3,3,1),
747      4  LE(1002),IA(3000,3,3,1),IE(3000),
748      5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
749
750      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XUTMP(602),
751      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
752      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
753      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
754      1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
755      2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST

```

```

755 C
756 IF (NETA .LE. O) GO TO 9000
757 DO 1000 I = 1,NETA
758 IK = NETA - I + 1
759 LL = LE(IK)
760 KK = LE(IK+1) - 1
761 IPIV = IE(LL)
762 DP = E(LL)
763 DY = Y(IPIV)
764 DSUM = O.
765 IF (KK .LE. LL) GO TO 600
766 LL = LL + 1
767 DO 500 J = LL,KK
768 IR = IE(J)
769 DE = E(J)
770 DPROD = DE * Y(IR)
771 DSUM = DSUM + DPROD
772 CONTINUE
773 500
774 C
775 Y(IPIV) = (DY - DSUM) / DP
776 CONTINUE
777 JC=JCUR
778 J2=JPER(2)
779 J3=JPER(3)
780 KNR=NROW(JC,J2,J3)
781 DO 700 I=1,KNR
782 YPI(I,JC,J2,J3)=Y(I)
783 CONTINUE
784 700
785 RETURN
786 9000
787 END
788 C-----
789 SUBROUTINE FORMC
790 C
791 C FORMS OBJ. FUNCTION VECTOR AND CHECKS FEASIBILITY OF BASIS
792 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
793 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
794 C
795 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
796 C 1 INTEGER*4 (I-N,Q)
797 C INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
798 C DOUBLE PRECISION E(3000)
799 C REAL A(3000,3,3,1)
800 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
801 C NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QR,QM,QQ,QS,QP
802 C COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,MINF,NOPT
803 C COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
804 C 22),XLB(602,3,3,1),XJB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
805 C 1 A,E,MSTAT,IOBJ,IROWP,ITCNT,
806 C 2 INVFRQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELEM(3,3,1
807 C ),NETA,
808 C 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
809 C 602,3,3,1),
810 C 4 LE(1002),IA(3000,3,3,1),IE(3000),
811 C 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
812 C 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
813 C 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
814 C COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER

```

```

813 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
814 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
815 2 NXNF(3),INFLG,NETND(10),INST(5),MXNSI,NST
816
817 C
818 NINF=0
819 MSTAT = QF
820 Y(IOBJ) = 0.
821 JC=JCUR
822 J2=JPER(2)
823 J3=JPER(3)
824 KNR=NROW(JC,J2,J3)
825 DO 30 I=1,KNR
826 IF (I.EQ. IOBJ) GO TO 30
827 ICOL = JH(I,JC,J2,J3)
828 IF (X(I,JC,J2,J3) .LE. (XLB(ICOL,JC,J2,J3) - ZTOLZE)) GO TO 10
829 IF (X(I,JC,J2,J3) .GE. (XUB(ICOL,JC,J2,J3) + ZTOLZF)) GO TO 20
830 Y(I) = 0.
831 GO TO 30
832 Y(I) = 1.
833 MSTAT = QI
834 NINF=NINF+1
835 GO TO 30
836 Y(I) = -1.
837 NINF=NINF+1
838 MSTAT = QI
839 CONTINUE
840 IF (MSTAT.EQ.QF) Y(IOBJ) = 1.
841 RETURN
842 END
843 C-----
844 SUBROUTINE PRICE
845 C
846 C PRICES OUT NONBASIC COLUMNS; CHOOSES PIVOT COLUMN JCOLP FOR
847 C CURRENT PRIMAL SIMPLEX ITERATION
848 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
849 C BY J. A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
850 C
851 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
852 1 INTEGER*4 (I-N,Q)
853 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
854 DOUBLE PRECISION E(3000)
855 REAL A(3000,3,3,1)
856 C
857 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
858 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QQ,QS,QP
859 COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IFTYPE,CMIN,CMAX,APV,NINF,NOPT
860 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YIEMP1(6
861 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
862 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
863 2 INVFREQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
864 ),NETA,
865 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
866 3 602,3,3,1),
867 4 LE(1002),IA(3000,3,3,1),IE(3000),
868 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
869 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
870 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR

```

```

871 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
872 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
873 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
874 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
875 C
876 CMIN = 1.E10
877 NOPT=0
878 CMAX = -1.E10
879 JC=JCUR
880 J2=JPER(2)
881 J3=JPER(3)
882 KNC=NCOL(JC,J2,J3)
883 DO 1000 J=1,KNC
884 IF (KINBAS(J,JC,J2,J3).GT.0) GO TO 1000
885 IF ((XUB(J,JC,J2,J3) - XLB(J,JC,J2,J3)) .LT. ZTOLZF) GO TO 1000
886 DSUM = 0.
887 LL = LA(J,JC,J2,J3)
888 KK = LA(J+1,JC,J2,J3) - 1
889 DO 500 I = LL, KK
890 IR = IA(I,JC,J2,J3)
891 DE = A(I,JC,J2,J3)
892 DPRD = DE * Y(IR)
893 DSUM = DSUM + DPRD
894 CONTINUE
895 IF (KINBAS(J,JC,J2,J3).EQ.-1) GO TO 600
896 IF(DSUM.LT.-ZTOLZE)NOPT=NOPT+1
897 IF (DSUM .GE. CMIN) GO TO 1000
898 CMIN = DSUM
899 JCOL1 = J
900 GO TO 1000
901 600 IF(DSUM.GT.ZTOLZE)NOPT=NOPT+1
902 IF (DSUM .LE. CMAX) GO TO 1000
903 CMAX = DSUM
904 JCOL2 = J
905 CONTINUE
906 C
907 IF (CMIN .LE. -ZTCOST) GO TO 1500
908 IF (CMAX .GE. ZTCOST) GO TO 2000
909 JCOLP = 0
910 RETURN
911 IF (CMAX .GE. ZTCOST) GO TO 2500
912 1500 JCOLP = JCOL1
913 RETURN
914 2000 JCOLP = JCOL2
915 RETURN
916 2500 IF (ABS(CMIN) - CMAX) 2000,2000,1600
917 END
918 C-----
919 SUBROUTINE CHUZR
920 C
921 C PERFORMS MIN-RATIO TEST FOR PIVOT COLUMN JCOLP DETERMINED IN
922 C SUBROUTINE PRICE. SELECTS PIVOT ROW IROWP FOR CURRENT PRIMAL
923 C SIMPLEX ITERATION.
924 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
925 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
926 C
927 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
928 1 INTEGER*4 (I-N,Q)

```



```

929 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
930 DOUBLE PRECISION E(3000)
931 REAL A(3000,3,3,1)
932
933 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZICOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
934 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QQ,OS,OP
935 COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,MINF,NODT
936 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350,3,3,1),YTEMP(602),YTEMP1(G
937 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YFI(350,3,3,1)
938 1 .A.E,MSTAT,IOBJ,IROWP,ITCNT,
939 2 INVERQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
940 2 ),NETA,
941 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
942 3 602,3,3,1),
943 4 LE(1002),IA(3000,3,3,1),IE(3000),
944 5 ATMP(3000),ABN(600,3,3,1),IEN(600,3,3,1),LBN(602,3,3,1),
945
946 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
947 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
948 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
949 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
950 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),HOUR(3),
951 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
952
953 JC=JCUR
954 J2=JPER(2)
955 J3=JPER(3)
956 IF (KINBAS(JCOLP,JC,J2,J3) .EQ. -1) GO TO 1000
957
958 C
959 INCOMING VARIABLE AT LOWER ROUND
960
961 DP = 1.E10
962 KNR=NROW(JC,J2,J3)
963 DO 500 I=1,KNR
964 IF (I .EQ. IOBJ) GO TO 500
965 ICOL = JH(I,JC,J2,J3)
966 IF (Y(I) .GT. ZTOLPV) GO TO 100
967 IF (Y(I) .LT. -ZTOLPV) GO TO 200
968 GO TO 500
969 C
970 POSITIVE COEFFICIENT
971 IF (X(I,JC,J2,J3) .LT. (XLB(ICOL,JC,J2,J3) - ZTOLZF))
972 GO TO 500
973 DE = (X(I,JC,J2,J3) - XLB(ICOL,JC,J2,J3))/Y(I)
974 IF (DE .GE. DP) GO TO 500
975 IPTYPE = 0
976 GO TO 250
977 C
978 NEGATIVE COEFFICIENT
979 IF (X(I,JC,J2,J3) .GT. (XUB(ICOL,JC,J2,J3) + ZTOLZE))
980 GO TO 500
981 DE = (X(I,JC,J2,J3) - XUB(ICOL,JC,J2,J3))/Y(I)
982 IF (DE .GE. DP) GO TO 500
983 IPTYPE = -1
984 APV=Y(I)
985 DP = DE
986 IROWP = I
987 CONTINUE
988 DE = DP + XLB(JCOLP,JC,J2,J3)
989 IF (DE .LT. XUB(JCOLP,JC,J2,J3)) GO TO 600

```

```

987 DP = XUB(JCOLP,JC,J2,J3) - XLB(JCOLP,JC,J2,J3)
988 NPivot = 0
989 RETURN
990 NPivot = 1
991 RETURN
992
993 C INCOMING VARIABLE AT UPPER BOUND
994 C
995 DP = -1.E10
996 KNR=NROW(JC,J2,J3)
997 DO 1500 I=1,KNR
998 IF (I.EQ.10BJ) GO TO 1500
999 ICOL = JH(I,JC,J2,J3)
1000 IF (Y(I) .GT. ZTOLPV) GO TO 1100
1001 IF (Y(I) .LT. -ZTOLPV) GO TO 1200
1002 GO TO 1500
1003 C POSITIVE COEFFICIENT
1004 1100 IF (X(I,JC,J2,J3) .GT. (XUB(ICOL,JC,J2,J3) + ZTOLZE))
1005 GO TO 1500
1006 DE = (X(I,JC,J2,J3) - XUB(ICOL,JC,J2,J3))/Y(I)
1007 IF (DE .LE. DP) GO TO 1500
1008 IPTYPE = -1
1009 GO TO 1250
1010 C NEGATIVE COEFFICIENT
1011 1200 IF (X(I,JC,J2,J3) .LT. (XLB(ICOL,JC,J2,J3) - ZTOLZE))
1012 GO TO 1500
1013 DE = (X(I,JC,J2,J3) - XLB(ICOL,JC,J2,J3))/Y(I)
1014 IF (DE .LE. DP) GO TO 1500
1015 IPTYPE = 0
1016 DP = DE
1017 IROWP = I
1018 APV=Y(I)
1019 CONTINUE
1020 DE = DP + XUB(JCOLP,JC,J2,J3)
1021 IF (DE .GT. XLB(JCOLP,JC,J2,J3)) GO TO 1600
1022 DP = XLB(JCOLP,JC,J2,J3) - XUB(JCOLP,JC,J2,J3)
1023 NPivot = 0
1024 RETURN
1025 NPivot = 1
1026 RETURN
1027 END
1028 C-----
1029 SUBROUTINE WRETA
1030 C
1031 C FORMS NEW ETA-VECTORS FOR PRODUCT FORM OF BASIS INVERSE
1032 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1033 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1034 C
1035 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1036 1 INTEGER*4 (I-N,Q)
1037 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1038 DOUBLE PRECISION E(3000)
1039 REAL A(3000,3,3,1)
1040 C
1041 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLISM,NEGINF,NEMAX,NRMAX,
1042 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QP
1043 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1044 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YFI(350,3,3,1)

```

```

1045      ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
1046      INVFREQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
1047      ),NETA,
1048      NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
1049      602,3,3,1),
1050      LE(1002),IA(3000,3,3,1),IE(3000),
1051      ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
1052
1053      PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
1054      LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1055      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1056      COMMON/BLOCK4/ BND(350),XTOPT,XRHD,YPIBAR(602),YBX(350),IBST(3),
1057      PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1058      NXNF(3),INFLG,NEIND(10),INST(5),MXNST,NST
1059
1060      NELEM = NELEM + 1
1061      IE(NELEM) = IROWP
1062      E(NELEM) = Y(IROWP)
1063
1064      JC=JCUR
1065      J2=JPER(2)
1066      J3=JPER(3)
1067      KNR=NROW(JC,J2,J3)
1068      DO 1000 I = 1,KNR
1069      IF (I.EQ. IROWP) GO TO 1000
1070      IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 1000
1071      NELEM = NELEM + 1
1072      IE(NELEM) = I
1073      E(NELEM) = Y(I)
1074      CONTINUE
1075
1076      NETA = NETA + 1
1077      LE(NETA+1) = NELEM + 1
1078      RETURN
1079      END
1080
1081      SUBROUTINE SHIFTR(IOLD,INEW)
1082
1083      REARRANGES DATA STORAGE
1084      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1085      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1086      ***DESCRIPTION OF PARAMETERS***
1087      IOLD,INEW = PARAMETERS INDEXING STORAGE LOCATIONS IN WHICH
1088      DATA IS TO BE TRANSFERRED (INPUT)
1089
1090      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1091      INTEGER*4 (I-N,Q)
1092      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1093      DOUBLE PRECISION E(3000)
1094      REAL A(3000,3,3,1)
1095
1096      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
1097      NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QP
1098      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1099      22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
1100      ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
1101      INVFREQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
1102      ),NETA,

```

```

1103 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),IA(
1104 3 602,3,3,1),
1105 4 LE(1002),IA(3000,3,3,1),IE(3000),
1106 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
1107
1108 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XIMP(602),
1109 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1110 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1111 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
1112 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1113 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1114
1115 C DIMENSION BARRAY(11900)
1116 EQUIVALENCE (BARRAY(1),B(1,1,1,1))
1117 IF(IOLD.EQ.1) IFO=350*(JCUR-1)+1050*(JPER(2)-1)+3150*(JPER(3)-1)
1118 IF(INEW.EQ.1) IFN=350*(JCUR-1)+1050*(JPER(2)-1)+3150*(JPER(3)-1)
1119 IF(IOLD.EQ.2) IFO=3150+350*(JCUR-1)+1050*(JPER(2)-1)
1120 1 +3150*(JPER(3)-1)
1121 IF(INEW.EQ.2) IFN=3150+350*(JCUR-1)+1050*(JPER(2)-1)
1122 1 +3150*(JPER(3)-1)
1123 IF(IOLD.EQ.3) IFO=6300
1124 IF(INEW.EQ.3) IFN=6300
1125 IF(IOLD.EQ.4) IFO=6650
1126 IF(INEW.EQ.4) IFN=6650
1127
1128 C JC=JCUR
1129 J2=JPER(2)
1130 J3=JPER(3)
1131 KNR=NROW(JC,J2,J3)
1132 DO 1000 I = 1,KNR
1133 BARRAY(IFN + I) = BARRAY(IFO + I)
1134
1135 1000 CONTINUE
1136 RETURN
1137 END
1138 C-----
1139 SUBROUTINE INVERT
1140
1141 C COMPUTES INVERSE OF CURRENT BASIS BY LU DECOMPOSITION
1142 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
1143 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1144
1145 C IMPLICIT REAL*4 (A,C,E,H,O,P,R,W,Z), REAL*8 (B,D,X,Y)
1146 1 INTEGER*4 (I-N,Q)
1147 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1148 DOUBLE PRECISION E(3000)
1149 REAL A(3000,3,3,1)
1150
1151 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTOLCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
1152 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QP
1153 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1154 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YFI(350,3,3,1)
1155 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
1156 2 INVFREQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELEM(3,3,1
1157 2 ),NETA,
1158 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
1159 4 602,3,3,1),
1160 5 LE(1002),IA(3000,3,3,1),IE(3000),
1161 6 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


```

117 COL00008 ROW00018 0.0
118 COL00009 ROW00019 0.0
119 COL00010 ROW00020 0.0
120 COL00011 ROW00020 0.0
121 COL00012 ROW00018 0.0
122 COL00013 ROW00021 0.0
123 COL00014 ROW00023 -.480000 ROW00031 -.490000
124 COL00015 ROW00023 -.480000 ROW00030 1.000000
125 COL00015 ROW00031 -.490000
126 COL00016 ROW00023 -.480000 ROW00031 -.490000
127 COL00017 ROW00016 -.500000 ROW00017 -.500000
128 COL00017 ROW00023 -9.320000 ROW00030 .700000
129 COL00017 ROW00031 -.560000 ROW00032 -1.000000
130 COL00017 ROW00034 1.000000
131 COL00018 ROW00020 1.000000 ROW00031 1.000000
132 COL00019 ROW00021 1.000000 ROW00032 1.000000
133 COL00020 ROW00022 1.000000 ROW00033 1.000000
134 ENDATA
135 1.000
136 ROWS
137 N F0B00001
138 E ROW00016
139 E ROW00017
140 E ROW00018
141 E ROW00019
142 L ROW00020
143 L ROW00021
144 L ROW00022
145 E ROW00023
146 L ROW00024
147 E ROW00025
148 L ROW00026
149 G ROW00027
150 E ROW00028
151 E ROW00029
152 E ROW00030
153 E ROW00031
154 E ROW00032
155 E ROW00033
156 L ROW00034
157 E ROW00035
158 E ROW00036
159 E ROW00037
160 E ROW00038
161 L ROW00039
162 L ROW00040
163 L ROW00041
164 E ROW00042
165 L ROW00043
166 E ROW00044
167 L ROW00045
168 G ROW00046
169 E ROW00047
170 E ROW00048
171 E ROW00049
172 E ROW00050
173 E ROW00051
174 E ROW00052

12001630
12001650
12001660
12001680
12001700
12001710
12001720
12001730
12001750
12001770
12001790

12000220
12000230
12000240
12000250
12000260
12000270
12000280
12000290
12000300
12000310
12000320
12000330
12000340
12000350
12000360
12000370
12000380
12000390
12000400
12000410
12000420
12000430
12000440
12000450
12000460
12000470
12000480
12000490
12000500
12000510
12000520
12000530
12000540
12000550
12000560
12000570
12000580

```


175	L ROW00053						12000590
176	COLUMNS						
177	COL00021	FOB00001	-35.000000	ROW00016	1.000000	12001800	
178	COL00022	FOB00001	54.900000	ROW00016	1.000000	12001810	
179	COL00022	ROW00018	1.000000	ROW00023	-4.000000	12001820	
180	COL00022	ROW00033	-5.500000			12001830	
181	COL00023	FOB00001	54.900000	ROW00017	1.000000	12001840	
182	COL00023	ROW00019	1.000000	ROW00023	-4.000000	12001850	
183	COL00023	ROW00033	-5.500000			12001860	
184	COL00024	FOB00001	-35.000000	ROW00017	1.000000	12001870	
185	COL00025	FOB00001	23.500000	ROW00018	-1.000000	12001880	
186	COL00025	ROW00023	-1.700000	ROW00028	1.000000	12001890	
187	COL00025	ROW00031	-2.450000	ROW00033	-1.400000	12001900	
188	COL00026	FOB00001	23.500000	ROW00019	-1.000000	12001910	
189	COL00026	ROW00023	-1.700000	ROW00029	1.000000	12001920	
190	COL00026	ROW00031	-2.450000	ROW00033	-1.400000	12001930	
191	COL00027	FOB00001	8.720000	ROW00025	1.000000	12001940	
192	COL00027	ROW00031	-2.000000			12001950	
193	COL00028	FOB00001	9.720000	ROW00023	1.500000	12001960	
194	COL00028	ROW00024	1.000000	ROW00025	1.000000	12001970	
195	COL00029	FOB00001	6.740000	ROW00025	1.000000	12001980	
196	COL00029	ROW00027	1.000000	ROW00032	1.200000	12001990	
197	COL00030	FOB00001	6.840000	ROW00025	1.000000	12002000	
198	COL00030	ROW00027	1.000000	ROW00033	1.000000	12002010	
199	COL00031	FOB00001	15.000000	ROW00023	1.000000	12002020	
200	COL00032	FOB00001	22.500000	ROW00033	1.000000	12002030	
201	COL00033	ROW00025	1.000000	ROW00026	1.000000	12002040	
202	COL00034	FOB00001	-50.000000	ROW00028	-1.000000	12002050	
203	COL00034	ROW00042	-48.000000	ROW00050	-49.000000	12002060	
204	COL00035	FOB00001	18.700000	ROW00028	-1.000000	12002070	
205	COL00035	ROW00042	-48.000000	ROW00049	1.000000	12002080	
206	COL00035	ROW00050	-49.000000			12002090	
207	COL00036	FOB00001	-258.300000	ROW00029	-1.000000	12002100	
208	COL00036	ROW00042	-48.000000	ROW00050	-49.000000	12002110	
209	COL00037	FOB00001	-662.000000	ROW00030	-1.000000	12002120	
210	COL00037	ROW00034	-1.000000	ROW00035	-5.000000	12002130	
211	COL00037	ROW00036	-5.000000	ROW00042	-9.320000	12002140	
212	COL00037	ROW00049	-7.000000	ROW00050	-5.600000	12002150	
213	COL00037	ROW00051	-1.000000	ROW00053	1.000000	12002160	
214	COL00038	FOB00001	3.000000	ROW00031	-1.000000	12002170	
215	COL00038	ROW00039	1.000000	ROW00050	1.000000	12002180	
216	COL00039	FOB00001	.390000	ROW00032	-1.000000	12002190	
217	COL00039	ROW00040	1.000000	ROW00051	1.000000	12002200	
218	COL00040	FOB00001	.470000	ROW00033	-1.000000	12002210	
219	COL00040	ROW00041	1.000000	ROW00052	1.000000	12002220	
220	COL00041	FOB00001	-35.000000	ROW00035	1.000000	12002230	
221	COL00042	FOB00001	54.900000	ROW00035	1.000000	12002240	
222	COL00042	ROW00037	1.000000	ROW00042	-4.000000	12002250	
223	COL00042	ROW00052	-5.000000			12002260	
224	COL00043	FOB00001	54.900000	ROW00036	1.000000	12002270	
225	COL00043	ROW00038	1.000000	ROW00042	-4.000000	12002280	
226	COL00043	ROW00052	-5.000000			12002290	
227	COL00044	FOB00001	-35.000000	ROW00036	1.000000	12002300	
228	COL00045	FOB00001	23.500000	ROW00037	-1.000000	12002310	
229	COL00045	ROW00042	-1.700000	ROW00047	1.000000	12002320	
230	COL00045	ROW00050	-2.450000	ROW00052	-1.400000	12002330	
231	COL00046	FOB00001	23.500000	ROW00038	-1.000000	12002340	
232	COL00046	ROW00042	-1.700000	ROW00048	1.000000	12002350	

