INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-174

DISOPT3 - A USER-ORIENTED PACKAGE FOR NONLINEAR CONTINUOUS AND DISCRETE OPTIMIZATION PROBLEMS

J.W. Bandler and D. Sinha

July 1977

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

AVAILABILITY OF REPORTS

In addition to extensive publications by G-SOC members, a series of reports covering simulation, optimization and control topics is published by the group. Preprints or extended versions of papers, reprints of papers appearing in conference proceedings, fully documented computer program descriptions including listings, theses, notes and manuals appear as reports.

A free list of SOC reports including numbers, titles, authors, dates of publication, and indication of the inclusion of computer listings is available on request. To offset preparation, printing and distribution costs the charges as noted must be made*.

Any number of these reports may be ordered**. Cheques should be made out in U.S. or Canadian Dollars and made payable to McMaster University. Requests must be addressed to the Coordinator G-SOC. Reports will not normally be sent out until payment is received.

Some reports may be temporarily restricted for internal use only. Some may be revised or ultimately superceded. Availability, descriptions or charges are subject to change without notice.

Some of the 174 reports published up to July 1977 are:

SOC-82	Optimal Choice of the Sampling Interval for Discrete Process Control	Mar. 1975	41 pp.	$5
SOC-113	Notes on Numerical Methods of Optimization with Applications in Optimal Design***	Nov. 1975	396 pp.	$150
SOC-121	Computer Programs for Control Applications***	Feb. 1976	53 pp.	$25
SOC-145	The Use of Versatec Printer Plotter and Variable Formats in Computer Plotting Routines***	Dec. 1976	49 pp.	$60
SOC-151	FLOPT4-A Program for Least pth Optimization with Extrapolation to Minimax Solutions***	Jan. 1977	97 pp.	$60
SOC-174	DISOPT3 - A User-Oriented Package for Nonlinear Continuous and Discrete Optimization Problems***	July 1977	182 pp.	$150

* Subscriptions and discounts are available.
** Special reduced rates will be quoted for multiple copies.
*** Include FORTRAN listings. Source decks usually available. Details and prices are contained in the 174 page SOC Catalog ($10.00).

DISOPT3 - A USER-ORIENTED PACKAGE FOR NONLINEAR CONTINUOUS AND DISCRETE OPTIMIZATION PROBLEMS

J.W. Bandler and D. Sinha

July 1977

ACKNOWLEDGEMENTS

The authors would like to thank Dr. C. Charalambous, Department of Systems Design, University of Waterloo, Waterloo, Canada, for making available prepublication copies of his manuscript of his important paper published in Mathematical Programming [2]. Discussions with John Hickin, Department of Electrical Engineering, McMaster University, leading to the idea of the pointers are acknowledged here.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

DISOPT3 - A USER-ORIENTED PACKAGE FOR NONLINEAR CONTINUOUS AND DISCRETE OPTIMIZATION PROBLEMS

J.W. Bandler and D. Sinha

Abstract

A package of FORTRAN subroutines called DISOPT3 for solving continuous and discrete, constrained or unconstrained general optimization problems is presented. The method used for arriving at the discrete solution involves conversion of the original constrained problem into a minimax problem by the Bandler-Charalambous technique, solving the continuous minimax problem using the latest (1977) Charalambous least pth algorithm, Fletcher's 1972 method for unconstrained minimization and use of the Dakin branch and bound technique to generate the additional constraints. These steps are iteratively implemented until all the discrete solutions have been found. DISOPT3 is based conceptually on the DISOPT program developed by Bandler and Chen. All of the desirable features of DISOPT have been retained in DISOPT3 and some more have been added. DISOPT has been used as a yardstick against which the performance and validity of DISOPT3 have been measured. A CDC 6400 computer was used for developing and running this program.

This work was supported by the National Research Council of Canada under Grant A7239.

The authors are with the Group on Simulation, Optimization and Control and Department of Electrical Engineering, McMaster University, Hamilton, Canada, L8S 4L7.

CHAPTER 1

INTRODUCTION

DISOPT3 is a package of FORTRAN subroutines for solving continuous and discrete, constrained or unconstrained general optimization problems. The method used for arriving at the discrete solution involves basically, three steps: (1) Conversion of the original constrained problem into a minimax problem by the Bandler-Charalambous technique [1], (2) Allowing all the variables to be continuous for solving this minimax problem using the latest (1977) Charalambous algorithm [2] and Fletcher's 1972 method for unconstrained minimization [3] and (3) Use of the Dakin branch and bound technique [4] to generate the additional constraints. These steps are iteratively implemented until all the discrete solutions have been found.

DISOPT3 is based conceptually on DISOPT [5,6], a program with similar objectives, developed by Bandler and Chen in 1974. All of the desirable features of DISOPT have been retained in DISOPT3 and some more have been added. DISOPT has been used as a yardstick against which the performance and validity of DISOPT3 have been measured. A CDC 6400 computer was used for developing and running this program.

The goal in developing DISOPT3 was to create an efficient user oriented program. This goal has been amply achieved. DISOPT3 not only incorporates some of the most efficient optimization algorithms but also conforms to the precepts of structured programming. For example, each subroutine performs only one function or some strongly related functions, the program listing is segmented into logical modules by means of comment cards, the use of GO TO statements is minimal, the logical structures are simple, and last but not least, descriptive comments are an integral part of the program listing enhancing its readability and ease of understanding.

This documentation is so organized that it should be possible to solve problems using DISOPT3 after only reading Chapter 2. Chapter 3 has a discussion of the many available options. Chapter 4 deals with the concepts used in developing this program. Chapter 5 summarizes some results obtained by this program. The program listing and some useful references are appended. The reader and potential user of this package should consult, in addition to the references mentioned already [1-6], the following material dealing with the least pth approach in optimization: the paper by Bandler and Charalambous [7] introducing the least pth approach, some extensions [8-10] and a review article by Charalambous [11].