A3. Listing of solution output for SCAGR7.S2

ITNS	RRES	PIV	IN	OUT	OBJ	CMIN	CMAJ	THET	NINF	NOPT	
175	0	0	0	0	0	0	0	0	0	0	
176	0	0	0	0	0	0	0	0	0	0	
177	0	0	0	0	0	0	0	0	0	0	
178	ITNS	RRES	PIV	IN	OUT	OBJ	CMIN	CMAJ	THET	NINF	NOPT
179	1	0.91E-12	0.15E+01	24	6	-0.20E+05	-0.25E+01	-0.10E+11	0.21E+04	9	14
180	2	0.91E-12	-0.10E+01	22	5	-0.20E+05	-0.28E+01	-0.10E+11	0.0	8	14
181	3	0.91E-12	0.10E+01	19	12	-0.20E+05	-0.45E+01	-0.10E+11	0.0	8	13
182	4	0.91E-12	-0.19E+01	32	16	-0.69E+04	-0.45E+01	-0.10E+11	0.79E+02	8	13
183	5	0.18E-11	0.53E+00	26	3	0.52E+04	-0.34E+01	-0.10E+11	0.15E+03	7	11
184	6	0.18E-11	-0.10E+01	21	4	0.52E+04	-0.42E+01	-0.10E+11	0.0	6	10
185	7	0.18E-11	0.10E+01	18	11	0.52E+04	-0.64E+01	-0.10E+11	0.0	6	9
186	8	0.27E-11	0.10E+01	31	2	-0.14E+05	-0.64E+01	-0.10E+11	0.16E+03	6	10
187	9	0.24E-11	0.10E+01	36	8	-0.43E+05	-0.20E+01	-0.10E+11	0.39E+04	5	7
188	10	0.56E-10	-0.10E+01	33	13	0.21E+06	-0.10E+01	-0.10E+11	0.38E+03	3	3
189	11	0.57E-10	-0.10E+01	34	14	0.21E+06	-0.10E+01	-0.10E+11	0.15E+02	2	2
190	12	0.57E-10	-0.10E+01	35	15	0.20E+06	-0.10E+01	-0.10E+11	0.68E+03	1	1
191	12 ITERATIONS SO FAR										
192											
193											
194											
195											
196											
197											
198											
199											
200											
201											
202											
203											
204											
205											
206											
207											
208											
209											
210											
211											
212											
213											
214											
215											
216											
217											
218											
219											
220											
221											
222											
223											
224											
225											
226											
227											
228											
229											
230											
231											
232											

WE HAVE MADE 1 PASSES
WE ARE AT PROBLEM 1, 1
X VALUES 158.00 4395.63 158.00 158.00 2504.37 62.30 3945.23
1600.00 3595.63 158.00 158.00 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 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 0.0 0.0 0.0 1.00 0.0 0.0 0.0 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 CMAJ 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

233	27	0.36E-11	-0.23E+01	68	38	-0.81E+04	-0.13E+02	-0.10E+11	0.0	12	14		
234	28	0.36E-11	0.50E+00	63	23	-0.81E+04	-0.93E+01	-0.10E+11	0.0	12	16		
235	29	0.36E-11	0.60E+00	60	39	-0.81E+04	-0.19E+02	-0.10E+11	0.0	12	15		
236	30	0.36E-11	-0.10E+01	54	14	-0.81E+04	-0.31E+02	-0.10E+11	0.0	12	15		
237	31	0.36E-11	-0.10E+01	44	4	-0.81E+04	-0.30E+02	-0.10E+11	0.0	12	15		
238	32	0.13E-09	0.10E+01	41	5	0.38E+06	-0.31E+02	-0.10E+11	0.96E+02	12	14		
239	33	0.15E-09	0.10E+02	50	29	0.49E+06	-0.16E+02	-0.10E+11	0.55E+02	12	13		
240	34	0.16E-09	0.98E+00	53	2	0.51E+06	-0.20E+01	-0.10E+11	0.67E+02	11	16		
241	35	0.25E-09	0.10E+01	45	3	0.52E+06	-0.21E+01	-0.10E+11	0.19E+03	10	13		
242	36	0.10E-09	0.83E+00	78	30	0.50E+06	-0.17E+01	-0.10E+11	0.48E+04	9	11		
243	37	0.89E-10	0.20E+00	46	17	0.46E+06	-0.12E+01	-0.10E+11	0.42E+04	7	9		
244	38	0.64E-10	0.50E+01	77	11	0.46E+06	-0.50E+01	-0.10E+11	0.20E+02	6	9		
245	39	0.52E-10	-0.10E+01	58	18	0.46E+06	-0.10E+01	-0.10E+11	0.31E+03	5	4		
246	40	0.51E-10	0.20E+00	48	11	0.46E+06	-0.10E+01	-0.10E+11	0.10E+03	4	4		
247	41	0.17E-09	0.10E+01	43	3	0.45E+06	-0.17E+01	-0.10E+11	0.19E+03	4	7		
248	42	0.27E-09	0.10E+01	40	2	0.43E+06	-0.17E+01	-0.10E+11	0.66E+02	4	6		
249	43	0.19E-09	0.35E+01	53	13	0.45E+06	-0.18E+01	-0.10E+11	0.56E+02	3	6		
250	44	0.13E-09	0.11E+00	59	20	0.39E+06	-0.72E+00	-0.10E+11	0.25E+03	3	5		
251	45	0.12E-09	0.10E+01	79	27	0.38E+06	-0.10E+01	-0.10E+11	0.95E+03	3	4		
252	46	0.24E-09	0.46E+01	20	22	0.44E+06	-0.25E+01	-0.10E+11	0.29E+02	3	4		
253	47	0.22E-09	0.28E+00	66	2	0.45E+06	-0.52E+00	-0.10E+11	0.37E+02	3	3		
254	48	0.17E-09	0.10E+01	45	3	0.46E+06	-0.18E+01	-0.10E+11	0.19E+03	3	2		
255	49	0.24E-09	-0.11E+00	70	16	0.49E+06	-0.11E+00	-0.10E+11	0.25E+03	3	1		
256	INFEAS AT PROB			1	1								
257	MAKING A FEASIBILITY CUT												
258	ITNS	RRES	PIV										
259	50	0.95E-11	0.10E+01	24	10	0.20E+06	-0.20E+01	-0.10E+11	0.36E+04	1	3		
260	51	0.95E-11	0.10E+01	31	11	0.28E+06	-0.52E+03	-0.10E+11	0.16E+03	0	5		
261	52	0.95E-11	0.10E+01	30	9	0.29E+06	-0.93E+01	-0.10E+11	0.16E+04	0	3		
262	53	0.10E-10	0.20E+01	10	17	0.30E+06	-0.20E+01	-0.10E+11	0.12E+04	0	2		
263	54	0.16E-10	0.10E+01	26	10	0.30E+06	-0.21E+01	-0.10E+11	0.79E+03	0	1		
264	55	0.28E-10	0.50E+00	17	16	0.30E+06	-0.51E-01	-0.10E+11	0.31E+04	0	1		
265	55	ITERATIONS	SO FAR										
266													
267													
268	WE HAVE MADE	3	PASSES										
269	WE ARE AT PROBLEM	1,	1										
270	X VALUES												
271	298780.88	158.00	158.00	158.00	158.00	158.00	2504.37	62.30	450.40				
272	1600.00	2345.23	158.00	158.00	158.00	375.20	14.70	3498.27	3112.88				
273	1995.63												
274	JH	1	19	20	22	23	25	26	31	33	34	35	36
275													
276	17	10											
277	KINBAS												
278	1	0	0	-1	-1	0	0	17	0	0	-1	-1	-1
279	-1	16	0	2	3	0	4	5	0	6	10	8	0
280	11	0	12	13	14	15	0						9
281	BS CURRENTLY												
282	0.0	158.00	158.00	158.00	158.00	0.0	0.0	0.0	3092.96	2566.67	6900.00		
283	1600.00	800.00	0.0	0.0	0.0	0.0	-375.20	-92.12	-684.00	-150.00			
284	2400.00												
285	CURRENT Y												
286	1.00	419.52	177.82	177.82	177.82	-474.90	-233.20	-1.67	0.0	0.0	7.21		
287	7.21	0.0	-500.00	-500.00	-258.30	-662.00	3.00	0.39	0.39	0.37			
288	0.0												
289	CURRENT PI												
290	1.00	419.52	177.82	177.82	-474.90	-233.20	-1.67	0.0	0.0	-7.21			

291 7.21 0.0 -500.00 -258.30 -662.00 3.00 0.39 0.37
 292 0.0
 293
 294
 295
 296

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+05	-0.22E+01	-0.10E+11	0.80E+03	5	8
57	0.30E-09	0.10E+01	62	22	0.33E+05	-0.21E+01	-0.10E+11	0.13E+03	5	8
298	0.34E-09	0.10E+01	61	21	0.31E+06	-0.21E+01	-0.10E+11	0.13E+03	5	7
299	0.34E-09	0.10E+01	29	29	0.26E+06	-0.15E+01	-0.10E+11	0.26E+04	5	6
300	0.35E-09	0.10E+01	40	14	0.22E+06	-0.19E+01	-0.10E+11	0.11E+03	4	5
301	0.35E-09	0.37E+01	43	36	0.21E+06	-0.19E+01	-0.10E+11	0.69E+02	4	4
302	0.36E-09	0.29E+01	77	9	0.20E+06	-0.26E+01	-0.10E+11	0.74E+02	4	5
303	0.35E-09	0.71E+00	10	3	0.20E+06	-0.14E+01	-0.10E+11	0.24E+02	4	4
304	0.38E-09	0.20E+00	13	9	0.20E+06	-0.10E+01	-0.10E+11	0.46E+03	4	3
305	0.37E-09	0.12E+01	63	19	0.19E+06	-0.12E+01	-0.10E+11	0.74E+02	4	6
306			1	1						
307										
308										
309										
310	0.35E-10	0.10E+01	25	10	0.29E+06	-0.24E+01	-0.10E+11	0.20E+04	1	3
311	0.43E-10	0.24E+01	10	18	0.30E+06	-0.21E+01	-0.10E+11	0.12E+04	0	1
312	0.49E-10	0.10E+01	38	10	0.30E+06	-0.20E+01	-0.10E+11	0.81E+03	0	1
313										
314										
315										

WE HAVE MADE 5 PASSES

WE ARE AT PROBLEM 1, 1

X VALUES

298698.68	158.00	158.00	158.00	158.00	158.00	158.00	2504.37	62.30	1539.43
1600.00	805.79	158.00					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	0	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	3092.95	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	ITNS	RRES	PIV	IN	OUT	OBJ	CMIN	CMAX	THET	NINF	NOPT
17	0.150E+04										
19	0.26E-09	0.14E+01	53	10	0.26E+06	-0.18E+02	-0.10E+11	0.17E+02	5	8	

NOPT	NINF	THET	CMA	CMIN	CMAX	OBJ	OBUT	OBV	IN	PIV	17 VALUE =	VAR3	VAR2	VAR1
407	1	0	0	-1	0	0	7	0	0	10	0	0	0	0
408	-1	17	0	0	2	3	0	4	5	0	5	0	6	8
409	0	9	19	11	12	13	14	15	16					
BS CURRENTLY														
410	0.0	158.00	158.00	0.0	0.0	3092.96	2566.67	6900.00						
411	1600.00	800.00	0.0	0.0	-375.20	-92.12	-684.00	-150.00						
413	4800.00	2400.00	-0.0											
CURRENT Y														
414	1.00	419.85	178.15	-475.16	-233.46	-1.61	0.0	7.31						
415	7.31	0.0	-500.00	-258.30	-506.39	3.00	0.47	0.47						
417	0.0	0.08	518.61											
CURRENT PI														
418	1.00	419.85	178.15	-475.16	-233.46	-1.61	0.0	-7.31						
419	7.31	0.0	-500.00	-258.30	-506.39	3.00	0.47	0.47						
420	0.0	0.08	518.61											
421	0.0													
422														
423														
424														
425														
426														
LOOKING FOR CUTS TO DELETE														
DELETING ROW 17 VALUE = 0.375E+03														
ITNS	RRES	PIV	IN	OBV	CMIN	CMA	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	KSUPI	OBJT01	CURR BAS							
1	1	1	0.125000	0.555E+06	0.551E+05	0.0	1							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	1	2	0.125000	0.553E+06	0.537E+05	0.693F+05	1*							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	1	2	0.125000	0.555E+06	0.551E+05	0.138E+06	1							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	1	2	0.125000	0.553E+06	0.537E+05	0.208F+06	1							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	2	1	0.125000	0.555E+06	0.551E+05	0.277E+06	1							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	2	1	0.125000	0.553E+06	0.537E+05	0.346E+06	1							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	2	1	0.125000	0.555E+06	0.551E+05	0.415E+06	1							
CHECKING FEAS WITH DCHUZR														
OK - FEAS	2	2	0.125000	0.553E+06	0.537E+05	0.485E+06	1							
Q(X) = -0.554E+06														
THETA = -0.915E+03														

465 MAKING A LOOKAHEAD CUT
 466 ITNS RRES PIV
 467 88 0.73E-10 0.10E+01 18 11 0.83E+06 -0.70E+03 -0.10E+11 0.45E+02 2
 468 89 0.58E-10 0.10E+01 26 16 0.83E+06 -0.75E+01 -0.10E+11 0.88E+03 1
 469 90 0.58E-10 0.83E+00 17 10 0.84E+06 -0.63E+01 -0.10E+11 0.13E+04 1
 470 90 ITERATIONS SO FAR

WE HAVE MADE 11 PASSES
 WE ARE AT PROBLEM 1, 1
 X VALUES
 841177.50 158.00 158.00 158.00 2504.37 62.30 319.60
 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 35 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 0.62E+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 OBJT01 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 DCIUZR
 ENT VAR - WILL RETURN TO 1 1 2
 CHECKING FEAS WITH DCHUZR
 OK - FEAS

CHECKING FEAS WITH DCHUZR
 1 2 1 0.125000 0.621E+06 0.266E+05 0.774E+05 1
 TRYING TO FIND ENTERING IN DCIUZR
 ENT VAR - WILL RETURN TO 1 2 2

471
 472
 473
 474
 475
 476
 477
 478
 479
 480
 481
 482
 483
 484
 485
 486
 487
 488
 489
 490
 491
 492
 493
 494
 495
 496
 497
 498
 499
 500

501
 502
 503
 504
 505
 506
 507
 508
 509
 510
 511
 512
 513
 514
 515
 516
 517
 518
 519
 520
 521
 522

```

523 CHECKING FEAS WITH DCHUZR
524 OK - FEAS
525 1 0.125000 0.619E+06 0.252E+05 0.155E+06 1
526 CHECKING FEAS WITH DCHUZR
527 TRYING TO FIND ENTERING IN DCHUZR
528 ENT VAR - WILL RETURN TO 2
529 CHECKING FEAS WITH DCHUZR
530 OK - FEAS
531 1 0.125000 0.621E+06 0.266E+05 0.232E+06 1
532 CHECKING FEAS WITH DCHUZR
533 TRYING TO FIND ENTERING IN DCHUZR
534 ENT VAR - WILL RETURN TO 2 2
535 RETURNING TO SOLVE NEXT BATCH
536 ITERATING IN DNORML
537 ITNS RRES PIV IN OUT OBJ CMIN MXINF THET NINF NOPT
538 101 0.33E-09 -0.10E+01 68 19 0.62E+06 0.0 0.78E+00 0.78E+00 1 0
539 VAR1 VAR2 VAR3 PROB OBJ KSIPI OBJTOT CURR BAS
540 1 1 2 0.125000 0.619E+06 0.252E+05 0.310E+06 2
541 CHECKING FEAS WITH DCHUZR
542 OK - FEAS
543 1 2 0.125000 0.621E+06 0.266E+05 0.388E+06 2
544 CHECKING FEAS WITH DCHUZR
545 OK - FEAS
546 1 2 0.125000 0.619E+06 0.252E+05 0.465E+06 2
547 CHECKING FEAS WITH DCHUZR
548 OK - FEAS
549 1 2 0.125000 0.621E+06 0.266E+05 0.543E+06 2
550 Q(X) = -0.620E+06
551 THETA = -0.629E+06
552 LOWER BOUND = -0.841E+06
553
554 UPPER BOUND = -0.833E+06
555 MAKING A LOOKAHEAD CUT
556 ITNS RRES PIV IN OUT OBJ CMIN CMAX THET NINF NOPT
557 102 0.73E-10 0.12E+01 10 17 0.83E+06 -0.77E+01 -0.10E+11 0.11E+04 1 3
558 103 0.87E-10 0.64E+01 17 20 0.83E+06 -0.63E+01 -0.10E+11 0.31E+02 0 1
559 103 ITERATIONS SO FAR
560
561
562 WE HAVE MADE 13 PASSES
563 WE ARE AT PROBLEM 1, 1
564 X VALUES
565 832935.41 158.00 158.00 158.00 158.00 2504.37 62.30 903.38
566 1600.00 1441.85 158.00 158.00 375.20 195.38 2414.21 450.40
567 1092.25 45.44 618017.08 31.23
568 JH
569 1 22 23 25 26 28 7 27 33 29 35 36 37 38 39
570 30 10 18 41 17
571 KINBAS
572 1 0 0 -1 -1 0 7 0 0 17 0 0 -1 -1 -1
573 -1 20 18 0 0 0 2 3 0 4 5 8 6 10 16
574 0 0 9 0 11 12 13 14 15 0 19
575 BS CURRENTLY
576 0.0 158.00 158.00 0.0 0.0 3092.96 2566.67 6900.00
577 1600.00 800.00 0.0 0.0 -375.20 -92.12 -584.00 -150.00
578 2400.00 -0.0 54420.09 25898.61
579 CURRENT Y
580 1.00 1141.21 169.38 -1195.88 -224.05 -1.05 0.0 -8.14

```

581	8.14	0.00	-1220.05	-248.22	-1723.39	-2.88	1.17	1.30
582	-0.00	0.0	0.02	0.98				
583	CURRENT PI							
584	1.00	1141.21	169.38	-1195.88	-224.05	-1.05	0.0	-8.14
585	8.14	0.00	-1220.05	-248.22	-1723.39	-2.88	1.17	1.30
586	-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

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

INFEAS AT PROB 1

MAKING A FEASIBILITY CUT

ITNS	RRES	PIV	IN	OUT	OBJ	CMIN	CMAX	THET	NINF	NOPT
601	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

MAKING A FEASIBILITY CUT

LOOKING FOR CUTS TO DELETE

DELETING ROW 17 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

MAKING A FEASIBILITY CUT

WE HAVE MADE 15 PASSES
WE ARE AT PROBLEM 1, 1

X VALUES

832935.32	158.00	158.00	158.00	158.00	158.00	2504.37	62.30	903.38
1600.00	1092.25	158.00	158.00	158.00	375.20	195.38	2400.01	11.85
45.43	618018.26	1430.01						

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	0	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	158.00	158.00	158.00	0.0	0.0	3092.96	2566.67	6900.00
1600.00	800.00	800.00	800.00	800.00	800.00	0.0	0.0	-92.12	-684.00	-150.00
2400.00	54420.09	25898.61	2400.00	2400.00	2400.00					

CURRENT Y

1.00	1141.24	169.40	169.40	169.40	169.40	-1195.90	-224.07	-224.07	-224.07	-1.05	0.0	0.0	-8.15
8.15	0.0	-1220.06	-1220.06	-1220.06	-1220.06	-248.22	-1723.41	-1723.41	-1723.41	-2.87	1.17	1.17	1.31
0.0	0.02	0.98	0.98	0.98	0.98	0.00	0.00	0.00	0.00				

CURRENT PI

1.00	1141.24	169.40	169.40	169.40	169.40	-1195.90	-224.07	-224.07	-224.07	-1.05	0.0	0.0	-8.15
8.15	0.0	-1220.06	-1220.06	-1220.06	-1220.06	-248.22	-1723.41	-1723.41	-1723.41	-2.87	1.17	1.17	1.31
0.0	0.02	0.98	0.98	0.98	0.98	0.00	0.00	0.00	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
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

MAKING A FEASIBILITY CUT

639	ITNS	RRES	PIV	IN	OUT	OBJ	CMIN	CMAx	THET	NINF	NOPT
640	108	0.40E-01	0.83E+00	19	18	0.83E+06	-0.10E+01	-0.10E+11	0.17E+04	1	2
641	ITNS	RRES	PIV	IN	OUT	OBJ	CMIN	CMAx	THET	NINF	NOPT
642	109	0.73E-10	0.12E+01	29	19	0.83E+06	-0.20E-02	-0.10E+11	0.14E+04	0	1
643	109 ITERATIONS SO FAR										
644											

WE HAVE MADE 17 PASSES

WE ARE AT PROBLEM 1.

X VALUES	832935.39	158.00	158.00	158.00	158.00	158.00	158.00	2504.37	62.30	462.25
	1600.00	1092.25	158.00	158.00	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	0	-1	-1	-1
-1	0	0	0	20	0	2	3	0	4	5	18	6	19	8	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	-584.00	-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	0.0	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	0.0	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	KSUPI	OBJT01	CURR BAS			
1	1	1	0.125000	0.617E+06	0.252E+05	0.0				

CHECKING FEAS WITH DCHUZR

TRYING TO FIND ENTERING IN DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

TRYING TO FIND ENTERING IN DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

TRYING TO FIND ENTERING IN DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

ENT VAR - WILL RETURN TO 1

CHECKING FEAS WITH DCHUZR

ENT VAR - WILL RETURN TO 1

```

697 CHECKING FEAS WITH DCHUZR
698 TRYING TO FIND ENTERING IN DCHUZR
699 ENT VAR - WILL RETURN TO 2 1
700 CHECKING FEAS WITH DCHUZR
701 TRYING TO FIND ENTERING IN DCHUZR
702 ENT VAR - WILL RETURN TO 2 2
703 RETURNING TO SOLVE NEXT BATCH
704 ITERATING IN DNORML
705 ITNS RRES PIV IN OUT OBJ CMIN
706 115 0.35E-09 -0.10E+01 31 13 0.62E+06 0.0
707 116 0.35E-09 -0.10E+01 68 36 0.62E+06 0.0
708 VAR1 VAR2 VAR3 PROB OBJ KSIPI
709 1 1 2 0.125000 0.617E+06 0.252E+05
710 CHECKING FEAS WITH DCHUZR
711 TRYING TO FIND ENTERING IN DCHUZR
712 ENT VAR - WILL RETURN TO 1 2 1
713 CHECKING FEAS WITH DCHUZR
714 OK - FEAS
715 1 2 2 0.125000 0.617E+06 0.266E+05
716 CHECKING FEAS WITH DCHUZR
717 OK - FEAS
718 2 1 2 0.125000 0.617E+06 0.252E+05
719 CHECKING FEAS WITH DCHUZR
720 TRYING TO FIND ENTERING IN DCHUZR
721 ENT VAR - WILL RETURN TO 2 1
722 CHECKING FEAS WITH DCHUZR
723 OK - FEAS
724 2 2 2 0.125000 0.617E+06 0.266E+05
725 RETURNING TO SOLVE NEXT BATCH
726 ITERATING IN DNORML
727 ITNS RRES PIV IN OUT OBJ CMIN
728 117 0.36E-09 -0.10E+01 59 36 0.62E+06 0.0
729 118 0.35E-09 -0.10E+01 13 18 0.62E+06 0.0
730 VAR1 VAR2 VAR3 PROB OBJ KSIPI
731 1 2 1 0.125000 0.618E+06 0.266E+05
732 CHECKING FEAS WITH DCHUZR
733 OK - FEAS
734 2 1 0.125000 0.618E+06 0.266E+05
735 Q(X) = -0.617E+06
736 THETA = -0.618E+06
737 LOWER BOUND = -0.833E+06
738 UPPER BOUND = -0.832E+06
739
740
741 TOTAL SOLUTION TIME = 1658 MILLISECONDS
742 SOLUTION TIME WITHOUT INPUT = 1374 MILLISECONDS
743
744
745 LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 1 1 1
746
747 ROW VALUE
748 F0B00001 617323.7260
749 ROW00016 187.6000
750 ROW00017 187.6000
751 ROW00018 187.6000
752 ROW00019 187.6000
753 ROW00020 1404.6237
754 ROW00021 0.0000

```


755	ROW00022	1188.1548
756	ROW00023	586.4588
757	ROW00024	2566.6700
758	ROW00025	1786.4842
759	ROW00026	1600.0000
760	ROW00027	146.8458
761	ROW00028	187.6000
762	ROW00029	187.6000
763	ROW00030	420.6400
764	ROW00031	95.7972
765	ROW00032	46.5954
766	ROW00033	900.2505
767	ROW00034	45.4400
768	ROW00035	319.2856
769	ROW00036	210.3200
770	ROW00037	210.3200
771	ROW00038	210.3200
772	ROW00039	1704.2028
773	ROW00040	2080.7144
774	ROW00041	1000.7842
775	ROW00042	1133.7995
776	ROW00043	2566.6700
777	ROW00044	2133.3300
778	ROW00045	1600.0000
779	ROW00046	210.3200
780	ROW00047	210.3200
781	ROW00048	210.3200
782	ROW00049	482.0480
783	ROW00050	199.2158
784	ROW00051	1660.0744
785	ROW00052	600.0000
786	ROW00053	61.4080

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 1 1 2

ROW	VALUE
791	616921.6054
792	187.6000
793	187.6000
794	187.6000
795	187.6000
796	187.6000
797	1404.6237
798	0.0000
799	1188.1548
800	586.4588
801	2566.6700
802	1933.3300
803	1600.0000
804	53.1542
805	187.6000
806	187.6000
807	420.6400
808	125.1663
809	98.9654
810	701.0346
811	45.4400
812	256.4416

813	ROW00036	210.3200
814	ROW00037	210.3200
815	ROW00038	210.3200
816	ROW00039	1674.8337
817	ROW00040	2143.5584
818	ROW00041	1200.0000
819	ROW00042	1133.7995
820	ROW00043	2566.6700
821	ROW00044	1986.4842
822	ROW00045	1546.8458
823	ROW00046	210.3200
824	ROW00047	210.3200
825	ROW00048	210.3200
826	ROW00049	482.0480
827	ROW00050	0.7842
828	ROW00051	1723.8594
829	ROW00052	799.2158
830	ROW00053	61.4080
831		
832		
833		
834		

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 1 2 1

835	ROW	VALUE
836	FOE00001	617653.4347
837	ROW00016	187.6000
838	ROW00017	187.6000
839	ROW00018	187.6000
840	ROW00019	187.6000
841	ROW00020	1404.6237
842	ROW00021	0.0000
843	ROW00022	1188.1548
844	ROW00023	586.4588
845	ROW00024	2566.6700
846	ROW00025	1833.0795
847	ROW00026	1600.0000
848	ROW00027	153.4046
849	ROW00028	187.6000
850	ROW00029	187.6000
851	ROW00030	420.6400
852	ROW00031	105.1162
853	ROW00032	100.2505
854	ROW00033	900.2505
855	ROW00034	45.4400
856	ROW00035	375.2000
857	ROW00036	210.3200
858	ROW00037	210.3200
859	ROW00038	210.3200
860	ROW00039	1694.8838
861	ROW00040	2024.8000
862	ROW00041	1000.7842
863	ROW00042	1133.7995
864	ROW00043	2566.6700
865	ROW00044	2086.7346
866	ROW00045	1646.5954
867	ROW00046	210.3200
868	ROW00047	210.3200
869	ROW00048	210.3200
870	ROW00049	482.0480

871	ROW00050	199.2158
872	ROW00051	1604.1600
873	ROW00052	600.0000
874	ROW00053	61.4080
875		
876		

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 1 2 2

877				
878				
879				
880	ROW00001	616921.6054		
881	ROW00016	187.6000		
882	ROW00017	187.6000		
883	ROW00018	187.6000		
884	ROW00019	187.6000		
885	ROW00020	1404.6237		
886	ROW00021	0.0000		
887	ROW00022	1188.1548		
888	ROW00023	586.4588		
889	ROW00024	2566.6700		
890	ROW00025	1933.3300		
891	ROW00026	1600.0000		
892	ROW00027	253.1542		
893	ROW00028	187.6000		
894	ROW00029	187.6000		
895	ROW00030	420.6400		
896	ROW00031	125.1663		
897	ROW00032	98.9654		
898	ROW00033	701.0346		
899	ROW00034	45.4400		
900	ROW00035	256.4416		
901	ROW00036	210.3200		
902	ROW00037	210.3200		
903	ROW00038	210.3200		
904	ROW00039	1674.8337		
905	ROW00040	2143.5584		
906	ROW00041	1200.0000		
907	ROW00042	1133.7995		
908	ROW00043	2566.6700		
909	ROW00044	1986.4842		
910	ROW00045	1546.8458		
911	ROW00046	210.3200		
912	ROW00047	210.3200		
913	ROW00048	210.3200		
914	ROW00049	482.0480		
915	ROW00050	0.7842		
916	ROW00051	1723.8594		
917	ROW00052	799.2158		
918	ROW00053	61.4080		
919				
920				

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 2 1 1

921				
922				
923	ROW00001	617323.7260		
924	ROW00016	187.6000		
925	ROW00017	187.6000		
926	ROW00018	187.6000		
927	ROW00019	187.6000		
928				

929	R0W00020	1604.6237
930	R0W00021	0.0000
931	R0W00022	1188.1548
932	R0W00023	586.4588
933	R0W00024	2566.6700
934	R0W00025	1786.4842
935	R0W00026	1600.0000
936	R0W00027	146.8458
937	R0W00028	187.6000
938	R0W00029	187.6000
939	R0W00030	420.6400
940	R0W00031	95.7972
941	R0W00032	46.5954
942	R0W00033	900.2505
943	R0W00034	45.4400
944	R0W00035	319.2856
945	R0W00036	210.3200
946	R0W00037	210.3200
947	R0W00038	210.3200
948	R0W00039	1704.2028
949	R0W00040	2080.7144
950	R0W00041	1000.7842
951	R0W00042	1133.7995
952	R0W00043	2566.6700
953	R0W00044	2133.3300
954	R0W00045	1600.0000
955	R0W00046	210.3200
956	R0W00047	210.3200
957	R0W00048	210.3200
958	R0W00049	482.0480
959	R0W00050	199.2158
960	R0W00051	1660.0744
961	R0W00052	600.0000
962	R0W00053	61.4080

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 2 1 2

ROW	VALUE
F0B00001	616921.6054
R0W00016	187.6000
R0W00017	187.6000
R0W00018	187.6000
R0W00019	187.6000
R0W00020	1604.6237
R0W00021	0.0000
R0W00022	1188.1548
R0W00023	586.4588
R0W00024	2566.6700
R0W00025	1933.3300
R0W00026	1600.0000
R0W00027	53.1542
R0W00028	187.6000
R0W00029	187.6000
R0W00030	420.6400
R0W00031	125.1663
R0W00032	98.9654
R0W00033	701.0346

987	ROW00034	45.4400
988	ROW00035	256.4416
989	ROW00036	210.3200
990	ROW00037	210.3200
991	ROW00038	210.3200
992	ROW00039	1674.8337
993	ROW00040	2143.5584
994	ROW00041	1200.0000
995	ROW00042	1133.7995
996	ROW00043	2566.6700
997	ROW00044	1986.4842
998	ROW00045	1546.8458
999	ROW00046	210.3200
1000	ROW00047	210.3200
1001	ROW00048	210.3200
1002	ROW00049	482.0480
1003	ROW00050	0.7842
1004	ROW00051	1723.8594
1005	ROW00052	799.2158
1006	ROW00053	61.4080

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 2 2 1

	ROW	VALUE
1011	FOB00001	617653.4347
1012	ROW00016	187.6000
1013	ROW00017	187.6000
1014	ROW00018	187.6000
1015	ROW00019	187.6000
1016	ROW00020	1604.6237
1017	ROW00021	0.0000
1018	ROW00022	1188.1548
1019	ROW00023	586.4588
1020	ROW00024	2566.6700
1021	ROW00025	1833.0795
1022	ROW00026	1600.0000
1023	ROW00027	153.4046
1024	ROW00028	187.6000
1025	ROW00029	187.6000
1026	ROW00030	420.6400
1027	ROW00031	105.1162
1028	ROW00032	100.2505
1029	ROW00033	900.2505
1030	ROW00034	45.4400
1031	ROW00035	375.2000
1032	ROW00036	210.3200
1033	ROW00037	210.3200
1034	ROW00038	210.3200
1035	ROW00039	1694.8838
1036	ROW00040	2024.8000
1037	ROW00041	1000.7842
1038	ROW00042	1133.7995
1039	ROW00043	2566.6700
1040	ROW00044	2086.7346
1041	ROW00045	1646.5954
1042	ROW00046	210.3200
1043	ROW00047	210.3200

1045	ROW00048	210.3200
1046	ROW00049	482.0480
1047	ROW00050	199.2158
1048	ROW00051	1604.1600
1049	ROW00052	600.0000
1050	ROW00053	61.4080

LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO 2 ? ? 2

1051	ROW	VALUE
1052	ROW00001	616921.6054
1053	ROW00016	187.6000
1054	ROW00017	187.6000
1055	ROW00018	187.6000
1056	ROW00019	187.6000
1057	ROW00020	1604.6237
1058	ROW00021	0.0000
1059	ROW00022	1188.1548
1060	ROW00023	586.4588
1061	ROW00024	2566.6700
1062	ROW00025	1933.3300
1063	ROW00026	1600.0000
1064	ROW00027	253.1542
1065	ROW00028	187.6000
1066	ROW00029	187.6000
1067	ROW00030	420.6400
1068	ROW00031	125.1663
1069	ROW00032	98.9654
1070	ROW00033	701.0346
1071	ROW00034	45.4400
1072	ROW00035	256.4416
1073	ROW00036	210.3200
1074	ROW00037	210.3200
1075	ROW00038	210.3200
1076	ROW00039	1674.8337
1077	ROW00040	2143.5584
1078	ROW00041	1200.0000
1079	ROW00042	1133.7995
1080	ROW00043	2566.6700
1081	ROW00044	1986.4842
1082	ROW00045	1546.8458
1083	ROW00046	210.3200
1084	ROW00047	210.3200
1085	ROW00048	210.3200
1086	ROW00049	482.0480
1087	ROW00050	0.7842
1088	ROW00051	1723.8594
1089	ROW00052	799.2158
1090	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
FOB00001	1	832935.3907
ROW00001	0	0.0
ROW00002	0	0.0
ROW00003	-1	0.0
ROW00004	-1	0.0
ROW00005	0	0.0
ROW00006	7	62.2967
ROW00007	0	0.0
ROW00008	0	0.0
ROW00009	10	1092.2453
ROW00010	0	0.0
ROW00011	0	0.0
ROW00012	-1	0.0
ROW00013	-1	0.0
ROW00014	-1	0.0
ROW00015	-1	0.0
COL00001	0	0.0
COL00002	2	158.0000
COL00003	3	158.0000
COL00004	0	0.0
COL00005	4	158.0000
COL00006	5	158.0000
COL00007	18	903.3814
COL00008	6	2504.3733
COL00009	19	1430.0002
COL00010	8	462.2451
COL00011	0	0.0
COL00012	0	0.0
COL00013	9	1600.0000
COL00014	0	0.0
COL00015	11	158.0000
COL00016	12	158.0000
COL00017	13	375.2000
COL00018	14	195.3763
COL00019	15	2400.0000
COL00020	16	11.8452

OPTIMAL DUAL VALUES		
1.00	1141.23	169.40
-1.95	90	-224.07
8.15	0.0	-248.22
0.02	0.98	0.00
		0.0
		-1.05
		-2.87
		0.0
		1.17
		-8.15
		1.31

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
FOB00001	1	617627.4347
ROW00016	0	0.0
ROW00017	0	0.0
ROW00018	-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

1		
2	FOBOOOO1	FOBOOOO1
3	ROWOOO06	ROWOOO06
4	ROWOOO09	ROWOOO09
5	COLOOO02	ROWOOO01
6	COLOOO03	ROWOOO02
7	COLOOO05	ROWOOO03
8	COLOOO06	ROWOOO04
9	COLOOO08	ROWOOO05
10	COLOOO10	ROWOOO07
11	COLOOO13	ROWOOO08
12	COLOOO15	ROWOOO10
13	COLOOO16	ROWOOO11
14	COLOOO17	ROWOOO12
15	COLOOO18	ROWOOO13
16	COLOOO19	ROWOOO14
17	COLOOO20	ROWOOO15
18		
19	FOBOOOO1	FOBOOOO1
20	ROWOOO20	ROWOOO20
21	ROWOOO21	ROWOOO21
22	ROWOOO22	ROWOOO22
23	ROWOOO27	ROWOOO27
24	ROWOOO34	ROWOOO34
25	ROWOOO39	ROWOOO39
26	ROWOOO40	ROWOOO40
27	ROWOOO41	ROWOOO41
28	ROWOOO45	ROWOOO45
29	ROWOOO53	ROWOOO53
30	COLOOO22	ROWOOO16
31	COLOOO23	ROWOOO17
32	COLOOO25	ROWOOO18
33	COLOOO26	ROWOOO19
34	COLOOO27	ROWOOO25
35	COLOOO28	ROWOOO24
36	COLOOO30	ROWOOO33
37	COLOOO31	ROWOOO23
38	COLOOO33	ROWOOO26
39	COLOOO35	ROWOOO28
40	COLOOO36	ROWOOO29
41	COLOOO37	ROWOOO30
42	COLOOO38	ROWOOO31
43	COLOOO39	ROWOOO32
44	COLOOO40	ROWOOO52
45	COLOOO42	ROWOOO35
46	COLOOO43	ROWOOO36
47	COLOOO45	ROWOOO37
48	COLOOO46	ROWOOO38
49	COLOOO47	ROWOOO50
50	COLOOO48	ROWOOO43
51	COLOOO50	ROWOOO46
52	COLOOO51	ROWOOO42
53	COLOOO53	ROWOOO44
54	COLOOO54	ROWOOO47
55	COLOOO56	ROWOOO48
56	COLOOO57	ROWOOO49
57	COLOOO59	ROWOOO51

```

1161
1162 6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
1163 7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1164 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1165 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
1166 1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1167 2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1168 C
1169 INTEGER MREG,HREG,VREG
1170 DIMENSION MREG(350),HREG(350),VREG(350)
1171 EQUIVALENCE (MREG(1),YTEMP(1)),(HREG(1),YTEMP(301))
1172 1 , (VREG(1),YTEMP1(6))
1173 C
1174 C SET PARAMETERS
1175 C
1176 NETA = 0
1177 NLETA = 0
1178 NUETA = 0
1179 NELEM = 0
1180 NLELEM = 0
1181 NUELEM = 0
1182 NABOVE = 0
1183 LE(1) = 1
1184 LR1 = 1
1185 KR1 = 0
1186 JC=JCUR
1187 J2=JPER(2)
1188 J3=JPER(3)
1189 LR4 = NROW(JC,J2,J3) + 1
1190 C
1191 C PUT SLACKS AND ARTIFICIALS IN PART 4 AND REST IN PART 1
1192 C
1193 KNR=NROW(JC,J2,J3)
1194 DO 100 I = 1,KNR
1195 IF (JH(I,JC,J2,J3) .GT. NROW(JC,J2,J3)) GO TO 50
1196 LR4 = LR4 - 1
1197 MREG(LR4) = JH(I,JC,J2,J3)
1198 VREG(LR4) = JH(I,JC,J2,J3)
1199 GO TO 90
1200 KR1 = KR1 + 1
1201 VREG(KR1) = JH(I,JC,J2,J3)
1202 HREG(I) = -1
1203 JH(I,JC,J2,J3) = 0
1204 CONTINUE
1205 C
1206 KR3 = LR4 - 1
1207 LR3 = LR4
1208 C
1209 KNR=NROW(JC,J2,J3)
1210 DO 200 I = LR4,KNR
1211 IR = MREG(I)
1212 HREG(IR) = 0
1213 JH(IR,JC,J2,J3) = IR
1214 KINBAS(IR,JC,J2,J3) = IR
1215 CONTINUE
1216 C
1217 C PULL OUT VECTORS BELOW BUMP AND GET ROW COUNTS
1218 C
200

```

```

1219 NBNONZ = NROW(JC,J2,J3) - LR4 + 1
1220 IF (KR1 .EQ. O) GO TO 1190
1221 J = LR1
1222 IV = VREG(J)
1223 LL = LA(IV,JC,J2,J3)
1224 KK = LA(IV+1,JC,J2,J3) - 1
1225 IRCNT = O
1226 DO 220 I = LL, KK
1227 NBNONZ = NBNONZ + 1
1228 IR = IA(I,JC,J2,J3)
1229 IF (HREG(IR) .GE. O) GO TO 220
1230 IRCNT = IRCNT + 1
1231 HREG(IR) = HREG(IR) - 1
1232 IRP = IR
1233 CONTINUE
1234 IF (IRCNT - 1) 230,250,300
1235 WRITE(6,8000)
1236 FORMAT(16HMATRIX SINGULAR )
1237 KINBAS(IV,JC,J2,J3) = O
1238 VREG(J) = VREG(KR1)
1239 KR1 = KR1 - 1
1240 IF (J .GT. KR1) GO TO 310
1241 GO TO 210
1242 C
1243 250 VREG(J) = VREG(KR1)
1244 KR1 = KR1 - 1
1245 LR3 = LR3 - 1
1246 VREG(LR3) = IV
1247 MREG(LR3) = IRP
1248 HREG(IRP) = O
1249 JH(IRP,JC,J2,J3) = IV
1250 KINBAS(IV,JC,J2,J3) = IRP
1251 IF (J .GT. KR1) GO TO 310
1252 GO TO 210
1253 IF (J .GE. KR1) GO TO 310
1254 J = J+1
1255 GO TO 210
1256 C
1257 C PULL OUT REMAINING VECTORS ABOVE AND BELOW THE
1258 C BUMP AND ESTABLISH MERIT COUNTS OF COLUMNS
1259 C
1260 NVREM = O
1261 IF(KR1 .EQ. O) GO TO 1190
1262 J = LR1
1263 IV = VREG(J)
1264 LL = LA(IV,JC,J2,J3)
1265 KK = LA(IV+1,JC,J2,J3) - 1
1266 IRCNT = O
1267 DO 800 I = LL, KK
1268 IR = IA(I,JC,J2,J3)
1269 IF (HREG(IR) .NE. -2) GO TO 400
1270 C
1271 C PIVOT ABOVE BUMP (PART OF L)
1272 C
1273 NABOVE = NABOVE + 1
1274 IROWP = IR
1275 CALL UNPACK(IV)
1276 CALL WRETA