CHAPTER 2

USING DISOPT3

DISOPT3 may be used for solving a mixed continuous-discrete non-linear programming problem which can be formulated as follows:

$$\text{minimize } f(x_1, x_2, \ldots, x_N)$$

subject to

$$g_1(x_1, x_2, \ldots, x_N) \geq 0$$
$$g_2(x_1, x_2, \ldots, x_N) \geq 0$$
$$\cdot$$
$$\cdot$$

where x_1, x_2, ..., x_K or X(1), X(2), ..., X(K) (K.LE.N) are variables that can vary continuously but must assume only certain specified values. These are called discrete variables. Out of the N variables, it is always the first K variables that may be discrete. There are two kinds of discrete variables. The first kind of variable can only assume a finite number of values. The second kind of variable can assume values that correspond to uniformly spaced points on a line, i.e., any value belonging to the infinite set (..., -3a, -2a, -a, 0, a, 2a, 3a, ...) where a is a finite positive quantity. The number a may be called the step size of a uniformly discrete variable. Each of the x_1, x_2, ..., x_K can be a discrete variable of either kind (but always the first K out of the N variables must be discrete).

To use DISOPT3 the main program and a subroutine called FUN have to be provided by the user. The main program is used for dimensioning and initializing some variables and for calling subroutine DISOPT3.

Subroutine FUN evaluates the objective function, the constraints and the gradient vectors at a given point X. Example 1, at the end of this Chapter, illustrates these two subprograms as well as the resulting output. According to a convention used in DISOPT3 the objective function is described as the first constraint and must be counted in along with the constraints.

The arrays and variables that are used in the main program and subroutine FUN are described here.

CONS An array storing the constraints of the problem. The objective function is, by convention, called the first constraint. It must be dimensioned in subroutine FUN as CONS(1) or CONS(NORCONS).

DIS An array of M + IEXTRA * (N+2) elements that must be dimensioned in the main program. The first M elements of DIS must be initialized in the main program according to the following convention:

(a) If a discrete solution is required go to step (b); otherwise, let DIS(1) = 0. In this case M = 1 and skip the following steps.

(b) Let I = 1 and J = 1.

(c) If X(I) is not uniformly discrete go to step (d); otherwise, let DIS(J) = 1 and DIS(J+1) = the step size of X(I). Let J = J+2. go to step (e).

(d) If the number of available discrete values, V(1) ... V(NI), for X(I) is NI, let DIS(J) = NI, DIS(J+1) = V(1), DIS(J+2) = V(2), ... and DIS(J+NI) = V(NI). Let J = J+NI+1. Go to step (e).

(e) Is X(I) the last discrete variable? If yes, let DIS(J) = 0. The initialization of array DIS is complete and M = J. Otherwise, let I = I+1 and return to step (b).

To further illustrate this convention, consider the following example. The problem considered has three variables which are discrete. X(1) has a set of values {1.0, 2.5, 3.7}; X(2) has a uniform step size of 1.5 and X(3) has a set of values {2.0, 5.0, 10.0, 15.0}. The correct initialization of DIS would require:

DIS(1) = 3.0 DIS(2) = 1.0 DIS(3) = 2.5 DIS(4) = 3.7

DIS(5) = 1.0 DIS(6) = 1.5

DIS(7) = 4.0 DIS(8) = 2.0 DIS(9) = 5.0 DIS(10) = 10.0 DIS(11) = 15.0

DIS(12) = 0.0

GCONS An array of (N, NORCONS) elements storing the gradient vectors of the constraints. For each of the NORCONS constraints there are N elements storing its partial derivatives. It must be dimensioned in subroutine FUN.

IAR An array of 6 * IEXTRA + 4 * N + 2 * NORCONS elements used as working space. It must be dimensioned in the main program.

IEXTRA The default value is 2 * N. IEXTRA is a measure of the space allowed by the user to accomodate the additional constraints generated by the branch and bound algorithm.

IFN Serves as a counter for the function evaluations.

N The number of variables in the problem. It must always be greater than 1.

NORCONS The number of constraints in the problem. The objective function must be counted in along with the constraints.

X — An array of (10 * IEXTRA + N ** 2 + 15 * N + 2 * N * NORCONS + 10 * NORCONS)/2 elements used as working space. The first N elements store the starting point at the beginning and the solution point at all other times. This array must be dimensioned in the main program and the first N elements should be initialized. It should also be dimensioned as X(1) in subroutine FUN.

XD — An array of N elements storing the best discrete solution. It must be dimensioned in the main program.

Example 1: The modified banana shaped function [5]

Minimize

$$f = 100((x_2+0.5) - (x_1+0.6)^2)^2 + (0.4 - x_1)^2$$

where x_1 and x_2 are constrained to be natural numbers.

The optimal solution is

$$f = 0.72$$

$$x_1 = 1.0$$

$$x_2 = 2.0$$

In order to arrive at this solution, many nodes are generated by the branch and bound algorithm. The solution and the constraint added at each node are shown in Table I and Figure 1. The nodes are numbered to reflect the order in which they are generated. A listing of the main program, subroutine FUN and the output are also presented.