```

```

1277 NLETA = NETA
1278 JH(IR,JC,J2,J3) = IV
1279 KINBAS(IV,JC,J2,J3) = IR
1280 VREG(J) = VREG(KR1)
1281 KR1 = KR1 - 1
1282 NVREM = NVREM + 1
1283 HREG(IR) = IV
1284 GO TO 940
1285
1286 IF (HREG(IR) .GE. O) GO TO 800
1287 IRCNT = IRCNT + 1
1288 IRP = IR
1289 CONTINUE
1290
1291 IF (IRCNT - 1) 810,900,1000
1292 WRITE(6,8000)
1293 KINBAS(IV,JC,J2,J3) = O
1294 VREG(J) = VREG(KR1)
1295 NVREM = NVREM + 1
1296 KR1 = KR1 - 1
1297 IF (J .GT. KR1) GO TO 1010
1298 GO TO 320
1299
1300 C PUT VECTOR BELOW BUMP
1301
1302 C
1303 C
1304 C
1305 C
1306 C
1307 C
1308 C
1309 C
1310 C
1311 C
1312 C
1313 C
1314 C
1315 C
1316 C
1317 C
1318 C
1319 C
1320 C
1321 C
1322 C
1323 C
1324 C
1325 C
1326 C
1327 C
1328 C
1329 C
1330 C
1331 C
1332 C
1333 C
1334 C

```

```

940 DO 950 I1 = LL, KK
    IIR = IA(II,JC,J2,J3)
    IF (HREG(IIR) .GE. O) GO TO 950
    HREG(IIR) = HREG(IIR) + 1
    CONTINUE
950
1000 IF (J .GT. KR1) GO TO 1010
    GO TO 320
    IF (J .GE. KR1) GO TO 1010
    J = J+1
    GO TO 320
1010 IF(NVREM .GT. O) GO TO 310
C
C GET MERIT COUNTS
C
1020 IF (KR1 .EQ. O) 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 = O
        DO 1050 I = LL, KK

```

```

1335      IR = IA(I,JC,J2,J3)
1336      IF (HREG(IR) .GE. O) GO TO 1050
1337      IMCNT = IMCNT - (HREG(IR) + 1)
1338      CONTINUE
1339      MREG(J) = IMCNT
1340      CONTINUE
1341      C
1342      C SORT COLUMNS INTO MERIT ORDER USING SHELL SORT
1343      C
1344      ISD = 1
1345      IF (KR1 .LT. 2*ISD) GO TO 1108
1346      ISD = 2*ISD
1347      GO TO 1106
1348      ISD = ISD - 1
1349      C END OF INITIALIZATION
1350      IF (ISD .LE. O) GO TO 1107
1351      ISK = 1
1352      ISJ = ISK
1353      ISL = ISK + ISD
1354      ISY = MREG(ISL)
1355      ISZ = VREG(ISL)
1356      IF (ISY .LT. MREG(ISJ)) GO TO 1104
1357      ISL = ISJ + ISD
1358      MREG(ISL) = ISY
1359      VREG(ISL) = ISZ
1360      ISK = ISK + 1
1361      IF ((ISK + ISD) .LE. KR1) GO TO 1102
1362      ISD = (ISD - 1) / 2
1363      GO TO 1101
1364      ISL = ISJ + ISD
1365      MREG(ISL) = MREG(ISJ)
1366      VREG(ISL) = VREG(ISJ)
1367      ISJ = ISJ - ISD
1368      IF (ISJ .GT. O) GO TO 1103
1369      GO TO 1105
1370      CONTINUE
1371      C
1372      C END OF SORT ROUTINE
1373      C PUT OUT BELOW BUMP ETAS (PART OF U)
1374      C
1375      1190 NSLCK = O
1376      NBELOW = O
1377      NELAST = NEMAX
1378      NTLAST = NTMAX
1379      LE(NTLAST + 1) = NELAST + 1
1380      C
1381      LR = LR3
1382      IF (LR3 .GE. LR4) LR = LR4
1383      IF (LR .GT. NROW(JC,J2,J3)) GO TO 2050
1384      JK = NROW(JC,J2,J3) + 1
1385      KNR=NROW(JC,J2,J3)
1386      DO 2000 JJ= LR,KNR
1387         JK = JK - 1
1388         IV = VREG(JK)
1389         I = MREG(JK)
1390         NBELOW = NBELOW + 1
1391         IF (IV .GT. NROW(JC,J2,J3)) GO TO 1200
1392         NSLCK = NSLCK + 1

```

```

1393 LL = LA(IV,JC,J2,J3)
1394 KK = LA(IV+1,JC,J2,J3) -1
1395 IF (KK .GT. LL) GO TO 1300
1396 IF (ABS(A(LL,JC,J2,J3) - 1.) .LE. ZTOLZE) GO TO 2000
1397 C
1398 NUETA = NUETA + 1
1399 DO 1400 J = LL, KK
1400 IR = IA(J,JC,J2,J3)
1401 IF (IR .EQ. I) GO TO 1390
1402 IE(NELAST) = IR
1403 E(NELAST) = A(J,JC,J2,J3)
1404 NELAST = NELAST - 1
1405 NUELEM = NUELEM + 1
1406 GO TO 1400
1407 EP = A(J,JC,J2,J3)
1408 CONTINUE
1409 IE(NELAST) = I
1410 E(NELAST) = EP
1411 LE(NLAST) = NELAST
1412 NELAST = NELAST - 1
1413 NTLAST = NTLAST - 1
1414 NUELEM = NUELEM + 1
1415 CONTINUE
1416 2000 IF(KR1 .EQ. O) GO TO 3500
1417 C
1418 DO L-U DECOMPOSITION OF BUMP
1419 C
1420 DO 3000 J = LR1, KR1
1421 IV = VREG(J)
1422 CALL UNPACK(IV)
1423 CALL FTRAN(2)
1424 IROWP = O
1425 IRCMIN = -999999
1426 KNR=NROW(JC,J2,J3)
1427 DO 2100 I = 1,KNR
1428 IF (DABS(Y(I)) .LE. ZTOLPV) GO TO 2100
1429 IF (HREG(I) .GE. O) GO TO 2100
1430 IF (HREG(I) .LE. IRCMIN) GO TO 2100
1431 IRCMIN = HREG(I)
1432 IROWP = I
1433 CONTINUE
1434 IF (IROWP .GT. O) GO TO 2150
1435 WRITE(6,8000)
1436 KINBAS(IV,JC,J2,J3) = O
1437 GO TO 3000
1438 C
1439 2150 INCR = HREG(IROWP) + 3
1440 C
1441 WRITE L AND U ETAS
1442 C
1443 IF (J .EQ. KR1) GO TO 2160
1444 NELEM = NELEM + 1
1445 IE(NELEM) = IROWP
1446 E(NELEM) = Y(IROWP)
1447 KNR=NROW(JC,J2,J3)
1448 DO 2300 I = 1,KNR
1449 IF (I .EQ. IROWP) GO TO 2300
1450 IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2300

```

```

1451 IF (HREG(I) .GE. O) GO TO 2200
1452
1453 C L ETA ELEMENTS
1454
1455 NELEM = NELEM + 1
1456 IE(NELEM) = I
1457 E(NELEM) = Y(I)
1458 GO TO 2300
1459
1460 C U ETA ELEMENTS
1461
1462 IE(NELAST) = I
1463 E(NELAST) = Y(I)
1464 NELAST = NELAST - 1
1465 NUELEM = NUELEM + 1
1466 CONTINUE
1467
1468 JH(IROWP,JC,J2,J3) = IV
1469 KINBAS(IV,JC,J2,J3) = IROWP
1470 NUETA = NUETA + 1
1471 IE(NELAST) = IROWP
1472 IF (J .NE. KR1) GO TO 2330
1473 E(NELAST) = Y(IROWP)
1474 GO TO 2340
1475 E(NELAST) = 1.
1476 NETA = NETA + 1
1477 LE(NETA+1) = NELEM + 1
1478 NUELEM = NUELEM + 1
1479 LE(NLAST) = NELAST
1480 NELAST = NELAST - 1
1481 NTLAST = NTLAST - 1
1482
1483 C UPDATE ROW COUNTS
1484
1485 KNR=NROW(JC,J2,J3)
1486 DO 2350 I = 1,KNR
1487 IF (DABS(Y(I)) .LE. ZTOLZE) GO TO 2350
1488 IF (HREG(I) .GE. O) GO TO 2350
1489 HREG(I) = HREG(I) - INCR
1490 IF (HREG(I) .GE. O) HREG(I) = -1
1491 CONTINUE
1492 HREG(IROWP) = O
1493 CONTINUE
1494
1495 C MERGE L AND U ETAS
1496
1497 NLETA = NETA
1498 NETA = NLETA + NUETA
1499 NLELEM = NELEM
1500 NELEM = NLELEM + NUELEM
1501 IF (NUELEM .EQ. O) GO TO 3550
1502 CALL SHFTE
1503
1504 C INSERT SLACKS FOR DELETED COLUMNS
1505
1506 KNR=NROW(JC,J2,J3)
1507 DO 3600 I = 1,KNR
1508 IF (JH(I,JC,J2,J3) .NE. O) GO TO 3600

```



```

1509      JH(I,JC,J2,J3) = I
1510      IROWP = I
1511      CALL UNPACK(IROWP)
1512      CALL FTRAN(1)
1513      CALL WRETA
1514      CONTINUE
1515      3600
1516      C
1517      C   UPDATE X
1518
1519      CALL SHIFTR(1,3)
1520      KNC=NCOL(JC,J2,J3)
1521      DO 9000 J=1,KNC
1522      IF (KINBAS(J,JC,J2,J3)) 8600,.8700,9000
1523      DE = XUB(J,JC,J2,J3)
1524      GO TO 8750
1525      DE = XLB(J,JC,J2,J3)
1526      LL = LA(J,JC,J2,J3)
1527      KK = LA(J+1,JC,J2,J3) - 1
1528      DO 8800 I=LL,KK
1529      IR = IA(I,JC,J2,J3)
1530      Y(IR) = Y(IR) - A(I,JC,J2,J3)*DE
1531      CONTINUE
1532      CALL FTRAN(1)
1533      CALL SHIFTR(3,2)
1534      RETURN
1535      END
1536      C-----
1537      SUBROUTINE UNPACK(IV)
1538      C
1539      C   EXPANDS COMPRESSED MATRIX COLUMNS
1540      C   SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
1541      C   BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1542      C   ***DESCRIPTION OF PARAMETERS***
1543      C   IV = PARAMETER INDEXING COLUMN TO BE EXPANDED (INPUT)
1544      C
1545      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
1546      1   INTEGER*4 (I-N,Q)
1547      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1548      DOUBLE PRECISION E(3000)
1549      REAL A(3000,3,3,1)
1550      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1551      22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
1552      1   A,E,MSTAT,IOBU,IROWP,ITCNT,
1553      2   INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1
1554      ),NETA,
1555      3   NLELEM,NLETA,NUELEM,NUETA,JII(350,3,3,1),KINBAS(602,3,3,1),LA(
1556      602,3,3,1),
1557      4   LE(1002),IA(3000,3,3,1),IE(3000),
1558      5   ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
1559      6   PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XITMP(602),
1560      7   LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1561      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1562      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IRST(3),
1563      1   PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1564      2   NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1565      C
1566      JC=JCUR

```

```

1567 J2=JPER(2)
1568 J3=JPER(3)
1569 IF(JC.EQ.NPER)J2=1
1570 KNR=NROW(JC,J2,J3)
1571 DO 100 I = 1,KNR
1572 Y(I) = O.
1573 CONTINUE
1574 C
1575 LL = LA(IV,JC,J2,J3)
1576 KK = LA(IV+1,JC,J2,J3) - 1
1577 DO 200 I = LL,KK
1578 IR = IA(I,JC,J2,J3)
1579 Y(IR) = A(I,JC,J2,J3)
1580 CONTINUE
1581 RETURN
1582 END
1583 C-----SUBROUTINE BUNPCK(IV)
1584 C
1585 C EXPANDS COMPRESSED MATRIX COLUMNS
1586 C SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1, WRITTEN
1587 C BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1588 C ***DESCRIPTION OF PARAMETERS***
1589 C IV = PARAMETER INDEXING COLUMN TO BE EXPANDED (INPUT)
1590 C
1591 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
1592 C INTEGER*4 (I-N,Q)
1593 C
1594 C INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1595 C DOUBLE PRECISION E(3000)
1596 C REAL A(3000,3,3,1)
1597 C COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1598 C 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
1599 C 1 .A.E.MSTAT,IOBJ,IROWP,ITCNT,
1600 C 2 INVFRQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
1601 C ),NETA,
1602 C 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
1603 C 3 602,3,3,1),
1604 C 4 LE(1002),IA(3000,3,3,1),IE(3000),
1605 C 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1).
1606 C
1607 C 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
1608 C 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1609 C COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1610 C COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
1611 C 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1612 C 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1613 C
1614 C
1615 C JC=JCUR
1616 C J2=JPER(2)
1617 C J3=JPER(3)
1618 C KNRO=NROW(JC+1,1,1)
1619 C DO 100 I = 1,KNRO
1620 C Y(I) = O.
1621 C CONTINUE
1622 C
1623 C LL = LBN(IV,JC,J2,J3)
1624 C KK = LBN(IV+1,JC,J2,J3) - 1
1625 C DO 200 I = LL,KK

```

```

1625 IR = IBN(I,JC,J2,J3)
1626 Y(IR) = ABN(I,JC,J2,J3)
1627 CONTINUE
1628 RETURN
1629 END
1630 C-----
1631 SUBROUTINE SHFTE
1632 C
1633 SUBROUTINE FOR INVERT
1634 SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM 1. WRITTEN
1635 BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1636 C
1637 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
1638 1 INTEGER*4 (I-N,Q)
1639 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1640 DOUBLE PRECISION E(3000)
1641 REAL A(3000,3,3,1)
1642 C
1643 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
1644 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QQ,QS,QP
1645 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1646 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
1647 1 A,E,MSTAT,IOBJ,IROWP,ITCNT.
1648 2 INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
1649 ) ,NETA.
1650 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
1651 3 602,3,3,1).
1652 4 LE(1002),IA(3000,3,3,1),IE(3000),
1653 5 ATPM(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1).
1654
1655 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XI,IMP(602),
1656 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1657 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1658 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
1659 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1660 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1661 C
1662 C SHIFT IE AND E OF U ELEMENTS
1663 C
1664 C
1665 NF = NEMAX - NUELEM + 1
1666 INCR = 0
1667 DO 1000 I = NF,NEMAX
1668 INCR = INCR + 1
1669 IE(NLELEM + INCR) = IE(I)
1670 E(NLELEM + INCR) = E(I)
1671 CONTINUE
1672 1000
1673 C
1674 IDIF = NEMAX - NLELEM - NUELEM
1675 NF = NTMAX - NUETA + 1
1676 INCR = 0
1677 DO 2000 I = NF,NTMAX
1678 INCR = INCR + 1
1679 LE(NLETA + INCR) = LE(I) - IDIF
1680 CONTINUE
1681 LE(NETA+1) = NELEM + 1
1682 RETURN
1683 END
1684 C-----

```

```

1683 SUBROUTINE UPBETA
1684
1685 C      UPDATES RIGHT-HAND SIDES TO REFLECT NEW BASIS RESULTING FROM
1686 C      CURRENT SIMPLEX PIVOT
1687 C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1688 C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1689
1690 C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
1691 1  INTEGER*4 (I-N,Q)
1692 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1693 DOUBLE PRECISION E(3000)
1694 REAL A(3000,3,3,1)
1695 COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
1696 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1697 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
1698 1  A,E,MSTAT,IOBJ,IROWP,ITCNT
1699 2  INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1)
1700 2  ),NETA,
1701 3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
1702 3  602,3,3,1)
1703 4  LE(1002),IA(3000,3,3,1),IE(3000),
1704 5  ATPM(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
1705
1706 6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
1707 7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1708 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1709 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
1710 1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
1711 2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1712
1713 C
1714 JC=JCUR
1715 J2=JPER(2)
1716 J3=JPER(3)
1717 KNR=NROW(JC,J2,J3)
1718 DO 1000 I=1,KNR
1719 X(I,JC,J2,J3) = X(I,JC,J2,J3) - Y(I)*DP
1720 IF (NPIVOT .EQ. 1) GO TO 2000
1721 KINBAS(JCOLP,JC,J2,J3) = -(KINBAS(JCOLP,JC,J2,J3) + 1)
1722 RETURN
1723 X(IROWP,JC,J2,J3) = DE
1724 IVOUT = JH(IROWP,JC,J2,J3)
1725 KINBAS(JCOLP,JC,J2,J3) = IROWP
1726 KINBAS(IVOUT,JC,J2,J3) = IPTYPE
1727 JH(IROWP,JC,J2,J3) = JCOLP
1728 RETURN
1729 END
1730
1731 C-----SUBROUTINE NORMAL(ITSINV)
1732
1733 C      SERVES AS MASTER PROGRAM FOR LINEAR PROGRAMMING COMPONENT
1734 C      (REVISED,PRIMAL-SIMPLEX METHOD) OF BRANCH-AND-BOUND ROUTINE BB
1735 C      SUBROUTINE ADAPTED FROM LINEAR PROGRAMMING CODE LPM-1, WRITTEN
1736 C      BY J.A. TOMLIN (OPERATIONS RESEARCH, STANFORD UNIVERSITY)
1737 C      ***DESCRIPTION OF PARAMETERS***
1738 C      ITSINV = NUMBER OF SIMPLEX ITERATIONS SINCE LAST BASIS
1739 C      INVERSION (INPUT/OUTPUT)
1740
1741 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),

```

```

1741 1 INTEGER*4 (I-N,Q)
1742 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1743 DOUBLE PRECISION E(3000)
1744 REAL A(3000,3,3,1)
1745
1746 C
1747 COMMON/BL5/DRES,ICNAM
1748 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
1749 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QQ,QS,QP
1750 COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOP1
1751 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
1752 22),XLB(602,3,3,1),XJB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
1753 1 A,E,MSTAT,IOBJ,IROWP,ITCNT,
1754 2 INVFRQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
1755 ),NETA,
1756 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
1757 602,3,3,1),
1758 4 LE(1002),IA(3000,3,3,1),IE(3000),
1759 5 ATPM(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
1760 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XIMP(602),
1761 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1762 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCJR,JPASS,NPER
1763 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIRAR(602),YBX(350),IBST(3),
1764 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JUSTCH(5,5,5),NCLUR(3),
1765 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1766
1767 C
1768 JC=JCUR
1769 J2=JPER(2)
1770 J3=JPER(3)
1771 NELEM=NELM(JC,J2,J3)
1772 IF (ITSINV.LT. INVFRQ) GO TO 1500
1773 CALL INVERT
1774 WRITE(6,1001)
1775 FORMAT(' ITNS',2X,'RRES',5X,' PIV',6X,' IN',3X,'OUT',2X,
1776 1 'OBJ',7X,'CMIN',7X,'CMAX',7X,'THET',3X,'NINF',1X,'NOP1')
1777 ITSINV = 0
1778 C
1779 SIMPLEX CYCLE
1780 C
1781 CALL FORMC
1782 TRANSFER AFTER PHASE 1 FOR FIRST PASS
1783 IF((MSTAT.EQ.QF).AND.(JPASS.EQ.1)) GO TO 6000
1784 CALL BTRAN
1785 CALL PRICE
1786 IF (JCOLP.GT. 0) GO TO 3000
1787 IF (MSTAT.EQ.QI) GO TO 2000
1788 MSTAT = QBL
1789 GO TO 6000
1790 MSTAT = QN
1791 GO TO 6000
1792 CALL UNPACK(JCOLP)
1793 CALL FTRAN(1)
1794 CALL CHUZR
1795 CALL UPBETA
1796 ITCNT = ITCNT + 1
1797 ITSINV = ITSINV + 1
1798 IF (NPIVOT.EQ. 0) GO TO 4010
1799 IF (NELEM.GT.(NEMAX-NROW(JC,J2,J3))) GO TO 1000

```

```

1799 CALL WRETA
1800 CALL RHCHK
1801 IF(ITCNT.GE.ITRFRQ) WRITE(6,4011) ITCNT
1802 FORMAT(' MORE THAN',I7,' ITERATIONS')
1803 IF(ITCNT.GE.ITRFRQ) JCUR=NPER+1
1804 IF (ITCNT.GE. ITRFRQ) GO TO 6000
1805 WRITE(6,1002)ITCNT,DRES,APV,JCOLP,IROWP,X(1,JC,J2,J3),CMIN,CMAX
1806 1 .DP,NINF,NOPT
1807 1002 FORMAT(I4,I4,E9.2,1X,E9.2,1X,I4,1X,I4,1X,I4,1X,I4,1X,E10.2,1X,E9.2,1X,F9.2,1X,
1808 1 E9.2,1X,I4,1X,I4)
1809 IF(DRES.GT.10*ZTOLZE) GO TO 1000
1810 4010 IF (ITSINV .GE. INVFRQ) GO TO 1000
1811 GO TO 1500
1812 6000 RETURN
1813 END
1814 C-----
1815 SUBROUTINE NDCOM
1816 C
1817 C THIS SUBROUTINE PERFORMS ALL THE NECESSARY STOCHASTIC DECOMP
1818 C OSITION STEPS.
1819 C
1820 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
1821 1 INTEGER*4 (I-N,Q)
1822 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
1823 INTEGER ICNAM(602.2),NAME(6)
1824 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
1825 REAL A(3000.3,3,1)
1826 C
1827 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
1828 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,OG,OS,OP
1829 COMMON DE,DP,B(350.3,3,1),X(350.3,3,1),Y(350),YTEMP(602),YTEMP1(6
1830 1 22),XLB(602.3,3,1),XUB(602.3,3,1),XKSI(350.3,3,1),YPI(350.3,3,1)
1831 1 .A.E,MSTAT,IOBJ,IROWP,ITCNT,
1832 2 INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
1833 ),NETA,
1834 3 NLELEM,NLETA,NUELEM,NUETA,JH(350.3,3,1),KINBAS(602.3,3,1),LA(
1835 3 602.3,3,1),
1836 4 LE(1002),IA(3000.3,3,1),IE(3000),
1837 5 ATMP(3000),ABN(600.3,3,1),IBN(600.3,3,1),LBN(602.3,3,1),
1838 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
1839 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
1840 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
1841 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
1842 1 PRBV(3,5),PRST(5,5,5),CBST(3,5,5),IBASE,JSTCH(5,5,5),MCUR(3),
1843 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
1844 FOR ALL FEASIBLE, PASSBACK
1845 C
1846 C IF(JPASS .EQ. 2) GO TO 70
1847 C
1848 C IF CURRENT FEASIBLE GO ON
1849 C
1850 WRITE(6,300) MSTAT
1851 FORMAT(' STATUS AT NDCOM=' ,A4)
1852 WRITE(6,301) JCUR
1853 FORMAT(' AT BEG OF NDCOM, JCUR=' ,I6)
1854 C
1855 C IF(MSTAT.NE. QN) GO TO 20
1856 C

```

```

1857 C      HERE TOTAL PROBLEM IS INFEASIBLE
1858 C
1859 IF(JCUR .EQ. 1) JCUR=NPUR+1
1860 IF(JCUR .EQ. NPUR+1) RETURN
1861 WRITE(6,303) JCUR
1862 FORMAT(' INFEASIBLE AT JCUR = ',I6)
1863 C
1864 C      ADD CUT AND RETURN
1865 C
1866 WRITE(6,201)
1867 FORMAT('MAKING A FEASIBILITY CUT')
1868 CALL FEASCT
1869 JPER(JCUR)=1
1870 JCUR=JCUR-1
1871 RETURN
1872 C
1873 C      FEASIBLE, SO CONTINUE DOWN
1874 C
1875 C
1876 IF(JPER(JCUR) .LT. NND(JCUR)) GO TO 30
1877 K=JCUR
1878
1879 C      ITERATE TO FIND THE PROPER INDEX TO CHANGE
1880 C
1881 C
1882 CONTINUE
1883 K=K-1
1884 IF(K .LE. 1) GO TO 28
1885 IF(JPER(K) .EQ. NND(K)) GO TO 25
1886 JPER(K)=JPER(K)+1
1887 KK=K+1
1888 DO 26 I=KK,JCUR
1889 JPER(I)=1
1890 CONTINUE
1891 RETURN
1892 C
1893 C      THERE, YOU'VE FINISHED THE CURRENT PERIOD
1894 C
1895 C
1896 C      MULTIPLY OUT THE CURRENT PERIOD, CARRY TO NEXT
1897 C
1898 IF(JCUR .GT. 1) GO TO 285
1899 CALL BPRODX
1900 CALL FRMRHS
1901 GO TO 290
1902 C
1903 IF(JCUR .EQ. NPUR) GO TO 290
1904 NND=NND(JCUR)
1905 DO 280 L=1,NND
1906 JPER(JCUR)=L
1907 CALL BPRODX
1908 CALL FRMRHS
1909 CONTINUE
1910 C
1911 JCUR=JCUR+1
1912 DO 29 I=1,JCUR
1913 JPER(I)=1
1914 CONTINUE

```

```

1915 WRITE(6,302) JCUR
1916 FORMAT(' AT END OF ND, JCUR=',I6)
1917 IF(JCUR .LE. NPER) RETURN
1918 JPASS=2
1919 JCUR=NPER
1920 RETURN
1921 30 CONTINUE
1922 JPER(JCUR)=JPER(JCUR)+1
1923 RETURN
1924 C
1925 C NOW, FOR THE BACK PASS
1926 C
1927 70 CONTINUE
1928 C
1929 C THIS SAYS SET THE OPTIMAL FLAG AT FALSE
1930 C
1931 C NFLG = 0
1932 C
1933 C CHECK IF FEASIBLE
1934 C
1935 C IF(MSTAT .NE. QN) GO TO 80
1936 C
1937 C IF FIRST PERIOD INFEASIBLE
1938 C
1939 C IF(JCUR .EQ. 1) WRITE(6,72)
1940 72 FORMAT(' INFEASIBLE ON PASS 2')
1941 IF(JCUR .EQ. 1) STOP
1942 C
1943 C FEASIBLE CUT
1944 C
1945 C CALL FEASCT
1946 JPER(JCUR)=1
1947 JCUR=JCUR-1
1948 RETURN
1949 C
1950 C HERE IT'S FEASIBLE
1951 C
1952 80 CONTINUE
1953 C
1954 C GET RID OF CUTS YOU DON'T NEED
1955 C
1956 C
1957 CALL DLETCT
1958 IF((JCUR .EQ. NPER-1).AND.(NFLG.EQ.1)) GO TO 85
1959 C
1960 C THIS PART SAYS MOVE ON DOWN TO NEXT PERIOD
1961 C
1962 CALL BPRODX
1963 CALL FRMRHS
1964 JCUR=JCUR+1
1965 DO 82 I=JCUR,NPER
1966 JPER(I)=1
1967 CONTINUE
1968 82 CONTINUE
1969 JJC=JCUR+1
1970 DO 83 I=JJC,NPER
1971 JPER(I)=1
1972 83 CONTINUE

```



```

1973 NFLG=0
1974 RETURN
1975 CONTINUE
1976 85
1977 C CHECK FOR PERIOD CURRENTLY OPTIMAL
1978 C
1979 C IF(JPER(NPER-1) .EQ. NND(NPER-1)) GO TO 90
1980 JPER(NPER-1)=JPER(NPER-1)+1
1981 NFLG=0
1982 RETURN
1983 CONTINUE
1984 90
1985 C CHECK LAST PERIOD OPTIMALITY
1986 C
1987 C NFLG=0
1988 IF(NTH(1,1,1).EQ.1) CALL OPTCHK(2)
1989 IF(NFLG .EQ. 1) GO TO 95
1990 C
1991 C THERE, IT'S NOT OPTIMAL, GO BACK
1992 C
1993 CALL LKHDCT(NPER-1)
1994 JCUR=JCUR-1
1995 GO TO 81
1996 CONTINUE
1997 95
1998 K=JCUR
1999 C
2000 C ITERATE BACK THROUGH EACH PERIOD TO CHECK OPTIMALITY
2001 C
2002 CONTINUE
2003 NFLG=0
2004 K=K-1
2005 C
2006 C IF K=1, YOU'RE DONE
2007 C
2008 IF(K .EQ. 1) JCUR=NPER+1
2009 IF(K .EQ. 1) RETURN
2010 JPER(K)=JPER(K)+1
2011 IF(JPER(K) .LE. NND(K)) GO TO 101
2012 JPER(K)=JPER(K)-1
2013 C
2014 C CHECK IF OPTIMAL
2015 C
2016 J2=JPER(2)
2017 J3=JPER(3)
2018 IF((K .EQ. 3) .AND. (NTH(2,J2,1) .EQ. 1)) CALL OPTCHK(K)
2019 IF((K .EQ. 2) .AND. (NTH(1,1,1) .EQ. 1)) CALL OPTCHK(K)
2020 C
2021 C YES, THEN GO BACK
2022 C
2023 IF(NFLG .EQ. 1) GO TO 96
2024 C
2025 C NO, THEN PLACE A CUT
2026 C
2027 CALL LKHDCT(K)
2028 JCUR=K-1
2029 GO TO 81
2030 CONTINUE
2031 KK=K+1
2032 101

```

```

2031 DO 102 I=KK,JCUR
2032 JPER(I)=1
2033 CONTINUE
2034 JCUR=K
2035 RETURN
2036 END
2037 -----
2038 SUBROUTINE STRPT
2039 C
2040 C REPORTS ON THE CURRENT STATUS
2041 C
2042 C
2043 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2044 1 INTEGER*4 (I-N,Q)
2044 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2045 INTEGER ICNAM(602,2),NAME(6)
2046 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2047 REAL A(3000,3,3,1)
2048 C
2049 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2050 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2051 1 A,E,MSTAT,IOBJ,IROWP,ITCNT
2052 2 INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFILM(3,3,1
2053 ),NETA,
2054 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2055 3 602,3,3,1),
2056 4 LE(1002),IA(3000,3,3,1),IE(3000),
2057 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
2058
2059 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XIMP(602),
2060 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
2061 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIRAR(602),YBX(350),IRST(3),
2063 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NFCUR(3),
2064 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2065 JC=JCUR
2066 J2=JPER(2)
2067 J3=JPER(3)
2068 KNR=NROW(JC,J2,J3)
2069 KNC=NCOL(JC,J2,J3)
2070 WRITE(6,11) ITCNT
2071 WRITE(6,10) NPASS
2072 WRITE(6,1) JC,J2,J3
2073 WRITE(6,2)
2074 WRITE(6,3)(X(I,JC,J2,J3),I=1,KNR)
2075 WRITE(6,4)
2076 WRITE(6,5)(JH(I,JC,J2,J3),I=1,KNR)
2077 WRITE(6,6)
2078 WRITE(6,5)(KINBAS(I,JC,J2,J3),I=1,KNR)
2079 WRITE(6,7)
2080 WRITE(6,3)(B(I,JC,J2,J3),I=1,KNR)
2081 WRITE(6,8)
2082 WRITE(6,3)(Y(I),I=1,KNR)
2083 WRITE(6,9)
2084 WRITE(6,3)(YPI(I,JC,J2,J3),I=1,KNR)
2085 FORMAT(' WE ARE AT PROBLEM',I5,',',',',I5,',',',',I5)
2086 FORMAT(' X VALUES')
2087 FORMAT(' JH')
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188

```

```

2089      5  FORMAT(15I5)
2090      6  FORMAT(' KINBAS')
2091      7  FORMAT(' BS CURRENTLY')
2092      8  FORMAT(' CURRENT Y')
2093      9  FORMAT(' CURRENT PI')
2094     10  FORMAT('//, ' WE HAVE MADE ',I5, ' PASSES')
2095     11  FORMAT(I4, ' ITERATIONS SO FAR')
2096      RETURN
2097      END
2098
2099      C-----
2100      C  SUBROUTINE ACHECK
2101
2102      C  LOOK AT THE VALUES IN A AT THE MOMENT
2103      C
2104      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
2105      1  INTEGER*4 (I-N,Q)
2106      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2107      INTEGER ICNAM(602,2),NAME(6)
2108      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2109      REAL A(3000,3,3,1)
2110
2111      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2112      22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2113      1  .A.E,MSTAT,IOBJ,IROWP,ITCNT.
2114      2  INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NEIM(3,3,1
2115      3  ),NETA.
2116      3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2117      4  602,3,3,1),
2118      4  LE(1002),IA(3000,3,3,1),IE(3000),
2119      5  ATEMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1).
2120
2121      6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602).
2122      7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
2123      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2124      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
2125      1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),PICUR(3),
2126      2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2127      JC=JCUR
2128      J2=JPER(2)
2129      J3=JPER(3)
2130
2131      WRITE(6,1) JC,J2,J3,NPASS
2132      FORMAT('//, ' A MATRIX OF ',I5,I5,I5, ' AT PASS ',I5)
2133      KNRO=NROW(JC,J2,J3)+1
2134      KNC=NCOL(JC,J2,J3)
2135      KNR=NROW(JC,J2,J3)
2136      DO 10 I=KNRO,KNC
2137      IND=I
2138      CALL UNPACK(IND)
2139      WRITE(6,2) IND
2140      FORMAT(' COLUMN ',I5)
2141      WRITE(6,3)(Y(J),J=1,KNR)
2142      FORMAT(15F7.2)
2143      CONTINUE
2144      RETURN
2145      END
2146      C-----
2147      C  SUBROUTINE DLETC

```

```

2147 C DELETES CUTS THAT ARE SLACK.
2148 C
2149     IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2150     1  INTEGER*4 (I-N,Q)
2151     INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2152     INTEGER ICNAM(602,2),NAME(6)
2153     DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2154     REAL A(3000,3,3,1)
2155
2156     COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2157     1  22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2158     1  .A,E,MSTAT,IOBJ,IROWP,ITCNT.
2159     2  INVERQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELEM(3,3,1
2160     2  ),NETA,
2161     3  NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2162     3  602,3,3,1),
2163     4  LE(1002),IA(3000,3,3,1),IE(3000),
2164     5  ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1).
2165
2166     6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
2167     7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
2168     COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2169     COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
2170     1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSICH(5,5,5),NCUR(3),
2171     2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2172
2173 C CHECK FOR BASIC SLACKS
2174 C
2175     WRITE(6,20)
2176     FORMAT('/',/, '  LOOKING FOR CUTS TO DELETE')
2177     JC=JCUR
2178     J2=JPER(2)
2179     J3=JPER(3)
2180     NRX=NROW(JC,J2,J3)
2181     IF(NRX.EQ.NROWP(JC,J2,J3)) GO TO 10
2182     KNRPO=NROWP(JC,J2,J3)+1
2183     DO 10 I=KNRPO,NRX
2184     IF (KINBAS(I,JC,J2,J3).NF.I) GO TO 10
2185     C SLACK ON I-M IS BASIC
2186     IF(X(I,JC,J2,J3).LT.O.5) GO TO 10
2187     C LARGE POSITIVE SLACK TO DELETE
2188     WRITE(6,21) I,X(I,JC,J2,J3)
2189     FORMAT('/',/, '  DELETING ROW',I6,'  VALUE = ',E11.3)
2190     C UP COLUMNS
2191     KNR=NROW(JC,J2,J3)
2192     DO 13 J=1,KNR
2193     IF(JH(J,JC,J2,J3).GT.I) JH(J,JC,J2,J3)=JH(J,JC,J2,J3)-1
2194     13 CONTINUE
2195     IF(I.EQ.NRX) GO TO 15
2196     KNRM=NROW(JC,J2,J3)-1
2197     DO 14 J=I,KNRM
2198     C MOVE DOWN ROW INFO
2199     X(J,JC,J2,J3)=X(J+1,JC,J2,J3)
2200     B(J,JC,J2,J3)=B(J+1,JC,J2,J3)
2201     JH(J,JC,J2,J3)=JH(J+1,JC,J2,J3)
2202     14 CONTINUE
2203     15 CONTINUE
2204     KNCM=NCOL(JC,J2,J3)-1

```

```

2205      DO 3 J=1,KNCM
2206      IF(J.LT.I) GO TO 115
2207      C MOVE DOWN COL INFO
2208      XLB(J,JC,J2,J3)=XLB(J+1,JC,J2,J3)
2209      XUB(J,JC,J2,J3)=XUB(J+1,JC,J2,J3)
2210      KINBAS(J,JC,J2,J3)=KINBAS(J+1,JC,J2,J3)
2211      CONTINUE
2212      IF(KINBAS(J,JC,J2,J3).GT.I)
2213      1 KINBAS(J,JC,J2,J3)=KINBAS(J,JC,J2,J3) - 1
2214      CONTINUE
2215      C SHIFT UP LISTS
2216      KNC=NCOL(JC,J2,J3)
2217      LNEW=LA(I-1,JC,J2,J3)
2218      II=I+1
2219      DO 200 J=II,KNC
2220      K1=LA(J,JC,J2,J3)
2221      K2=LA(J+1,JC,J2,J3) - 1
2222      LA(J,JC,J2,J3)=LNEW+1
2223      DO 30 K=K1,K2
2224      IF(IA(K,JC,J2,J3).EQ.I) GO TO 30
2225      LNEW=LNEW+1
2226      A(LNEW,JC,J2,J3)=A(K,JC,J2,J3)
2227      IA(LNEW,JC,J2,J3)=IA(K,JC,J2,J3)
2228      IF(IA(K,JC,J2,J3).GT.I) IA(LNEW,JC,J2,J3)=IA(I NEW,JC,J2,J3) - 1
2229
2230      30 CONTINUE
2231      200 CONTINUE
2232      LA(KNC+1,JC,J2,J3)=LNEW+1
2233      C SHIFT UP LA
2234      DO 35 J=I,KNC
2235      LA(J,JC,J2,J3)=LA(J+1,JC,J2,J3)
2236      CONTINUE
2237      NELM(JC,J2,J3)=LNEW
2238      NCOL(JC,J2,J3)=NCOL(JC,J2,J3) - 1
2239      NROW(JC,J2,J3)=NROW(JC,J2,J3) - 1
2240      CALL STRPRT
2241      CALL ACHECK
2242      10 CONTINUE
2243      11 CONTINUE
2244      RETURN
2245      END
2246      C-----
2247      SUBROUTINE FEASCT
2248      C FINDS AND ADDS A FEASIBILITY CUT TO NODE 1
2249      C
2250      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2251      1 INTEGER*4 (I-N,Q)
2252      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2253      INTEGER ICNAM(602,2),NAME(6)
2254      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2255      REAL A(3000,3,3,1)
2256      C
2257      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2258      1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2259      1 .A,E,MSTAT,IOBJ,IROWP,ITCNT,
2260      2 INVFREQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
2261      ),NETA,
2262      3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(

```

```

2263      3 602,3,3,1),
2264      4 LE(1002),IA(3000,3,3,1),IE(3000),
2265      5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
2266
2267      6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XITMP(602),
2268      7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NFH(3,3,1),NR
2269      COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2270      COMMON/BLOCK4/ BND(350),XIDPT,XRHO,YPIBAR(602),YBX(350),IDST(3),
2271      1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),JCUR(3),
2272      2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2273
2274      C FIRST, FORM THE ROW
2275
2276      C
2277      WRITE(6,200)
2278      FORMAT(' MAKING A FEASIBILITY CUT')
2279      JC=JCUR
2280      J2=JPER(2)
2281      J3=JPER(3)
2282      JCT=JC
2283      J2L=J2
2284      J3T=J3
2285      IF(JC.EQ.2) J2=1
2286      IF(JC.EQ.3) J3=1
2287      KNC=NCOL(JC-1,J2,J3)+1
2288      DO 10 J=1,KNC
2289          Y(J)=0.
2290      YTEMP(J)=0.
2291      10 CONTINUE
2292
2293      C COLUMN BY COLUMN IN TEMP STOR
2294      IF(JCUR.EQ.NPER) J2L=1
2295
2296      C
2297      IF(JCT.EQ.NPER) NROWP(JCT,J2L,J3T)=NROWP(JCT,1,1)
2298      JND=NROW(JC-1,J2,J3)+1
2299      IF(NROWP(JC-1,J2,J3).GE.NROWP(JCT,J2L,J3T)) NRO=NROWP(JCT,J2L,J3T)
2300      IF(NROWP(JC-1,J2,J3).LT.NROWP(JCT,J2L,J3T)) NRO=NROWP(JC-1,J2,J3)
2301      JNX = NROW(JC-1,J2,J3) - NRO
2302      JCUR=JC-1
2303      JPER(2)=J2
2304      JPER(3)=J3
2305      KNC=NCOL(JC-1,J2,J3)
2306      DO 30 J=JND,KNC
2307          JNL=J-JNX
2308          CALL BUNPCK(JNL)
2309          KNR=NROW(JCT,J2L,J3T)
2310          DO 20 K=1,KNR
2311              YTEMP(J)=YTEMP(J)+YPI(K,JCT,J2L,J3T)*Y(K)
2312          CONTINUE
2313      20 CONTINUE
2314      30 CONTINUE
2315      C FIND RHS TOO
2316      C
2317      KNR=NROW(JCT,J2L,J3T)
2318      DO 40 K=1,KNR
2319          YTEMP(NCOL(JC-1,J2,J3)+1)=YTEMP(NCOL(JC-1,J2,J3)+1)+
2320          1 YPI(K,JCT,J2L,J3T)*XKSI(K,JCT,J2L,J3T)
2321      40 CONTINUE