TABLE I SUMMARY OF RESULTS FOR EXAMPLE 1

Node number	Upper bound	Objective function	Solution x_1, x_2	Description
0	10^{10}	0	0.40, 0.50	continuous
1	2.12	0.16	0.00, -0.14	feasible
2	--	2.14	-0.56, -0.61	nonfeasible
3	--	2.12	0.00, 0.00	discrete
4	--	0.36	1.00, 2.06	feasible
5	--	0.72	1.00, 2.00	discrete
6	0.72	0.75	1.26, 2.99	nonfeasible

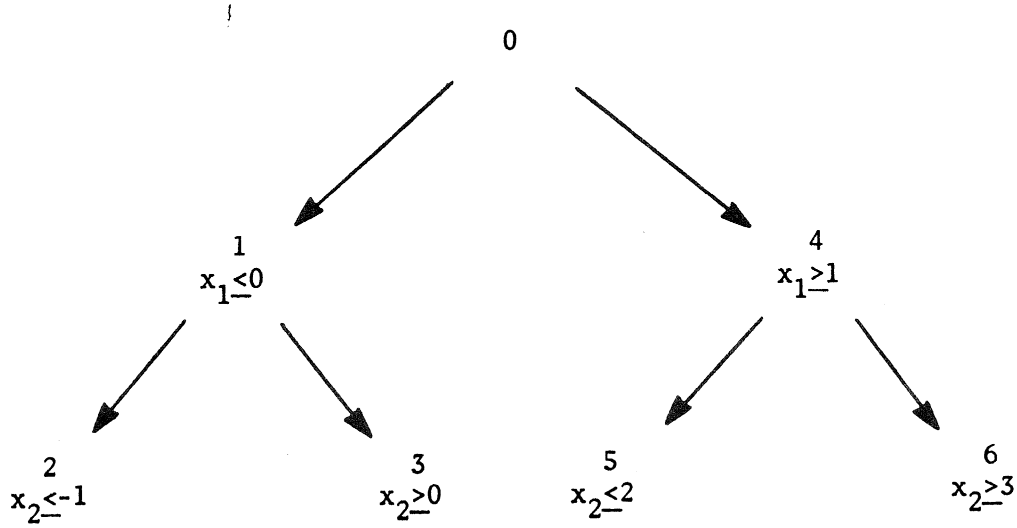


Fig. 1 Tree structure for Example 1.

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 1
C
      DIMENSION DIS(25), IAR(35), X(45), XD(2)
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
C
      DATA X(1),X(2)/-1.8,0.5/
      DATA DIS(1),DIS(2),DIS(3),DIS(4),DIS(5)/4*1.0,0./
C
      N=2
      NORCONS=1
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE MODIFIED BANANA SHAPED FUNCTION
C
C     THIS SUBROUTINE DEFINES THE CONSTRAINTS AND THEIR GRADIENT VECTORS
C     ACCORDING TO THE CONVENTION FOLLOWED IN THIS PROGRAM THE OBJECTIVE
C     FUNCTION IS CALLED THE FIRST CONSTRAINT
C
      DIMENSION CONS(1), GCONS(2), X(2)
C
      COMMON /7/ IFN,IND1,IND2
C
      A=X(1)+.6
      B=X(2)+.5
      C=.4-X(1)
      D=B-A*A
C
C     DEFINE THE OBJECTIVE FUNCTION
C
      CONS(1)=100.*D*D+C*C
C
C     DEFINE THE GRADIENT VECTOR
C
      GCONS(2)=200.*D
      GCONS(1)=-2.*(GCONS(2)*A+C)
      IFN=IFN+1
      RETURN
      END
```

INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISOPT3

```
INITIAL VALUE OF THE ELEMENTS OF AL ... ALMIN =  .10000000E+02

OPTIMAL OBJECTIVE AT NODE 0 (GUESS) ..... EST = 0.

VALUE OF PARAMETER P ..................... IP =    10

(-LARGE,LARGE) BRACKETS ALL VARIABLES . LARGE =  .10000000E+11

ALLOWED FUNCTION CALLS AT EACH NODE .. MAXIFN =  1000

ALLOWED QUASID CALLS AT EACH NODE .... MAXITN =    15

ALLOWED NUMBER OF NODES ............. MAXNODE =  1000

NUMBER OF DISCRETE VARIABLES ........... NDIS =     0

NUMBER OF CONSTRAINTS IN THE PROBLEM  NORCONS =     1

NUMBER OF UNIFORM STEP VARIABLES ....... NUNI =     2

TOLERANCE FOR THE CONSTRAINTS ....... TOLCONS = -.10000000E-02

TOLERANCE FOR THE DISCRETE VARIABLES . TOLDIS =  .10000000E-02

STOPPING CRITERION FOR UOPT ......... TOLHEXI =  .10000000E-02

TOLERANCE FOR THE MULTIPLIERS ....... TOLMULT =  .10000000E-07

STOPPING CRITERION FOR QUASID .......... TOLX =  .10000000E-06

INITIAL VALUE OF THE UPPER BOUND ...... UPBND =  .10000000E+11

STARTING POINT FOR THIS PROBLEM ........... X 1 -.18000000E+01
                                              2  .50000000E+00

X(   1) IS UNIFORM STEP WITH STEP SIZE        =  .10000000E+01

X(   2) IS UNIFORM STEP WITH STEP SIZE        =  .10000000E+01
```

OPTIONS IN EFFECT

GRADIENT CHECK AT THE STARTING POINT

ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION

VERTICES AROUND NODE 0 SOLUTION EXAMINED

OPTIMAL SOLUTION AT EACH NODE PRINTED

GRADIENT CHECK AT THE STARTING POINT

ANALYTICAL GRADIENT VECTOR G(I)	NUMERICAL GRADIENT VECTOR Y(I)	PERCENTAGE ERROR VECTOR PERCENT(I)
1 -.21560000E+03	1 -.21560002E+03	1 .72116698E-05
2 -.88000000E+02	2 -.88000000E+02	2 .31356897E-08

THE GRADIENTS APPEAR TO BE CORRECT

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

THE SOLUTION WITH 1 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X 1 .40000000E+00 2 .50000000E+00

CONS 1 .25710870E-24

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 57

OUT OF THESE 52 WERE PERFORMED AT THIS NODE

THE UPPER BOUND HAS BEEN UPDATED AT THIS NODE. THE DISCRETE SOLUTION AND THE CONSTRAINTS (CONS(1)=UPPER BOUND) FOLLOWING A CHECK AT THE VERTICES SURROUNDING THE NODE 0 SOLUTION ARE

X 1 0. 2 0.

CONS 1 .21200000E+01

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE   1

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE.   1 0.

THE SOLUTION WITH   2 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1 0.                  2 -.14000000E+00
CONS     1  .16000000E+00     2 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS   131

OUT OF THESE   69 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 2

THIS SOLUTION IS NONFEASIBLE

```
THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE.   2 -.10000000E+01   1 0.

THE SOLUTION WITH   3 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

   X     1 -.55673257E+00   2 -.60868830E+00

CONS     1  .21376967E+01   2  .55673257E+00   3 -.39131170E+00

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS  177

OUT OF THESE   46 WERE PERFORMED AT THIS NODE
```

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 3

THIS IS A DISCRETE SOLUTION

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 1 0.

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 2 0.

THE SOLUTION WITH 3 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	0.	2	0.		
CONS	1	.21200000E+01	2	0.	3	0.

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 248

OUT OF THESE 71 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 4

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 1 .10000000E+01

THE SOLUTION WITH 2 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.10000000E+01	2	.20600000E+01
CONS	1	.36000000E+00	2	0.

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 304

OUT OF THESE 56 WERE PERFORMED AT THIS NODE

```
OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE    5
---------------------------------------------------
THIS IS A DISCRETE SOLUTION
---------------------------

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE.   2  .20000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE.   1  .10000000E+01

THE SOLUTION WITH   3 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

   X     1  .10000000E+01   2  .20000000E+01

CONS     1  .72000000E+00   2 0.                3 0.

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS  379

OUT OF THESE   75 WERE PERFORMED AT THIS NODE
```

```
OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE   6
---------------------------------------------------
THIS SOLUTION IS NONFEASIBLE
----------------------------

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE.    2  .30000000E+01    1  .10000000E+01