```

```

2321 C CHANGE RHS FOR UPPER BOUNDED VARIABLES
2322 KNC=NCOL(JCT,J2L,J3T)
2323 DO 41 K=1,KNC
2324 IF(KINBAS(K,JCT,J2L,J3T).NE.-1) GO TO 41
2325 JCUR=JCUR+1
2326 CALL UNPACK(K)
2327 JCUR=JCUR-1
2328 CBAR=O
2329 DO 42 J=1,KNR
2330 CBAR=CBAR + Y(J)*YPI(J,JCT,J2L,J3T)
2331 CONTINUE
2332 42 YTEMP(NCOL(JC-1,J2,J3)+1)=YTEMP(NCOL(JC-1,J2,J3)+1)-CRAR*
2333 1 XUB(K,JCT,J2L,J3T)
2334 41 CONTINUE
2335 C
2336 C SHIFT A(*,JC-1,J2,J3) DOWN
2337 C FIRST, COPY INTO TEMPS
2338 C
2339 KLM=NELM(JC-1,J2,J3)
2340 DO 50 I=1,KLM
2341 ATMP(I)=A(I,JC-1,J2,J3)
2342 ITMP(I)=IA(I,JC-1,J2,J3)
2343 50 CONTINUE
2344 C
2345 C ADD COL AND ROW FOR SLACK
2346 C
2347 NCOL(JC-1,J2,J3)=NCOL(JC-1,J2,J3)+1
2348 NROW(JC-1,J2,J3)=NROW(JC-1,J2,J3)+1
2349 NELM(JC-1,J2,J3)=NELM(JC-1,J2,J3)+NCOL(JC-1,J2,J3)-
2350 1 NROW(JC-1,J2,J3)+1
2351 JH(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
2352 C
2353 C COPY COL VALUES
2354 C
2355 KNCM=NCOL(JC-1,J2,J3)-1
2356 DO 60 I=1,KNCM
2357 KBTMP(I)=KINBAS(I,JC-1,J2,J3)
2358 LTMP(I)=LA(I,JC-1,J2,J3)
2359 XLTMP(I)=XLB(I,JC-1,J2,J3)
2360 XUTMP(I)=XUB(I,JC-1,J2,J3)
2361 60 CONTINUE
2362 C
2363 C UPDATE COLUMNS
2364 C
2365 KINBAS(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
2366 LA(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
2367 XLB(NROW(JC-1,J2,J3),JC-1,J2,J3)=O.
2368 XUB(NROW(JC-1,J2,J3),JC-1,J2,J3)=1.OE+7
2369 JND=O
2370 C
2371 KNRO=NROW(JC-1,J2,J3)+1
2372 KNC=NCOL(JC-1,J2,J3)
2373 DO 70 I=KNRO,KNC
2374 JND=JND+1
2375 LA(I,JC-1,J2,J3)=LTMP(I-1)+JND
2376 KINBAS(I,JC-1,J2,J3)=KBTMP(I-1)
2377 XLB(I,JC-1,J2,J3)=XTMP(I-1)
2378 XUB(I,JC-1,J2,J3)=XUTMP(I-1)

```

```

2379 70 CONTINUE
2380 LA(NCOL(JC-1,J2,J3)+1,JC-1,J2,J3)=NELM(JC-1,J2,J3)+1
2381 C
2382 A(NROW(JC-1,J2,J3),JC-1,J2,J3)=1.
2383 IA(NROW(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
2384 C
2385 C UPDATE A AND IA
2386 C
2387 DO 80 I=KNRO,KNC
2388 IND=I
2389 KND=LA(IND,JC-1,J2,J3)-1
2390 IF(KND.EQ.NROW(JC-1,J2,J3)) GO TO 81
2391 IF(DABS(YTEMP(IND-2)) .GE. ZTOLZE) GO TO 75
2392 C
2393 C SHIFT ALL DOWN
2394 C
2395 NELM(JC,J2,J3)=NELM(JC,J2,J3)-1
2396 KKNC=KNC+1
2397 DO 72 II=IND,KKNC
2398 LA(II,J2,J3)=LA(II,J2,J3)-1
2399 CONTINUE
2400 KLM=NELM(JC,J2,J3)+1
2401 DO 73 II=KND,KLM
2402 A(II,J2,J3)=A(II+1,J2,J3)
2403 IA(II,J2,J3)=IA(II+1,J2,J3)
2404 CONTINUE
2405 GO TO 81
2406 C
2407 CONTINUE
2408 A(KND,JC-1,J2,J3)=YTEMP(IND-2)
2409 IA(KND,JC-1,J2,J3)=NROW(JC-1,J2,J3)
2410 CONTINUE
2411 MND=LA(IND+1,JC-1,J2,J3)-2
2412 IF(IND.EQ.NCOL(JC-1,J2,J3)) MND=NELM(JC-1,J2,J3)-1
2413 KKND=KND+1
2414 DO 90 J=KKND,MND
2415 JNX=J-IND+NROW(JC-1,J2,J3)
2416 A(J,J2,J3)=ATMP(JNX)
2417 IA(J,J2,J3)=ITMP(JNX)
2418 C
2419 CONTINUE
2420 GO TO 80
2421 C
2422 C SPECIAL FOR NO LOOK CUTS
2423 IF(NTH(JC-1,J2,J3).EQ.1) GO TO 82
2424 A(NELM(JC-1,J2,J3),JC-1,J2,J3)=YTEMP(NCOL(JC-1,J2,J3)-1)
2425 IA(NELM(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
2426 GO TO 83
2427 C
2428 CONTINUE
2429 A(NELM(JC-1,J2,J3),JC-1,J2,J3)=0.
2430 IA(NELM(JC-1,J2,J3),JC-1,J2,J3)=NROW(JC-1,J2,J3)
2431 C
2432 C UPDATE THE JH TOO
2433 KNRM=NROW(JC-1,J2,J3)-1
2434 DO 85 I=1,KNRM
2435 IF (JH(I,J2,J3).GE.NROW(JC-1,J2,J3))
2436 1 JH(I,J2,J3)=JH(I,J2,J3)+1
2437 C
2438 CONTINUE
2439 GO TO 85
2440 C
2441 C NEW RHS
2442 C

```



```

2437 C
2438 XKSI(NROW(JC-1,J2,J3),JC-1,J2,J3)=YTEMP(NCOL(JC-1,J2,J3))
2439 B(NROW(JC-1,J2,J3),JC-1,J2,J3)=YTEMP(NCOL(JC-1,J2,J3))
2440 JCUR=JCT
2441 JPER(2)=J2T
2442 JPER(3)=J3T
2443 DO 3001 I=1,3000
2444 IF(IA(I,1,1).GT.300) STOP
2445 CONTINUE
2446 RETURN
2447 END
2448
2449 C-----
2450 SUBROUTINE LKHDCT(NODE)
2451 C
2452 C ADDS A LOOK-AHEAD CUT TO THE PREVIOUS NODE.
2453 C
2454 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2455 1 INTEGER*4 (I-N,Q)
2456 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2457 INTEGER ICNAM(602,2),NAME(6)
2458 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2459 REAL A(3000,3,3,1)
2460 C
2461 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2462 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YFI(350,3,3,1)
2463 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
2464 2 INVERQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1
2465 2 ),NETA,
2466 3 NLELFM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2467 4 LE(1002),IA(3000,3,3,1),IE(3000),
2468 5 ATMP(3000),ABN(600,3,3,1),IEN(600,3,3,1),LBN(602,3,3,1),
2469 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
2470 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
2471 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2472 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
2473 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),MOUR(3),
2474 2 NXNF(3),INFLG,NETND(10),INST(5),MXNSI,NST
2475 C
2476 C INITIALIZE
2477 C
2478 C
2479 WRITE(6,200)
2480 FORMAT(' MAKING A LOOKAHFAD CUT')
2481 JC=NODE
2482 JCT=JCUR
2483 J2=JPER(2)
2484 J2T=JPER(2)
2485 J2L=JPER(2)
2486 IF(JC.EQ.NPER-1) J2L=1
2487 J3=1
2488 J3T=JPER(3)
2489 IF(NODE.EQ.2) J2=1
2490 KNC=NCOL(JC,J2,J3)+1
2491 DO 10 J=1,300
2492 Y(J)=0.
2493 YTEMP(J)=0.
2494 YTEMP1(J)=0.

```

```

2495      10 CONTINUE
2496      C
2497      C COL BY COL INTO YTEMP
2498      C
2499      IF(JC+1.GE.NPER) NROWP(JC+1,J2L,J3) = NROWP(JC+1,1,1)
2500      JC=JC-1
2501      JND=NROW(JC,J2,J3)+1
2502      IF(NROWP(JC,J2,J3).GE.NROWP(JC+1,J2L,J3)) NRO = NROWP(JC+1,J2L,J3)
2503      IF(NROWP(JC,J2,J3).LT.NROWP(JC+1,J2L,J3)) NRO = NROWP(JC,J2,J3)
2504      JNX = NROW(JC,J2,J3) - NRO
2505      KNC=NCOL(JC,J2,J3)
2506      IF(NTH(JC,J2,J3).EQ.1) KNC=KNC-1
2507      DO 30 J=JND,KNC
2508      JNL=J-JNX
2509      JCUR=JC
2510      JPER(2)=J2
2511      JPER(3)=J3
2512      CALL BUNPCK(JNL)
2513      IF(NODE.LT.NPER) GO TO 19
2514      KNR=NROW(NPER,1,1)
2515      DO 18 L=2,KNR
2516      YTEMP(J)=YTEMP(J)+Y(L)*YPIBAR(L)
2517      18 CONTINUE
2518      GO TO 30
2519      19 CONTINUE
2520      NNDD=NND(JC+1)
2521      DO 25 L=1,NNDD
2522      YTEMP1(J)=O.
2523      IF(JC+1.EQ.2) KNR=NROW(JC+1,L,J3)
2524      IF(JC+1.EQ.3) KNR=NROW(JC+1,J2L,L)
2525      DO 20 K=1,KNR
2526      IF(JC+1.EQ.2) YTEMP1(J)=YTEMP1(J)+YPI(K,JC+1,L,J3)*Y(K)
2527      IF(JC+1.EQ.3) YTEMP1(J)=YTEMP1(J)+YPI(K,JC+1,J2L,L)*Y(K)
2528      20 CONTINUE
2529      IF(JC+1.EQ.2) YTEMP(J)=YTEMP(J)+PROB(JC+1,L,J3)*YTEMP1(J)
2530      IF(JC+1.EQ.3) YTEMP(J)=YTEMP(J)+PROB(JC+1,J2L,L)+YTEMP1(J)
2531      25 CONTINUE
2532      30 CONTINUE
2533      C
2534      C RHS GOES IN TOO
2535      C
2536      IF(NODE.LT.NPER) GO TO 31
2537      YTEMP(NCOL(JC,J2,J3)+1)=XRHO
2538      GO TO 39
2539      31 CONTINUE
2540      DO 35 L=1,NNDD
2541      YTEMP1(NCOL(JC,J2,J3)+1)=O.
2542      IF(JC+1.EQ.2) KNR=NROW(JC+1,L,J3)
2543      IF(JC+1.EQ.3) KNR=NROW(JC+1,J2L,L)
2544      DO 40 K=1,KNR
2545      IF(JC+1.EQ.2) YTEMP1(NCOL(JC,J2,J3)+1)=
2546      YTEMP1(NCOL(JC,J2,J3)+1)+YPI(K,JC+1,L,J3)*XKSI(K,JC+1,L,J3)
2547      IF(JC+1.EQ.3) YTEMP1(NCOL(JC,J2,J3)+1)=
2548      YTEMP1(NCOL(JC,J2,J3)+1)+YPI(K,JC+1,J2L,L)*XKSI(K,JC+1,J2L,L)
2549      40 CONTINUE
2550      IF(JC+1.EQ.2) YTEMP(NCOL(JC,J2,J3)+1)=
2551      YTEMP(NCOL(JC,J2,J3)+1)+PROB(JC+1,L,J3)*YTEMP1(NCOL(JC,J2,J3)+1)
2552

```

```

2553 IF(JC+1.EQ.3) YTEMP(NCOL(JC,J2,J3)+1)=
2554 1 YTEMP(NCOL(JC,J2,J3)+1)+PROB(JC+1,J2L,L)*YTEMP1(NCOL(JC,J2,J3)+1)
2555 CONTINUE
2556 C
2557 C SHIFT A(*,1) DOWN
2558 C FIRST, COPY INTO TEMPS
2559 C
2560 39 CONTINUE
2561 KLM=NELM(JC,J2,J3)
2562 DO 50 I=1,KLM
2563 ATMP(I)=A(I,JC,J2,J3)
2564 ITMP(I)=IA(I,JC,J2,J3)
2565 CONTINUE
2566 50 CONTINUE
2567 C
2568 C ADD COL AND ROW FOR SLACK
2569 C
2570 NCOL(JC,J2,J3)=NCOL(JC,J2,J3)+1
2571 NROW(JC,J2,J3)=NROW(JC,J2,J3)+1
2572 NELM(JC,J2,J3)=NELM(JC,J2,J3)+NCOL(JC,J2,J3)-NROW(JC,J2,J3)+1
2573 JH(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2574 C
2575 C COPY COL VALUES
2576 C
2577 KNCM=NCOL(JC,J2,J3)-1
2578 DO 60 I=1,KNCM
2579 KBTMP(I)=KINBAS(I,JC,J2,J3)
2580 LTMP(I)=LA(I,JC,J2,J3)
2581 XLTMP(I)=XLB(I,JC,J2,J3)
2582 XUTMP(I)=XUB(I,JC,J2,J3)
2583 CONTINUE
2584 60 CONTINUE
2585 C
2586 C UPDATE COLUMNS
2587 C
2588 KINBAS(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2589 LA(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2590 XLB(NROW(JC,J2,J3),JC,J2,J3)=O.
2591 XUB(NROW(JC,J2,J3),JC,J2,J3)=1.E7
2592 JND=O
2593 C
2594 KNRO=NROW(JC,J2,J3)+1
2595 KNC=NCOL(JC,J2,J3)
2596 DO 70 I=KNRO,KNC
2597 JND=JND+1
2598 LA(I,JC,J2,J3)=LTMP(I-1)+JND
2599 KINBAS(I,JC,J2,J3)=KBTMP(I-1)
2600 XLB(I,JC,J2,J3)=XLTMP(I-1)
2601 XUB(I,JC,J2,J3)=XUTMP(I-1)
2602 CONTINUE
2603 70 CONTINUE
2604 LA(NCOL(JC,J2,J3)+1,JC,J2,J3)=NELM(JC,J2,J3)+1
2605 C
2606 A(NROW(JC,J2,J3),JC,J2,J3)=1.
2607 IA(NROW(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2608 C
2609 C UPDATE A AND IA
2610 C
2611 DO 80 I=KNRO,KNC
2612 IND=I
2613 KND=LA(IND,JC,J2,J3)-1

```

```

2611 IF (KND.EQ.NROW(JC,J2,J3))=NELM(JC,J2,J3)-1
2612 IF (DABS(YTEMP(IND-2)) .GE. ZTOLZE) GO TO 75
2613
2614 C SHIFT ALL DOWN
2615 C
2616 NELM(JC,J2,J3)=NELM(JC,J2,J3)-1
2617 KKNC=KNC+1
2618 DO 72 II=IND,KKNC
2619 LA(II,JC,J2,J3)=LA(II,JC,J2,J3)-1
2620 CONTINUE
2621 KLM=NELM(JC,J2,J3)+1
2622 DO 73 II=KND,KLM
2623 A(II,JC,J2,J3)=A(II+1,JC,J2,J3)
2624 IA(II,JC,J2,J3)=IA(II+1,JC,J2,J3)
2625 CONTINUE
2626 GO TO 81
2627
2628 C CONTINUE
2629 A(KND,JC,J2,J3)=YTEMP(IND-2)
2630 IA(KND,JC,J2,J3)=NROW(JC,J2,J3)
2631 MND=LA(IND+1,JC,J2,J3)-2
2632 IF(IND.EQ.NCOL(JC,J2,J3)) MND=NELM(JC,J2,J3)-1
2633 KKND=KND+1
2634 DO 90 J=KKND,MND
2635 JNX=J-IND+NROW(JC,J2,J3)
2636 A(J,JC,J2,J3)=ATMP(JNX)
2637 IA(J,JC,J2,J3)=ITMP(JNX)
2638 CONTINUE
2639 GO TO 80
2640
2641 C CHECK FOR FIRST CUT
2642 IF (NTH(JC,J2,J3).EQ.1) GO TO 811
2643 A(NELM(JC,J2,J3),JC,J2,J3)=YTEMP(NCOL(JC,J2,J3)-1)
2644 IA(NELM(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2645 CONTINUE
2646
2647 C NEW RHS
2648 C
2649 XKSI(NROW(JC,J2,J3),JC,J2,J3)=YTEMP(NCOL(JC,J2,J3))
2650 B(NROW(JC,J2,J3),JC,J2,J3)=YTEMP(NCOL(JC,J2,J3))
2651 C CHECK IF THIS IS THE FIRST TIME
2652 IF (NTH(JC,J2,J3).EQ.1) GO TO 100
2653 NTH(JC,J2,J3)=1
2654 NCOL(JC,J2,J3)=NCOL(JC,J2,J3)+1
2655 NELM(JC,J2,J3)=NELM(JC,J2,J3)+1
2656 A(NELM(JC,J2,J3),JC,J2,J3)=-1
2657 IA(NELM(JC,J2,J3),JC,J2,J3)=1
2658 NELM(JC,J2,J3)=NELM(JC,J2,J3)+1
2659 C ADD THETA
2660 LA(NCOL(JC,J2,J3)+1,JC,J2,J3)=NELM(JC,J2,J3)+1
2661 XUB(NCOL(JC,J2,J3),JC,J2,J3)=1.OE7
2662 XLB(NCOL(JC,J2,J3),JC,J2,J3)=-1.OE7
2663 C MAKE THETA BASIC NOT THE SLACK
2664 KINBAS(NROW(JC,J2,J3),JC,J2,J3)=0
2665 KINBAS(NCOL(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2666 JH(NROW(JC,J2,J3),JC,J2,J3)=NCOL(JC,J2,J3)
2667 C UPDATE THE JH TOO
2668 100 KNRM=NROW(JC,J2,J3)-1

```

```

2669 DO 85 I=1,KNRM
2670 IF (JH(I,JC,J2,J3).GE.NROW(JC,J2,J3))
2671 1 JH(I,JC,J2,J3)=JH(I,JC,J2,J3)+1
2672 CONTINUE
2673 85
2674 C NOW, A "1" FOR THETA.
2675 C
2676 A(NELM(JC,J2,J3),JC,J2,J3)=1.
2677 IA(NELM(JC,J2,J3),JC,J2,J3)=NROW(JC,J2,J3)
2678 JCUR=JCT
2679 JPER(2)=J2T
2680 JPER(3)=J3T
2681 RETURN
2682 END
2683 -----
2684 SUBROUTINE FRMRHS
2685 C
2686 C TAKES B.X AND B TO MAKE A NEW RHS
2687 C
2688 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2689 1 INTEGER*4 (I-N,Q)
2690 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2691 INTEGER ICNAM(602,2),NAME(6)
2692 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2693 REAL A(3000,3,3,1)
2694 C
2695 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2696 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2697 1 .A,E,MSTAT,IOBJ,IROWP,ITCNT,
2698 2 INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1)
2699 2 ),NETA,
2700 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2701 3 602,3,3,1),
2702 4 LE(1002),IA(3000,3,3,1),IE(3000),
2703 5 ATP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
2704 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XITMP(602),
2705 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
2706 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2707 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
2708 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
2709 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2710 JC=JCUR+1
2711 J2=JPER(2)
2712 J3=JPER(3)
2713 IF (JCUR.EQ.NPER-1) J2=1
2714 IF (JCUR.EQ.NPER-1) J3=1
2715 NND=NND(JC)
2716 DO 200 J=1,NND
2717 KNR=NROW(JC,J2,J3)
2718 DO 100 I=2,KNR
2719 IF (JC.EQ.3) B(I,JC,J2,J)=XKSI(I,JC,J2,J)-Y(I)
2720 IF (JC.EQ.2) B(I,JC,J,1)=XKSJ(I,JC,J,1)-Y(I)
2721 100 CONTINUE
2722 200 CONTINUE
2723 RETURN
2724 END
2725 -----
2726 SUBROUTINE BPRODX

```

```

2727 C
2728 C FINDS TH VALUE OF BX FOR THE NEXT PERIOD INVENTORIES
2729 C AND PUTS IT INTO Y
2730 C
2731 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
2732 1 INTEGER*4 (I-N,Q)
2733 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2734 INTEGER ICNAM(602,2),NAME(6)
2735 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2736 REAL A(3000,3,3,1)
2737 C
2738 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2739 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YTI(350,3,3,1)
2740 1 A,E,MSTAT,IOBJ,IROWP,ITCNT,
2741 2 INVFRQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1
2742 ),NETA,
2743 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2744 3 602,3,3,1),
2745 4 LE(1002),IA(3000,3,3,1),IE(3000),
2746 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
2747 C
2748 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
2749 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
2750 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2751 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
2752 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JUSTCH(5,5,5),NCUR(3),
2753 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2754 C
2755 C SET Y'S
2756 C
2757 JCT=JCUR
2758 J2=JPER(2)
2759 J3=JPER(3)
2760 JC=JCUR+1
2761 IF(JC.EQ.NPER)NROWP(JC,J2,J3)=NROWP(JC,1,1)
2762 KNRO=NROW(JC,J2,J3)
2763 DO 10 I=1,KNRO
2764 YTEMP(I)=O.
2765 Y(I)=O.
2766 10 CONTINUE
2767 C
2768 C MULTIPLY BY THE BASIC COLS' VALUES
2769 IF(NROWP(JCT,J2,J3).GE.NROWP(JC,J2,J3)) NRO = NROWP(JC,J2,J3)
2770 IF(NROWP(JCT,J2,J3).LT.NROWP(JC,J2,J3)) NRO = NROWP(JCT,J2,J3)
2771 KNRO=NROW(JCT,J2,J3)+1
2772 KNC=NCOL(JCT,J2,J3)
2773 DO 20 J=KNRO,KNC
2774 JND=J-NROW(JCT,J2,J3)+NRO
2775 IF(JND-NRO.GT.NCOLP(JCT,J2,J3)-NROWP(JCT,J2,J3)) GO TO 20
2776 20 CONTINUE
2777 13 CALL BUNPCK(JND)
2778 IF(KINBAS(J,JCT,J2,J3).EQ.O) XIN=XLB(J,JCT,J2,J3)
2779 IF(KINBAS(J,JCT,J2,J3).EQ.O) GO TO 114
2780 IF(KINBAS(J,JCT,J2,J3).EQ.-1) XIN=XUB(J,JCT,J2,J3)
2781 IF(KINBAS(J,JCT,J2,J3).EQ.-1) GO TO 114
2782 XIN=X(KINBAS(J,JCT,J2,J3),JCT,J2,J3)
2783 114 CONTINUE
2784 KNRD=NROW(JC,J2,J3)