THE SOLUTION WITH    3 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

   X     1  .12663456E+01    2  .29855667E+01

CONS     1  .75109331E+00    2  .26634555E+00    3 -.14433311E-01

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS  396

OUT OF THESE   17 WERE PERFORMED AT THIS NODE
```

CHAPTER 3

OPTIONS IN DISOPT3

The feature of default values for many of the variables in DISOPT3 has been provided for the convenience of the user; but it is, indeed, possible and sometimes desirable to initialize these variables in the main program choosing different values. The user could, thus, opt for fast execution, no printout at all or, a detailed printout, etc. Table II at the end of this Chapter lists the default values of all the variables. Examples 2 and 3 illustrate the use of these variables.

By choosing appropriate values for the variables, by initializing these variables in the main program (without using DATA statements) and, by including a relevant COMMON statement in the main program the user can greatly influence the performance of the program. The many alternatives to choose from will now be described.

1. Only ONE discrete solution or are ALL required?

If there are many optimal discrete solutions to a problem, will the user be satisfied with just one? If the answer is yes, let ONESOL, a logical variable, be TRUE; otherwise, FALSE. Finding all the solutions requires more effort than finding just one.

2. VERTICES to be checked for an UPPER BOUND?

The effort required to find an optimal discrete solution using the branch and bound algorithm strongly depends on how soon a good upper bound can be found. If the user thinks that the objective function for his problem could not be larger than, say, 10.5 at the optimal discrete solution, he could set UPBND (the upper bound) = 10.5 in the main program. A value of UPBND which is lower than the actual objective function value (at the optimal discrete solution) will result in the program's inability to find any solution at all; whereas, too large a value will not save any effort.

An upper bound is automatically generated and updated whenever a discrete solution is found at a node but DISOPT3 also examines the discrete points surrounding the solution at node 0 if VERTCHK, a logical variable, is TRUE. This method of generating the upper bound could save a lot of effort if the user has no idea about the upper bound. If the user has a good idea, let VERTCHK be FALSE, and save some function evaluations.

3. TOLERANCES?

The choice of numbers for such variables as TOLCONS, TOLDIS, TOLHEXI, TOLMULT and TOLX is critical to the efficiency of the program. All the tolerances should be chosen sufficiently small with respect to the magnitude of numbers involved in a problem. While too small a value for TOLX and TOLHEXI may result in excessive effort, too large a value could lead to the program's inability to find any solution at all. In test runs and to gain information about a problem, one could use large values and then switch to tight values along with some of the above features to economize on effort and obtain a highly refined solution.

These tolerances are described as follows:

TOLCONS A small negative number. If a constraint value is smaller than 0 but larger than or equal to TOLCONS, it is considered as satisfied.

TOLDIS A small positive number. If a variable lies within TOLDIS neighbourhood of a discrete value, it is assumed to be discrete.

TOLHEXI A small positive number. Used by subroutine UOPT as a stopping criterion in the algorithm (see Charalambous [2]) that determines the continuous solution at each node.

TOLMULT A small positive number. Used in subroutine UOPT to select active constraints. If the multiplier (see Charalambous [2]) for a constraint exceeds TOLMULT, it is considered to be active. The active constraints are the only constraints that are used during the following optimization. By choosing TOLMULT as 0, the user can force all the constraints to be active all the time.

TOLX A small positive number. Used in subroutine QUASID (Fletcher algorithm [3]) to test the convergence of the solution.

4. <u>Alternatives for PRINTING results?</u>

Two hollerith variables, PRINTID and PRINTP, influence printing and offer the following options.

PRINTID = 3HYES if the input data is to be printed, 2HNO otherwise.

PRINTP = 4HNONE for no printing at all by any part of the program.

7HONLYDIS for printing discrete solutions only.

7HNODEOPT for printing the optimal solution at each node whether or not it is discrete.

3HALL for printing the details of the optimization at each node. Results are printed after every IPT iterations of subroutine QUASID. IPT may also be changed by the user.

5. Check the user's definition of the gradients?

Often, there is a mistake in the definition of gradients in subroutine FUN. The results obtained as such will be meaningless. This waste of effort might be avoided by setting GRADCHK, a logical variable, as TRUE.

When GRADCHK is true, the gradients are calculated (at the starting point) numerically and also by the user's definition. If the discrepancy is less than 10%, the user's definition is assumed to be correct; the possibility that the gradients are wrong must not still be ruled out, though. If the gradients are correct, a logical variable WRONG is FALSE; otherwise, it is TRUE and the program is terminated. In either case a message is printed.

6. Hold a DISCRETE VARIABLE constant?

In the branch and bound algorithm, additional constraints e.g., $X \leq XL$ or $X \geq XU$ are added to the problem if X is supposed to be a discrete variable but does not assume a discrete value in the optimal solution. There are two ways to implement it: (1) add the constraint and optimize, (2) do not add the constraint, hold X constant at the appropriate bound and optimize. The second alternative is, generally, more efficient and may be chosen by setting HOLDVAR, a logical variable, equal to TRUE. In the rare case when this method fails, it should not be used.

7. Branching on the FIRST or the LAST variable?

Many of the discrete variables may not have a discrete value in the solution. For the additional constraint, as explained above, should the first variable be chosen or the last? It is not possible to predict the best choice for every problem. However, if REVERSE, a logical variable, is TRUE the last variable is chosen.

8. Other options?

In addition to the variables described in the above options, the following could also be of interest to the user.

ALMIN Used to initialize each element of vector AL. Vector AL is used to convert the nonlinear programming problem at each node into an exact minimax problem as proposed by Bandler and Charalambous [1]. ALMIN greatly influences the efficiency of the program but usually there is no way to predict a good value for a particular problem.

EST An estimate of the optimal least pth function value at node 0. If initialized properly, this could save some function evaluations in the very first optimization.

IDCONS An array identifying the active constraints, i.e., those constraints which are actually being used in the optimization at any node. This array may be used in subroutine FUN to evaluate only those constraints which are required.

IDVAR An array identifying all the variables except the one which is held constant. If the evaluation of partial derivatives is very time consuming then IDVAR should be used in subroutine FUN to avoid the evaluation of those derivatives which are not needed.

IP The parameter p of least pth optimization (see [2, 7-11]).

An exhaustive list and a complete description of the variables is provided in the program listing of subroutine DISOPT3.

TABLE II DEFAULT VALUES

Variable Name	Default Value
ALMIN	10.
EST	0.
GRADCHK	.TRUE.
HOLDVAR	.TRUE.
IEXTRA	2 * N
IP	10
IPT	500
LARGE	1.0 E+10
MAXIFN	1000
MAXITN	15
MAXNODE	1000
ONESOL	.FALSE.
PRINTID	3HYES
PRINTP	7HNODEOPT
REVERSE	.FALSE.
TOLCONS	-0.001
TOLDIS	0.001
TOLHEXI	0.001
TOLMULT	0.1E-7
TOLX	0.1E-6
UPBND	1.0E+10
VERTCHK	.TRUE.

A variable, with a default value, should not be initialized in the main program by a DATA statement.

Example 2: The Beale constrained problem [5]

Minimize, as in the Beale problem [12],

$$f = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1 x_2 + 2x_1 x_3$$

subject to

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

$$3 - x_1 - x_2 - 2x_3 \geq 0$$

but where x_1, x_2 and x_3 are constrained to be natural numbers.

The optimal solutions are

$$f = 1.0$$

$x_1 = 2.0$	$x_1 = 1.0$	$x_1 = 2.0$
$x_2 = 0.0$	$x_2 = 1.0$	$x_2 = 1.0$
$x_3 = 0.0$	$x_3 = 0.0$	$x_3 = 0.0$

The tree generated by the branch and bound algorithm is shown in Figure 2 and results summarized in Table III. A listing of the main program, subroutine FUN and the output is also presented.

TABLE III SUMMARY OF RESULTS FOR EXAMPLE 2

Node number	Upper bound	Objective function	Solution x_1, x_2, x_3	Description
0	10^{10}	0.11	1.33, 0.77, 0.44	continuous
1	1.00	0.22	1.00, 0.88, 0.55	feasible
2	--	1.34	1.41, 0.00, 0.59	nonfeasible
3	--	0.25	1.00, 1.00, 0.50	feasible
4	--	1.00	1.00, 1.00, 0.00	discrete
5	--	1.07	0.32, 0.91, 1.00	nonfeasible
6	--	0.50	2.00, 0.50, 0.00	feasible
7	--	1.00	2.00, 0.00, 0.00	discrete
8	--	1.00	2.00, 1.00, 0.00	discrete

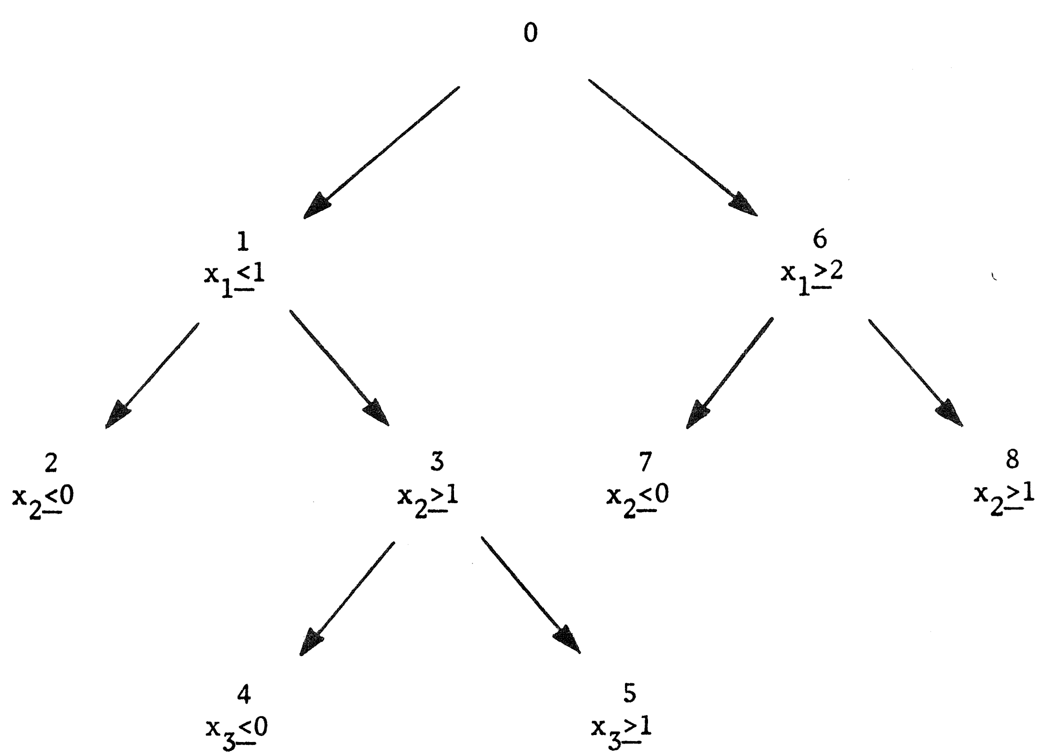


Fig. 2 Tree structure for Example 2.

```
      PROGRAM TST ( INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 2
C
      DIMENSION DIS(50), IAR(60), X( 100), XD(3)
C
      LOGICAL HOLDVAR,ONESOL,REVERSE,VERTCHK
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
C
      DATA X/1.0,2.0,1.0/
      DATA DIS/6*1.0,0.0/
C
      N=3
      NORCONS=5
      PRINTP=3HALL
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE BEALE CONSTRAINED FUNCTION
C
C     THIS SUBROUTINE DEFINES THE CONSTRAINTS AND THEIR GRADIENT VECTORS
C     ACCORDING TO THE CONVENTION FOLLOWED IN THIS PROGRAM THE OBJECTIVE
C     FUNCTION IS CALLED THE FIRST CONSTRAINT
C
      DIMENSION CONS(5), GCONS(15), IDCONS(1), X(3)
C
      COMMON /7/ IFN,IND1,IND2
C
      P=X(1)
      Q=X(2)
      R=X(3)
      A=P+Q
      B=P+R
C
      DO 60 I=1,5
      J=IDCONS(I)
      IF (J.GE.6) GO TO 60
      GO TO (10,20,30,40,50), J
10    CONTINUE
C
C     DEFINE THE OBJECTIVE FUNCTION
C
      CONS(1)=9.+(A-6.)*A+(B-2.)*B-R-R+Q*Q
      IF (IND1.EQ.0) RETURN
      GCONS(1)=(-4.+A+B)*2.
      GCONS(2)=(A+Q-3.)*2.
      GCONS(3)=-4.+2.*B
      GO TO 60
20    CONTINUE
      CONS(2)=P
      GCONS(4)=1.
      GCONS(5)=0.
      GCONS(6)=0.
      GO TO 60
30    CONTINUE
      CONS(3)=Q
      GCONS(7)=0.
      GCONS(8)=1.
      GCONS(9)=0.
      GO TO 60
40    CONTINUE
      CONS(4)=R
      GCONS(10)=0.
      GCONS(11)=0.
      GCONS(12)=1.
      GO TO 60
50    CONTINUE
      CONS(5)=3.-B-Q-R
      GCONS(13)=-1.
      GCONS(14)=-1.
      GCONS(15)=-2.
60    CONTINUE
C
      IFN=IFN+1
      RETURN
      END
```

INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISOPT3

```
INITIAL VALUE OF THE ELEMENTS OF AL ... ALMIN =  .10000000E+02
OPTIMAL OBJECTIVE AT NODE 0 (GUESS) ..... EST = 0.
VALUE OF PARAMETER P ..................... IP =    10
(-LARGE,LARGE) BRACKETS ALL VARIABLES . LARGE =  .10000000E+11
ALLOWED FUNCTION CALLS AT EACH NODE .. MAXIFN =  1000
ALLOWED QUASID CALLS AT EACH NODE .... MAXITN =    15
ALLOWED NUMBER OF NODES ............. MAXNODE =  1000
NUMBER OF DISCRETE VARIABLES ........... NDIS =     0
NUMBER OF CONSTRAINTS IN THE PROBLEM  NORCONS =     5
NUMBER OF UNIFORM STEP VARIABLES ....... NUNI =     3
TOLERANCE FOR THE CONSTRAINTS ....... TOLCONS = -.10000000E-02
TOLERANCE FOR THE DISCRETE VARIABLES . TOLDIS =  .10000000E-02
STOPPING CRITERION FOR UOPT ......... TOLHEXI =  .10000000E-02
TOLERANCE FOR THE MULTIPLIERS ....... TOLMULT =  .10000000E-07
STOPPING CRITERION FOR QUASID .......... TOLX =  .10000000E-06
INITIAL VALUE OF THE UPPER BOUND ...... UPBND =  .10000000E+11
STARTING POINT FOR THIS PROBLEM ........... X 1  .10000000E+01
                                              2  .20000000E+01
                                              3  .10000000E+01