```

```

2785 DO 15 I=1,KNRD
2786 Y(I)=Y(I)*XIN
2787 YTEMP(I)=Y(I)+YTEMP(I)
2788 CONTINUE
2789 CONTINUE
2790 C
2791 DO 25 I=1,KNRD
2792 Y(I)=YTEMP(I)
2793 CONTINUE
2794 RETURN
2795 END
2796 C-----
2797 SUBROUTINE OPTCHK(NODE)
2798 C
2799 C CHECKS FOR MASTER-SUB OPTIMALITY
2800 C
2801 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
2802 1 INTEGER*4 (I-N,Q)
2803 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2804 INTEGER ICNAM(602,2),NAME(6)
2805 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2806 REAL A(3000,3,3,1)
2807 C
2808 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2809 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2810 1 A,E,MSTAT,IOBJ,IRDWP,ITCNT.
2811 2 INVFRQ,ITRFRQ,JCPLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELEM(3,3,1
2812 2 ),NETA,
2813 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2814 3 602,3,3,1),
2815 4 LE(1002),IA(3000,3,3,1),IE(3000),
2816 5 ATPM(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1).
2817
2818 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XIMP(602),
2819 7 LTMP(602),NRDWP(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),IRST(3),
2822 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JUSTCH(5,5,5),PFCUR(3),
2823 2 NXNF(3),INFLG,NETND(10),INST(5),MXNSI,NST
2824 C
2825 C FIRST, GET THE EXPECTED VALUE
2826 C
2827 XT=0.
2828 J2=JPER(2)
2829 NNDD=NND(NODE)
2830 DO 10 J=1,NNDD
2831 IF(NODE.EQ.2) XT=XT+PROB(2,J,1)*X(1,2,J,1)
2832 IF(NODE.EQ.3) XT=XT+PROB(3,J2,J)*X(1,3,J2,J)
2833 CONTINUE
2834 10 CONTINUE
2835 C THEN, CHECK
2836 IF(NODE.EQ.2) JN=KINBAS(NCOL(1,1,1),1,1,1)
2837 IF(NODE.EQ.3) JN=KINBAS(NCOL(2,J2,1),2,J2,1)
2838 IF(JN.LE.0) RETURN
2839 IF((NODE.EQ.2).AND.(XT.GE.X(JN,1,1,1))) NFLG=1
2840 IF((NODE.EQ.3).AND.(XT.GE.X(JN,2,J2,1))) NFLG=1
2841 RETURN
2842 END
C-----

```

```

2843 SUBROUTINE WRAPUP
2844
2845 C OUTPUT OPTIMAL SOLUTION
2846 C
2847     IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
2848     1 INTEGER*4 (I-N,Q)
2849     INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2850     INTEGER ICNAM(602,2,3,3,1)
2851     DOUBLE PRECISION E(3000)
2852     REAL A(3000,3,3,1)
2853     DIMENSION XTEMP(602)
2854     EQUIVALENCE (XTEMP(1),Y(1))
2855 C
2856     COMMON/BL5/ DRES,ICNAM
2857     COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
2858     COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2859     1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2860     1 .A,E,MSTAT,IOBJ,IROWP,ITCNT,
2861     2 INVFRQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELEM(3,3,1)
2862     2 ),NETA,
2863     3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
2864     3 602,3,3,1),
2865     4 LE(1002),IA(3000,3,3,1),IE(3000),
2866     5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
2867
2868     6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XITMP(602),
2869     7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
2870     COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
2871     COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIPAR(602),YBX(350),IBST(3),
2872     1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NUR(3),
2873     2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
2874     IF (ITCNT .LT. ITFRFQ) GO TO 20
2875     WRITE (6,1) ITCNT
2876     1 FORMAT (' SIMPLEX ITERATIONS =',I8,' : COMPUTATIONS TERMINATED.')
2877     20 IF (MSTAT .EQ. QN) GO TO 1000
2878     JPER(2)=1
2879     JPER(3)=1
2880     JCUR=NPER
2881     IN1=INST(1)
2882     IN2=INST(2)
2883     IN3=INST(3)
2884     DO 400 I=1,IN1
2885     DO 400 J=1,IN2
2886     DO 400 K=1,IN3
2887     WRITE(6,401) I,J,K
2888     401 FORMAT('/', ' LAST PERIOD SOLUTIONS FOR FIRST NODE AT SCENARIO',3I6)
2889     WRITE(6,403)
2890     403 FORMAT('/', ' ROW',6X, ' VALUE')
2891     IBASE=JSTCH(I,J,K)
2892     NETA=NETND(IBASE)
2893     XKSI(IBST(1),NPER,1,1)=CBST(1,I)
2894     XKSI(IBST(2),NPER,1,1)=CBST(2,J)
2895     XKSI(IBST(3),NPER,1,1)=CBST(3,K)
2896     JCUR=NPER-1
2897     JPER(NPER-1)=1
2898     CALL BPRODX
2899     CALL FRMRHS
2900     JCUR = NPER

```



```

2901 KNR = NROW(NPER,1,1)
2902 DO 420 IL=1,KNR
2903 Y(IL)=B(IL,NPER,1,1)
2904 CONTINUE
2905 CALL FTRAN(1)
2906 DO 402 JK=1,KNR
2907 WRITE(6,412) ICNAM(JK,1,NPER,1,1),ICNAM(JK,2,NPER,1,1),Y(JK)
2908 FORMAT(2A4,2X,F13.4)
2909 CONTINUE
2910 DO 310 J=1,NPER
2911 NNDD=NND(2)
2912 IF(J.EQ.NPER)NNDD=1
2913 DO 300 K=1,NNDD
2914 NNDDD=NND(3)
2915 IF(J.EQ.NPER)NNDDD=1
2916 DO 295 L=1,NNDDD
2917 NODE=K
2918 JCUR=J
2919 JPER(2)=K
2920 JPER(3)=L
2921 IF((J.EQ.1).AND.(K.NE.1).OR.(L.NE.1))) GO TO 295
2922 IF((J.EQ.2.AND.L.NE.1) GO TO 295
2923 CALL INVERT
2924 WRITE(7,117)
2925 FORMAT(' BASIS')
2926 WRITE(6,17) J,K,L
2927 FORMAT('/', ' PROBLEM NODE = ',I4,'-',I4,'-',I3,/)
2928 WRITE(6,2) X(IOBJ,J,K,L)
2929 FORMAT(' OPTIMUM LP OBJECTIVE VALUE = ',F10.2)
2930 WRITE(6,11) NPASS
2931 FORMAT(' AFTER',I5,' PASSES FROM NODE TO NODE')
2932 WRITE(6,111) ITCNT
2933 FORMAT(' AND AFTER',I5,' ITERATIONS')
2934 WRITE(6,3)
2935 FORMAT(' OPTIMAL VALUE REACHED AT THE POINT')
2936 KNR=NROW(J,K,L)
2937 DO 100 I=1,KNR
2938 XTEMP(JH(I,J,K,L)) = X(I,J,K,L)
2939 WRITE(6,101)
2940 FORMAT('/', ' VARIABLE',10X,'STATUS',5X,'VALUE')
2941 DO 200 JJ=1,KNC
2942 JL=JJ
2943 NRDF=NROW(J,K,L)-NROWP(J,K,L)
2944 IF(JJ.GT.NROWP(J,K,L)) JL=JJ-NRDF
2945 IF(JL.GT.NCOLP(J,K,L)) GO TO 200
2946 IF((JL.LT.JJ).AND.(JL.LE.NROWP(J,K,L))) GO TO 200
2947 IF (KINBAS(JJ,J,K,L).GT.O) GO TO 151
2948 IF (KINBAS(JJ,J,K,L).EQ.O) GO TO 150
2949 XTEMP(JJ) = XUB(JJ,J,K,L)
2950 GO TO 151
2951 XTEMP(JJ) = XLB(JJ,J,K,L)
2952 CONTINUE
2953 KB=KINBAS(JJ,J,K,L)
2954 WRITE(6,102) ICNAM(JL,1,J,K,L),ICNAM(JL,2,J,K,L),KB,XTEMP(JJ)
2955 FORMAT(2A4,12X,I4,5X,F12.4)
2956 IF(KB.LE.O) GO TO 200
2957
2958

```

```

2959 IF(KB.GT.NROWP(J,K,L)) GO TO 200
2960 WRITE(7,118) ICNAM(JL,1,J,K,L),ICNAM(JL,2,J,K,L),ICNAM(KB,1,J,K,L)
2961 1 ,ICNAM(KB,2,J,K,L)
2962 118 FORMAT(4X,2A4,2X,2A4)
2963 200 CONTINUE
2964 CALL FORMC
2965 CALL BTRAN
2966 WRITE(6,6)
2967 FORMAT(' OPTIMAL DUAL VALUES')
2968 WRITE(6,4) (Y(JJ),JJ=1,KNR)
2969 4 FORMAT(8F9.2)
2970 CONTINUE
2971 295 CONTINUE
2972 300 CONTINUE
2973 310 CONTINUE
2974 RETURN
2975 1000 WRITE (6,5)
2976 5 FORMAT (' NO FEASIBLE SOLUTION FOUND. ')
2977 JCUR=1
2978 JPER(2)=1
2979 JPER(3)=1
2980 CALL STRPRT
2981 RETURN
2982 END
2983 C-----
2984 SUBROUTINE INIT
2985 C
2986 C
2987 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
2988 1 INTEGER*4 (I-N,Q)
2989 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
2990 INTEGER ICNAM(602,2),NAME(6)
2991 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
2992 REAL A(3000,3,3,1)
2993 C
2994 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
2995 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QG,QS,QP
2996 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
2997 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
2998 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT
2999 2 INVERQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
3000 ),NETA,
3001 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
3002 3 602,3,3,1),
3003 4 LE(1002),IA(3000,3,3,1),IE(3000),
3004 5 ATPM(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
3005 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
3006 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
3007 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
3008 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
3009 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),MCUR(3),
3010 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
3011 C
3012 DO 40 I=1,3000
3013 DO 30 J=1,3
3014 DO 20 K=1,2
3015 DO 10 L=1,2
3016 A(I,J,K,L)=0.0

```

```

3017 IA(I,J,K,L)=0
3018 IF(I .GT. 500) GO TO 10
3019 ABN(I,J,K,L)=0.0
3020 IBN(I,J,K,L)=0.0
3021 IF(I .GT. 522) GO TO 10
3022 XLB(I,J,K,L)=0.0
3023 XUB(I,J,K,L)=0.0
3024 KINBAS(I,J,K,L)=0
3025 LA(I,J,K,L)=0
3026 LBN(I,J,K,L)=0
3027 IF(I .GT. 300) GO TO 10
3028 XKSI(I,J,K,L)=0.0
3029 YPI(I,J,K,L)=0.0
3030 JH(I,J,K,L)=0
3031 X(I,J,K,L)=0.0
3032 B(I,J,K,L)=0.0
3033 CONTINUE
3034 CONTINUE
3035 CONTINUE
3036 CONTINUE
3037 DO 50 I=1,600
3038 ATMP(I)=0.0
3039 ITMP(I)=0
3040 IF(I .GT. 122 ) GO TO 50
3041 YTEMP(I)=0.0
3042 YTEMP1(I)=0.0
3043 KBTMP(I)=0
3044 XUTMP(I)=0.0
3045 XLTMP(I)=0.0
3046 LTMP(I)=0
3047 IF ( I .GT. 60) GO TO 50
3048 Y(350)=0.0
3049 CONTINUE
3050 DO 80 I=1,3
3051 DO 70 J=1,2
3052 DO 60 L=1,2
3053 NROW(I,J,L)=0
3054 NCOL(I,J,L)=0
3055 NELM(I,J,L)=0
3056 PROB(I,J,L)=0.0
3057 NROWP(I,J,L)=0
3058 NCOLP(I,J,L)=0
3059 NTH(I,J,L)=0
3060 CONTINUE
3061 CONTINUE
3062 CONTINUE
3063 DO 11 I=1,3000
3064 IE(I)=0
3065 E(I)=0.0
3066 CONTINUE
3067 DO 12 I=1,1002
3068 LE(I)=0
3069 CONTINUE
3070 DO 17 I=1,5
3071 NND(I)=0
3072 JPER(I)=0
3073 CONTINUE
3074 DE=0.0

```

```

3075 DP=0.0
3076 MSTAT=0
3077 IOBJ=0
3078 IROWP=0
3079 ITCNT=0
3080 INVFRQ=0
3081 ITRFRQ=0
3082 JCOLP=0
3083 NETA=0
3084 NLELEM=0
3085 NLETA=0
3086 NUELEM=0
3087 NUETA=0
3088 NELEM=0
3089 NR=0
3090 NPASS=0
3091 JCUR=0
3092 JPASS=0
3093 NPER=0
3094 NFLG=0
3095 ATEMP1=0.0
3096 ATEMP2=0.0
3097 NODE=0
3098 DO 13 I=1,6
3099 NAME(I)=0
3100 CONTINUE
3101 DO 14 I=1,122
3102 DO 15 J=1,2
3103 ICNAM(I,J)=0
3104 CONTINUE
3105 CONTINUE
3106 RETURN
3107 END
3108
3109 -----
3110 SUBROUTINE XOPTCK
3111
3112 C
3113 IMPLICIT REAL*4 (A,C,E,H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
3114 1 INTEGER*4 (I-N,Q)
3115 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
3116 INTEGER ICNAM(602,2),NAME(6)
3117 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
3118 REAL A(3000,3,3,1)
3119
3120 C
3121 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
3122 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QG,QS,QP
3123 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
3124 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
3125 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
3126 2 INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1
3127 2 ),NETA,
3128 3 NLELEM,NETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
3129 3 602,3,3,1),
3130 4 LE(1002),IA(3000,3,3,1),IE(3000),
3131 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
3132 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLTMP(602),
3133 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
3134 COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER

```

```

3133 COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
3134 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCRUR(3),
3135 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
3136
3137 C
3138 J2=JPER(2)
3139 J3=JPER(3)
3140 JN=KINBAS(NCOL(NPER-1,J2,J3),NPER-1,J2,J3)
3141 IF(XTOPT.LT.(X(JN,NPER-1,J2,J3)*(1/1.01))) NFLG=0
3142 C CHANGE IF NEGATIVE
3143 IF(XTOPT.GT.0) GO TO 12
3144 NFLG = 1
3145 IF(XTOPT.LT.(X(JN,NPER-1,J2,J3)*(1/.99))) NFLG = 0
3146
3147 12 CONTINUE
3148 XQX=-XTOPT
3149 WRITE(6,10) XQX
3150 FORMAT(' Q(X) =',E11.3)
3151 XTH=-X(JN,NPER-1,J2,J3)
3152 WRITE(6,11) XTH
3153 FORMAT(' THETA=',E11.3)
3154 C CHECK IF THIS IS THE FIRST TIME
3155 IF(NTH(NPER-1,J2,J3).EQ.0) NFLG=0
3156 IF(NTH(NPER-1,J2,J3).EQ.0) RETURN
3157
3158 C FIND BOUNDS
3159
3160 IF(NPER.NE.2) RETURN
3161 X00=-X(1,1,1,1)
3162 WRITE(6,1) X00
3163 FORMAT(' LOWER BOUND=',E11.3)
3164 XC=-X(1,1,1,1)+X(JN,1,1,1)-XTOPT
3165 WRITE(6,2) XC
3166 FORMAT('/', ' UPPER BOUND=',E11.3)
3167 RETURN
3168 END
3169 C-----
3170 SUBROUTINE DNORML
3171
3172 C
3173 IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y),
3174 1 INTEGER*4 (I-N,Q)
3175 INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
3176 INTEGER ICNAM(602,2),NAME(6)
3177 DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
3178 REAL A(3000,3,3,1)
3179
3180 COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPTYPE,CMIN,CMAX,AFV,NINF,NOPT
3181 COMMON/BL5/DRES,ICNAM
3182 COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
3183 1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,OS,OP
3184 COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
3185 1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKST(350,3,3,1),YPI(350,3,3,1)
3186 1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
3187 2 INVERQ,ITFRFQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFIM(3,3,1
3188 2 ),NETA,
3189 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
3190 3 602,3,3,1),
3191 4 LE(1002),IA(3000,3,3,1),IE(3000),
3192 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),

```

```

3191 6  PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XITMP(602),
3192 7  LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),MTH(3,3,1),NR
3193  COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
3194  COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIEAR(602),YBX(350),IBST(3),
3195  1  PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NUR(3),
3196  2  NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
3197  COMMON/BLS/ ITSIN
3198
3199  JC=JCUR
3200  J2=JPER(2)
3201  J3=JPER(3)
3202  CMIN=O.
3203  NINF=O
3204  NOPT=O
3205  MSTAT=QF
3206  WRITE(6,1001)
3207  1001 FORMAT(' ITNS',2X,'RRES',5X,' PIV',6X,' IN',3X,' OUT',2X,
3208  1  'OBJ',7X,'CMIN',7X,'MXINF',7X,'THET',3X,'NINF',1X,'NOPT')
3209  IF(ITSIN.LT. INVFRQ) GO TO 1500
3210  1000 CONTINUE
3211  CALL INVERT
3212  WRITE(6,1003) ITSIN
3213  1003 FORMAT(' WARNING -- TOO MANY ETAS IN DNRML',1X,' ITFPS =',16)
3214  C  ITSIN=O
3215  C
3216  C  DUAL SIMPLEX CYCLE
3217  C
3218  1500 CALL DCHUZR
3219  C
3220  C  IROWP=O IF OPTIMAL
3221  C
3222  IF(IROWP.EQ.O) GO TO 6000
3223  C
3224  C  FIND PIVOT COL. - JCOLP=O IF INFEASIBLE
3225  C
3226  CALL DCHUZC
3227  IF(JCOLP.EQ.O) MSTAT=QN
3228  IF(JCOLP.EQ.O) GO TO 6000
3229  CALL UPBETA
3230  ITCNT=ITCNT+1
3231  ITSIN=ITSIN+1
3232  IF(NPIVOT.EQ.O) GO TO 4010
3233  IF(NELEM.GT.(NEMAX-NROW(JCUR,JPER(2),JPER(3)))) GO TO 1000
3234  CMAX=YTEMP1(1)
3235  APV=YTEMP1(2)
3236  CALL WRETA
3237  CALL RHCHK
3238  WRITE(6,1002)ITCNT,DRES,APV,JCOLP,IROWP,X(1,JC,J2,J3).CMIN.CMAX
3239  1  ,DP,NINF,NOPT
3240  1002 FORMAT(I4,1X,E9.2,1X,E9.2,1X,I4,1X,I4,1X,I4,1X,E10.2,1X,E9.2,1X,E9.2,1X,
3241  1  E9.2,1X,I4,1X,I4)
3242  4010 IF(ITSIN.GE. INVFRQ) GO TO 1000
3243  IF(ITCNT.GE. ITRFRQ) GO TO 6000
3244  GO TO 1500
3245  6000 RETURN
3246  END
3247  C-----
3248  SUBROUTINE PARSFT

```

```

3249 C
3250 C THIS ROUTINE FINDS ALL THE BASES THAT WILL MAKE
3251 C THE LAST PERIOD OPTIMAL.
3252 C
3253 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y);
3254 C 1 INTEGER*4 (I-N,Q)
3255 C INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
3256 C INTEGER ICNAM(602,2),NAME(6)
3257 C DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
3258 C REAL A(3000,3,3,1)
3259 C
3260 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAX,NRMAX,
3261 C NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QQ,QS,QP
3262 C COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
3263 C 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
3264 C 1 .A.E.MSTAT,IOBJ,IROWP,ITCNT.
3265 C 2 INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELM(3,3,1
3266 C ),NETA.
3267 C 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
3268 C 602,3,3,1).
3269 C 4 LE(1002),IA(3000,3,3,1),IE(3000).
3270 C 5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
3271 C
3272 C 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
3273 C LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
3274 C COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
3275 C COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
3276 C 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NCUR(3),
3277 C 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
3278 C COMMON/BLS/ITSIN
3279 C
3280 C SET JPASS
3281 C
3282 C JPASS=2
3283 C
3284 C SET ALL THE PROPER VALUES
3285 C
3286 C 'IBASE' IS THE CURRENT BASIS
3287 C 'ITSIN' NUMBER OF ITERATIONS FOR DN
3288 C
3289 C IBASE=1
3290 C ITSIN=0
3291 C
3292 C SET JPER FOR CHECKING
3293 C JPOLD=JPER(NPER-1)
3294 C JPER(NPER-1)=1
3295 C 'XRHO' IS THE CURRENT EX VAL OF THE RHS
3296 C 'XTOPT' IS EX VAL OF Z
3297 C
3298 C XRHO=0
3299 C XTOPT=0
3300 C
3301 C
3302 C 'YPIBAR' KEEPS THE CURRENT EX VAL OF THE PI VECTOR
3303 C
3304 C KNR=NROW(NPER,1,1)
3305 C DO 10 I=1,KNR
3306 C YPIBAR(I)=0
3307 C
3308 C 10 CONTINUE