X(   1) IS UNIFORM STEP WITH STEP SIZE       =   .10000000E+01
X(   2) IS UNIFORM STEP WITH STEP SIZE       =   .10000000E+01
X(   3) IS UNIFORM STEP WITH STEP SIZE       =   .10000000E+01

OPTIONS IN EFFECT
GRADIENT CHECK AT THE STARTING POINT
ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION
VERTICES AROUND NODE 0 SOLUTION EXAMINED
DETAILED PRINTING REQUESTED
```

GRADIENT CHECK AT THE STARTING POINT

	ANALYTICAL GRADIENT VECTOR G(I)		NUMERICAL GRADIENT VECTOR Y(I)		PERCENTAGE ERROR VECTOR PERCENT(I)
1	.12000000E+02	1	.12000000E+02	1	.27853275E-09
2	.14000000E+02	2	.14000000E+02	2	.76291989E-09
3	.20000000E+02	3	.20000000E+02	3	.45793058E-08

THE GRADIENTS APPEAR TO BE CORRECT

FEASIBILITY CHECK AT NODE 0

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.20000000E+01	1	.10000000E+01	1	.10000000E+01
			2	.20000000E+01	2	.10000000E+01
			3	.10000000E+01	3	.20000000E+01
2	3	.33333333E+00	1	.22727273E+00	1	-.23229083E+00
			2	.12272727E+01	2	.17811275E-03
			3	.18181818E+00	3	-.45361504E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.28801653E+01
	2	.10000000E+02	2	.22727273E+00
	3	.10000000E+02	3	.12272727E+01
	4	.10000000E+02	4	.18181818E+00
	5	.10000000E+02	5	.11818182E+01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	5	.28801787E+01	1	.22727273E+00	1	-.42730980E+01
			2	.12272727E+01	2	-.63641762E+00
			3	.18181818E+00	3	-.31833479E+01
22	47	.11213284E+00	1	.13346134E+01	1	.21623398E-08
			2	.77692441E+00	2	.19607237E-08
			3	.44231104E+00	3	.44486781E-08

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .11196611E+00

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.11196611E+00
	2	.10000000E+02	2	.13346134E+01
	3	.10000000E+02	3	.77692441E+00
	4	.10000000E+02	4	.44231104E+00
	5	.10000000E+02	5	.38401338E-02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	49	-.99999564E-10	1	.13346134E+01	1	-.22307559E+00
			2	.77692441E+00	2	-.22307559E+00
			3	.44231104E+00	3	-.44615117E+00
8	77	-.84460118E-03	1	.13333449E+01	1	-.40486177E-04
			2	.77777004E+00	2	-.40526508E-04
			3	.44442512E+00	3	-.80986551E-04

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .11111884E+00

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.11111884E+00
2	.66651804E-45	2	INACTIVE	2	.13333449E+01
3	.25037991E-42	3	INACTIVE	3	.77777004E+00
4	.11797444E-39	4	INACTIVE	4	.44442512E+00
5	.22218895E+00	5	.44437790E+00	5	.34790830E-04

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	81	-.99999564E-10	1	.13333449E+01	1	-.22222994E+00
			2	.77777004E+00	2	-.22222998E+00
			3	.44442512E+00	3	-.44445989E+00
8	96	-.72137727E-05	1	.13333333E+01	1	-.30262976E-05
			2	.77777777E+00	2	-.30395519E-05
			3	.44444445E+00	3	-.60223060E-05

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

THE SOLUTION WITH 5 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .13333333E+01    2  .77777777E+00    3  .44444445E+00

CONS     1  .11111111E+00    2  .13333333E+01    3  .77777777E+00
         4  .44444445E+00    5 -.42769699E-09
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 107

OUT OF THESE 100 WERE PERFORMED AT THIS NODE

THE UPPER BOUND HAS BEEN UPDATED AT THIS NODE. THE DISCRETE SOLUTION AND THE CONSTRAINTS (CONS(1)=UPPER BOUND) FOLLOWING A CHECK AT THE VERTICES SURROUNDING THE NODE 0 SOLUTION ARE

```
   X     1  .20000000E+01    2 0.                3 0.

CONS     1  .10000000E+01    2  .20000000E+01    3 0.
         4 0.                5  .10000000E+01
```

FEASIBILITY CHECK AT NODE 1

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	-.22608922E+00	2	.77777777E+00	2	.23810759E+00
			3	.44444445E+00	3	.43072415E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 1

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.40740741E+00
	2	.10000000E+02	2	.10000000E+01
	3	.10000000E+02	3	.77777777E+00
	4	.10000000E+02	4	.44444445E+00
	5	.10000000E+02	5	.33333333E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	.40740741E+00	2	.77777777E+00	2	-.88888891E+00
			3	.44444445E+00	3	-.11111111E+01
11	27	.22585559E+00	2	.88816768E+00	2	-.32797472E-06
			3	.55267076E+00	3	-.60288207E-06

```
ITERATION NUMBER  2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)
--------------------------------------------------------------------
FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE    1
-------------------------------------------------

VALUE OF HEXI FOR THIS ITERATION ....... HEXI =  .22511639E+00

                 MULTIPLIER           ALPHA             CONSTRAINT
               VECTOR RMULT(I)     VECTOR AL(I)       VECTOR CONS(I)

               NOT CALCULATED  1   OBJECTIVE       1   .22511639E+00
                               2   .10000000E+02   2   .10000000E+01
                               3   .10000000E+02   3   .88816768E+00
                               4   .10000000E+02   4   .55267076E+00
                               5   .10000000E+02   5   .64908043E-02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD
-----------------------------------------------------------------

ITER. FUNC. LEAST PTH            VARIABLE           GRADIENT
 NO.  EVAL. FUNCTION            VECTOR X(I)        VECTOR G(I)