```

```

3307 C          'JSTCH(I,J,K,L)' HAS 0 IF ALTERNATIVE (I,J,K,L)
3308 C          HAS NOT YET BEEN ASSIGNED A BASIS.
3309 C
3310 C
3311 C          DO 11 I=1,MXNST
3312 C          DO 11 J=1,MXNST
3313 C          DO 11 K=1,MXNST
3314 C          JSTCH(I,J,K)=0
3315 C          11 CONTINUE
3316 C
3317 C          'NCUR' KEEPS THE CURRENT ALTERNATIVE
3318 C          'NXNF' KEEPS THE INDEX OF THE FIRST INFEASIBLE ALTERNATIVE.
3319 C
3320 C          DO 12 I=1,3
3321 C          NCUR(I)=1
3322 C          NXNF(I)=1
3323 C          12 CONTINUE
3324 C
3325 C          'NETND(I)' KEEPS THE INDEX OF THE ETA VECTOR FOR
3326 C          BASIS 'I'.
3327 C
3328 C          DO 13 I=1,3
3329 C          NETND(I)=NETA
3330 C          13 CONTINUE
3331 C
3332 C          'INFLG' EQUAL 0 SAYS THERE ARE NO INFEASIBILITIES CAUGHT YET.
3333 C          INFLG=0
3334 C
3335 C          'BND(I)' HAS THE BASELINE VALUES FOR THE RHS'S IN NPER.
3336 C
3337 C          KNR=NROW(NPER,1,1)
3338 C          DO 14 I=1,KNR
3339 C          XKSI(I,NPER,1,1)=BND(I)
3340 C          14 CONTINUE
3341 C          SET FOR INITIAL RHS
3342 C          DO 1014 I=1,NST
3343 C          XKSI(IBST(I),NPER,1,1)=CBST(I,NCUR(I))
3344 C          1014 CONTINUE
3345 C
3346 C          ADD FOR RHS
3347 C
3348 C          JCUR=JCUR-1
3349 C          JPER(NPER-1)=JPOLD
3350 C          CALL BPRODX
3351 C          JPER(NPER-1)=1
3352 C
3353 C          YBX KEEPS INVENTORIES
3354 C
3355 C          DO 27 I=2,KNR
3356 C          YBX(I)=Y(I)
3357 C          27 CONTINUE
3358 C
3359 C          START OF CYCLE
3360 C
3361 C          20 CONTINUE
3362 C          DO 28 I=2,KNR
3363 C          Y(I)=YBX(I)
3364 C          28 CONTINUE

```



```

3365 JCUR = NPER-1
3366 CALL FRMRHS
3367 JCUR = NPER
3368
3369 C IF 'IBASE'=1 FIND THE FIRST OPTIMAL
3370 C
3371
3372 IF(IBASE .GT. 1) GO TO 21
3373 SET JCUR BACK TO NPER FOR CHECKING
3374 JCUR = NPER
3375 ITSINU=9999
3376 CALL NORMAL(ITSINU)
3377 IF(JCUR.EQ.NPER+1) RETURN
3378 GO TO 211
3379
3380 21 CONTINUE
3381 WRITE(6,2201)
3382 FORMAT(' ITERATING IN DNORML')
3383 DO 1053 I=1,KNR
3384 Y(I)=B(I,NPER,1,1)
3385 1053 CONTINUE
3386 CALL FTRAN(1)
3387
3388 C COPY INTO X.
3389 C
3390 C
3391 DO 1054 I=1,KNR
3392 X(I,NPER,1,1)=Y(I)
3393 1054 CONTINUE
3394 CALL DNORML
3395 211 CONTINUE
3396 CALL STPRPT
3397 IF(MSTAT .NE. QN) GO TO 22
3398
3399 C IF INFEASIBLE ADD A FEASCT AND RETURN.
3400 C
3401
3402 WRITE(6,2022) (NCUR(I),I=1,NST)
3403 FORMAT(' INFEAS AT PROB',3I6)
3404 JPER(NPER-1)=JPOLD
3405 CALL FEASCT
3406 JCUR=NPER-1
3407 RETURN
3408 22 CONTINUE
3409
3410 C HERE IT'S FEASIBLE, SET FOR FIRST OF NEW BASIS.
3411 C
3412
3413 NETND(IBASE)=NETA
3414 JSTCH(NCUR(1),NCUR(2),NCUR(3))=IBASE
3415
3416 C UPDATE EX VAL OF RHS
3417 C
3418
3419 KNR=NROW(NPER,1,1)
3420 XXR=O.
3421 DO 25 I=2,KNR
3422 XXR=YPI(I,NPER,1,1)*XKSI(I,NPER,1,1)+XXR
3423 25 CONTINUE
3424 WRITE(6,2024)
3425 FORMAT(' VAR1',2X,' VAR2',2X,' VAR3',2X,' ORJ',10X,
3426 1 ' KSIPI',6X,'OBJTOT',5X,'CURR BAS')
3427 WRITE(6,2025) (NCUR(I),I=1,3),PRST(NCUR(1),NCUR(2),NCUR(3)),
3428 1 X(1,NPER,1,1),XXR,XTOPT,IBASE
3429

```

```
3423      2025      FORMAT(3I6,1X,F8.6,1X,E11.3,1X,E11.3,1X,E11.3,1X,16)
3424      XRH0=XXR*PRST(NCUR(1),NCUR(2),NCUR(3))*XRIO
3425      C
3426      C          UPDATE EX VAL OF PI'S
3427      C
3428      DO 26 I=2,KNR
3429      YPIBAR(I)=YPI(I,NPER,1,1)*PRST(NCUR(1),NCUR(2),NCUR(3))*YPIBAR(I)
3430      26 CONTINUE
3431      C
3432      C          UPDATE E.V. OF OBJECTIVE
3433      C
3434      XTOPT=XTOPT+PRST(NCUR(1),NCUR(2),NCUR(3))*X(1,NPER,1,1)
3435      C
3436      C          BEGIN LOOP TO CHECK FEASIBILITY M IS
3437      C          THE CURRENT STOCHASTIC VARIABLE.
3438      C
3439      M=NST
3440      29 CONTINUE
3441      IF(NCUR(M).LT.INST(M)) GO TO 30
3442      M=M-1
3443      IF(M.GT.0) GO TO 29
3444      IF(INFLG.NE.0) GO TO 31
3445      C
3446      C          HERE, ALL NODES ARE COVERED.
3447      C
3448      NFLAG=1
3449      JPER(NPER-1)=JPOLD
3450      CALL XOPTCK
3451      IF(NFLAG.EQ.1) GO TO 32
3452      C PUT YBAR IN FOR PI
3453      DO 319 I=2,KNR
3454      YPI(I,NPER,JPER(2),1)=YPIBAR(I)
3455      319 CONTINUE
3456      C CALL STRPT
3457      CALL LKHDCT(NPER)
3458      32 CONTINUE
3459      JCUR=NPER-1
3460      IF((NPER.EQ.2).AND.(NFLAG.EQ.1)) JCUR= NPER + 1
3461      RETURN
3462      C
3463      C          THERE ARE STILL INFEAS'S LEFT.
3464      C
3465      31 CONTINUE
3466      C
3467      C          RETURN TO SOLVE NEXT PROBLEM.
3468      C
3469      WRITE(6,2301)
3470      FORMAT(' RETURNING TO SOLVE NEXT BATCH ')
3471      DO 40 I=1,NST
3472      XKSI(1BST(I),NPER,1,1)=CBST(I,NXNF(I))
3473      40 CONTINUE
3474      IBASE=IBASE+1
3475      INFLAG=0
3476      DO 41 I=1,NST
3477      NCUR(I)=NXNF(I)
3478      41 CONTINUE
3479      GO TO 20
3480      C
```

```

3481 C          HERE, WE CHECK OUT THE NEXT POSSIBILITY.
3482 C
3483 C          30  NCUR(M)=NCUR(M)+1
3484 C          MM=M+1
3485 C          IF(MM.GT.NST) GO TO 43
3486 C          DO 42 I=MM,NST
3487 C          NCUR(I)=1
3488 C          42  CONTINUE
3489 C          43  CONTINUE
3490 C          SET M BACK TO THE LAST PLAC
3491 C          M=NST
3492 C
3493 C          CHECK IF THIS IS COVERED.
3494 C
3495 C          IF(JSTCH(NCUR(1),NCUR(2),NCUR(3)) .EQ. 0) GO TO 50
3496 C
3497 C          THERE, JSTCH WAS COVERED.
3498 C
3499 C          GO TO 29
3500 C
3501 C          HERE, NOT COVERED, CHECK FEASIBILITY.
3502 C
3503 C          50  CONTINUE
3504 C          DO 51 I=1,NST
3505 C          XKSI(IBST(I),NPER,1,1)=CBST(I,NCUR(I))
3506 C          51  CONTINUE
3507 C          DO 52 I=1,KNR
3508 C          Y(I)=YBX(I)
3509 C          52  CONTINUE
3510 C
3511 C          SET-UP RHS FOR CHECK.
3512 C
3513 C          JCUR FOR FRMRHS
3514 C          JCUR=NPER-1
3515 C          CALL FRMRHS
3516 C          JCUR=NPER
3517 C          DO 53 I=1,KNR
3518 C          Y(I)=B(I,NPER,1,1)
3519 C          53  CONTINUE
3520 C          CALL FTRAN(1)
3521 C
3522 C          COPY INTO X.
3523 C
3524 C          DO 54 I=1,KNR
3525 C          X(I,NPER,1,1)=Y(I)
3526 C          54  CONTINUE
3527 C          WRITE(6,2054)
3528 C          FORMAT(' CHECKING FEAS WITH DCHUZR')
3529 C          CALL DCHUZR
3530 C          IF(IROWP .EQ. 0) GO TO 60
3531 C
3532 C          HERE, THE NODE IS OPTIMAL FOR THE GIVEN BASIS.
3533 C          MUST CHECK FEASIBILITY.
3534 C
3535 C
3536 C          WRITE(6,2055)
3537 C          FORMAT(' TRYING TO FIND ENTERING IN DCHUZC')
3538 C          CALL DCHUZC

```

```

3539 IF(JCOLP. NE. O) GO TO 59
3540
3541 C HERE INFEASIBLE.
3542 C
3543 WRITE(6,2056)
3544 FORMAT(' NON ENTERING VAR - INFEAS')
3545 JPER(NPER-1)=JPOLD
3546 CALL STRPRT
3547 CALL FEASCT
3548 JCUR=NPER-1
3549 RETURN
3550 C
3551 C THERE, FEASIBLE
3552 C
3553 59 CONTINUE
3554 WRITE(6,2059) (NCUR(I),I=1,3)
3555 FORMAT(' ENT VAR - WILL RETURN TO',3I6)
3556 IF(INFLG.NE. O) GO TO 29
3557 C
3558 C FIRST INFEASIBLE FOUND.
3559 C
3560 INFLG=1
3561 DO 58 I=1,NST
3562 NXNF(I)=NCUR(I)
3563 58 CONTINUE
3564 GO TO 29
3565 C
3566 C THERE, IT IS FEASIBLE
3567 C
3568 60 CONTINUE
3569 WRITE(6,2060)
3570 FORMAT(' OK - FEAS')
3571 C
3572 C UPDATE
3573 C
3574 KNR=NROW(NPER,1,1)
3575 XNR=O.
3576 DO 255 I=2,KNR
3577 XNR=YPI(I,NPER,1,1)*XKSI(I,NPER,1,1)+XNR
3578 255 CONTINUE
3579 WRITE(6,2025) (NCUR(I),I=1,3),PRST(NCUR(1),NCUR(2),NCUR(3)),
3580 1 X(1,NPER,1,1),XNR,XTOPT,IBASE
3581 XRHO=XNR*PRST(NCUR(1),NCUR(2),NCUR(3))+XRIID
3582 XTOPT=XTOPT+PRST(NCUR(1),NCUR(2),NCUR(3))*X(1,NPER,1,1)
3583 C
3584 C UPDATE PRICES.
3585 C
3586 DO 61 I=1,KNR
3587 YPIBAR(I)=YPI(I,NPER,1,1)*PRST(NCUR(1),NCUR(2),NCUR(3))+YPIBAR(I)
3588 61 CONTINUE
3589 C
3590 C SET BASIS INDEX.
3591 C
3592 JSTCH(NCUR(1),NCUR(2),NCUR(3))=IBASE
3593 GO TO 29
3594 END
3595 C-----
3596 SUBROUTINE DCHUZR

```

```

3597 C
3598 C SELECTS PIVOT ROW IROWP FOR CURRENT DUAL-SIMPLEX ITERATION.
3599 C SETS IROWP=0 IF CURRENT BASIS IS OPTIMAL. OTHERWISE, IROWP IS
3600 C CHOSEN TO BE THE ROW WITH GREATEST PRIMAL INFEASIBILITY.
3601 C IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
3602 C 1 INTEGER*4 (I-N,Q)
3603 C INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
3604 C INTEGER ICNAM(602,2),NAME(6)
3605 C DOUBLE PRECISION E(3000).ATEMP1,ATEMP2
3606 C REAL A(3000,3,3,1)
3607 C
3608 C COMMON/BLOCK2/ ICOL,IVAL,IDIR,NPIVOT,IPIYPE,CMIN,CMAX,APV,MINF,NOPT
3609 C COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOI SM,NEGINF,NEMAX,NRMAX,
3610 C 1 NTMAX,OBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QQ,QR,QM,QQ,QS,QP
3611 C COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6)
3612 C 1 22) XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YPI(350,3,3,1)
3613 C 1 .A,E,MSTAT,IOBJ,IROWP,ITCNT,
3614 C 2 INVFREQ,ITFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NFLM(3,3,1)
3615 C 2 ).NETA,
3616 C 3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
3617 C 3 602,3,3,1),
3618 C 4 LE(1002),IA(3000,3,3,1),IE(3000),
3619 C 5 ATEMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
3620 C
3621 C 6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
3622 C 7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
3623 C COMMON/BLOCK3/ NND(5),NPASS,JPER(5),JCUR,JPASS,NPER
3624 C COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIRAR(602),YBX(350),IBST(3),
3625 C 1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),NOUR(3),
3626 C 2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
3627 C
3628 C NINF=0
3629 C IROWP = 0
3630 C DP = -1.E10
3631 C KNR=NROW(NPER,1,1)
3632 C DO 1000 I=1,KNR
3633 C IF (I.EQ. IOBJ) GO TO 1000
3634 C ICOL = JH(I,NPER,1,1)
3635 C IF (X(I,NPER,1,1) .LT. (XLB(ICOL,NPER,1,1) - ZTOLZE)) GO TO 100
3636 C IF (X(I,NPER,1,1) .GT. (XUB(ICOL,NPER,1,1) + ZTOLZE)) GO TO 200
3637 C GO TO 1000
3638 C
3639 C BASIC VARIABLE ON ROW I FALLS BELOW ITS LOWER BOUND)
3640 C DE = XLB(ICOL,NPER,1,1) - X(I,NPER,1,1)
3641 C NINF=NINF+1
3642 C IF (DE .LE. DP) GO TO 1000
3643 C IPTYPE = 0
3644 C GO TO 250
3645 C
3646 C BASIC VARIABLE ON ROW I EXCEEDS ITS UPPER BOUND
3647 C DE = X(I,NPER,1,1) - XUB(ICOL,NPER,1,1)
3648 C NINF=NINF+1
3649 C IF (DE .LE. DP) GO TO 1000
3650 C IPTYPE = -1
3651 C
3652 C IROWP = I
3653 C DP = DE
3654 C YTEMP1(1)=DP

```

```

3655      1000      CONTINUE
3656      RETURN
3657      END
3658      C-----
3659      SUBROUTINE DCHUZC
3660
3661      C      SELECTS PIVOT COLUMN JCOLP FOR CURRENT DUAL-SIMPLEX ITERATION.
3662      C      SETS JCOLP=0 IF LP-PROBLEM AT CURRENT NODE IS INFEASIBLE.
3663      C      OTHERWISE CHOOSES JCOLP TO MAINTAIN PRIMAL-OPTIMALITY.
3664
3665      C      IMPLICIT REAL*4 (A,C,E-H,O,P,R-W,Z), REAL*8 (B,D,X,Y).
3666      C      INTEGER*4 (I-N,Q)
3667      C      INTEGER JH,KINBAS,LA,LE,IA,IE,NODE
3668      C      INTEGER ICNAM(602,2),NAME(6)
3669      C      DOUBLE PRECISION E(3000),ATEMP1,ATEMP2
3670      C      REAL A(3000,3,3,1)
3671
3672      C      COMMON/BLOCK2/ ICOL,IVAL,IDIR,IPRIVOT,IPTYPE,CMIN,CMAX,APV,NINF,NOPT
3673      C      COMMON/BLOCK/ ZTOLZE,ZTOLPV,ZTCOST,ZTOLSM,NEGINF,NEMAY,NRMAX,
3674      C      1 NTMAX,QBL,QA,QI,QF,QN,QB,QC,QE,QH,QL,QO,QR,QM,QQ,QS,QP
3675      C      COMMON DE,DP,B(350,3,3,1),X(350,3,3,1),Y(350),YTEMP(602),YTEMP1(6
3676      C      1 22),XLB(602,3,3,1),XUB(602,3,3,1),XKSI(350,3,3,1),YF1(350,3,3,1)
3677      C      1 ,A,E,MSTAT,IOBJ,IROWP,ITCNT,
3678      C      2 INVFRQ,ITRFRQ,JCOLP,NROW(3,3,1),NCOL(3,3,1),NELEM,NELEM(3,3,1
3679      C      2 ),NETA,
3680      C      3 NLELEM,NLETA,NUELEM,NUETA,JH(350,3,3,1),KINBAS(602,3,3,1),LA(
3681      C      3 602,3,3,1),
3682      C      4 LE(1002),IA(3000,3,3,1),IE(3000),
3683      C      5 ATMP(3000),ABN(600,3,3,1),IBN(600,3,3,1),LBN(602,3,3,1),
3684
3685      C      6 PROB(3,3,1),NFLG,KBTMP(602),ITMP(3000),XUTMP(602),XLIMP(602),
3686      C      7 LTMP(602),NROWP(3,3,1),NCOLP(3,3,1),NTH(3,3,1),NR
3687      C      COMMON/BLOCK3/ NND(5),NPASS,JP(5),JCUR,JPASS,NPER
3688      C      COMMON/BLOCK4/ BND(350),XTOPT,XRHO,YPIBAR(602),YBX(350),IBST(3),
3689      C      1 PRBV(3,5),PRST(5,5,5),CBST(3,5),IBASE,JSTCH(5,5,5),MUR(3),
3690      C      2 NXNF(3),INFLG,NETND(10),INST(5),MXNST,NST
3691
3692      C      JCOLP = 0
3693      C      IF (IPTYPE .EQ. -1) GO TO 1000
3694
3695      C      LEAVING VARIABLE FALLS BELOW ITS LOWER BOUND
3696
3697      C      DP = -1.E10
3698      C      KNC=NCOL(NPER,1,1)
3699      C      DO 500 J=1,KNC
3700      C      IF (KINBAS(J,NPER,1,1) .GT. 0) GO TO 500
3701      C      IF ((XUB(J,NPER,1,1) - XLB(J,NPER,1,1)) .LE. ZTOLZE) GO TO 500
3702      C      K = J
3703      C      CALL UNPACK(K)
3704      C      CALL FTRAN(1)
3705      C      IF (KINBAS(J,NPER,1,1) .EQ. -1) GO TO 200
3706      C      IF (Y(IROWP) + ZTOLPV) 225,225,500
3707      C      IF (Y(IROWP) - ZTOLPV) 500,225,225
3708      C
3709      C      200      DE = Y(IOBJ)/Y(IROWP)
3710      C      IF (DE - DP) 500,500,250
3711      C      JCOLP = J
3712      C      DP = DE

```

```

3713 YTEMP1(2)=Y(IROWP)
3714 CONTINUE
3715 C
3716 IF (JCOLP .EQ. 0) RETURN
3717 CALL UNPACK(JCOLP)
3718 CALL FTRAN(1)
3719 ICOL = JH(IROWP,NPER,1,1)
3720 DP = (X(IROWP,NPER,1,1) - XLB(ICOL,NPER,1,1))/Y(IROWP)
3721 GO TO 2000
3722 C
3723 LEAVING VARIABLE EXCEEDS ITS UPPER BOUND
3724 C
3725 DP = 1.E10
3726 KNC=NCOL(NPER,1,1)
3727 DO 1500 J=1,KNC
3728 IF (KINBAS(J,NPER,1,1) .GT. 0) GO TO 1500
3729 IF ((XUB(J,NPER,1,1) - XLB(J,NPER,1,1)) .LE. ZTOLZF) GO TO 1500
3730 K = J
3731 CALL UNPACK(K)
3732 CALL FTRAN(1)
3733 IF (KINBAS(J,NPER,1,1) .EQ. -1) GO TO 1200
3734 IF (Y(IROWP) - ZTOLPV) 1500,1225,1225
3735 IF (Y(IROWP) + ZTOLPV) 1225,1225,1500
3736 C
3737 DE = Y(IOBJ)/Y(IROWP)
3738 IF (DE - DP) 1250,1500,1500
3739 JCOLP = J
3740 DP = DE
3741 YTEMP1(2)=Y(IROWP)
3742 CONTINUE
3743 C
3744 IF (JCOLP .EQ. 0) RETURN
3745 CALL UNPACK(JCOLP)
3746 CALL FTRAN(1)
3747 ICOL = JH(IROWP,NPER,1,1)
3748 DP = (X(IROWP,NPER,1,1) - XUB(ICOL,NPER,1,1))/Y(IROWP)
3749 C
3750 DE = DP + XLB(JCOLP,NPER,1,1)
3751 IF (KINBAS(JCOLP,NPER,1,1) .EQ. -1) DE = DP + XUB(JCOLP,NPER,1,1)
3752 NPivot = 1
3753 RETURN
3754 END

```