  0    29 -.99999120E-10   2  .88816768E+00   2 -.44732927E+00
                           3  .55267076E+00   3 -.89465849E+00


  6    52 -.28363944E-02   2  .88887869E+00   2 -.41278681E-06
                           3  .55551476E+00   3 -.83238579E-06
```

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 1

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .22226302E+00

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.22226302E+00
2	.97235922E-38	2	INACTIVE	2	.10000000E+01
3	.35513129E-37	3	INACTIVE	3	.88887869E+00
4	.62385672E-35	4	INACTIVE	4	.55551476E+00
5	.44448481E+00	5	.88896963E+00	5	.91787746E-04

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	56	-.99999120E-10	2	.88887869E+00	2	-.44448524E+00
			3	.55551476E+00	3	-.88897048E+00
6	70	-.38064505E-04	2	.88888889E+00	2	-.53115074E-08
			3	.55555556E+00	3	-.10631369E-07

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 1

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 1 .10000000E+01

THE SOLUTION WITH 5 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .10000000E+01    2  .88888889E+00    3  .55555556E+00

CONS     1  .22222222E+00    2  .10000000E+01    3  .88888889E+00
         4  .55555556E+00    5  .75795015E-09
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 185

OUT OF THESE 74 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 2

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.11975309E+01	1	.10000000E+01	1	-.28888889E+01
			3	.55555556E+00	3	-.88888889E+00
10	11	.53784145E+00	1	.14102455E+01	1	.28057037E-07
			3	.58975458E+00	3	.76916869E-07

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 2

THIS SOLUTION IS NONFEASIBLE

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 2 0. 1 .10000000E+01

THE SOLUTION WITH 6 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.14102455E+01	2	0.	3	.58975458E+00
CONS	1	.13478104E+01	2	.14102455E+01	3	0.
	4	.58975458E+00	5	.41024536E+00	6	-.41024548E+00

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 197

OUT OF THESE 12 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 3

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.11111111E+00	1	.10000000E+01	1	.10000000E+01
			3	.55555556E+00	3	.20000000E+01
1	2	.11111111E+00	1	.95555556E+00	1	.83215416E+00
			3	.46666667E+00	3	.99206532E-01

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 3

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.33580247E+00
	2	.10000000E+02	2	.95555556E+00
	3	.10000000E+02	3	.10000000E+01
	4	.10000000E+02	4	.46666667E+00
	5	.10000000E+02	5	.11111111E+00
	6	.10000000E+02	6	.44444444E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	4	.33580247E+00	1	.95555556E+00	1	-.12444444E+01
			3	.46666667E+00	3	-.11555556E+01
10	30	.25916355E+00	1	.99302956E+00	1	-.39806387E-07
			3	.49997180E+00	3	-.85146215E-07

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 3

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .25709620E+00

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.25709620E+00
	2	.10000000E+02	2	.99302956E+00
	3	.10000000E+02	3	.10000000E+01
	4	.10000000E+02	4	.49997180E+00
	5	.10000000E+02	5	.70268309E-02
	6	.10000000E+02	6	.69704366E-02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	32	-.99998232E-10	1	.99302956E+00	1	-.10279381E+01
			3	.49997180E+00	3	-.10139973E+01
8	68	-.67967310E-02	1	.99979308E+00	1	.21796770E-05
			3	.49999997E+00	3	.23605788E-05

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 3

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .25020704E+00

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.25020704E+00
2	.14850122E-33	2	INACTIVE	2	.99979308E+00
3	.14816379E-33	3	INACTIVE	3	.10000000E+01
4	.30115122E-30	4	INACTIVE	4	.49999997E+00
5	.50020819E+00	5	.15006246E+01	5	.20698885E-03
6	.50062184E+00	6	.15018655E+01	6	.20692153E-03

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	72	-.99998232E-10	1	.99979308E+00	1	-.10008278E+01
			3	.49999997E+00	3	-.10004139E+01
5	91	-.18550009E-03	1	.99999996E+00	1	-.43407569E-04
			3	.50000001E+00	3	-.10513514E-05

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 3

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 1 .10000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 2 .10000000E+01

THE SOLUTION WITH 6 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.10000000E+01	2	.10000000E+01	3	.50000001E+00
CONS	1	.24999999E+00	2	.10000000E+01	3	.10000000E+01
	4	.50000001E+00	5	-.11520996E-07	6	0.

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 292

OUT OF THESE 95 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 4

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	-.50000000E-10	1	.10000000E+01	1	-.12500000E+00
			2	.10000000E+01	2	-.12500000E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 4

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.10000000E+01
	2	.10000000E+02	2	.10000000E+01
	3	.10000000E+02	3	.10000000E+01
	4	.10000000E+02	4	0.
	5	.10000000E+02	5	.10000000E+01
	6	.10000000E+02	6	0.
	7	.10000000E+02	7	0.

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	.11486984E+01	1	.10000000E+01	1	.57434918E+00
			2	.10000000E+01	2	-.28717459E+01
10	17	.11136726E+01	1	.99238094E+00	1	.35642626E-05
			2	.10329596E+01	2	.30651024E-06

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 4

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10170247E+01

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.10170247E+01
	2	.10000000E+02	2	.99238094E+00
	3	.10000000E+02	3	.10329596E+01
	4	.10000000E+02	4	0.
	5	.10000000E+02	5	.97465949E+00
	6	.10000000E+02	6	.76190631E-02
	7	.10000000E+02	7	.32959577E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	19	-.93298335E-10	1	.99238094E+00	1	-.18329966E+01
			2	.10329596E+01	2	.10879182E+00
10	45	-.15310072E-01	1	.99989059E+00	1	.29964347E-07
			2	.10013870E+01	2	.34933965E-08

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 4

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10002224E+01

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.10002224E+01
2	.11842496E-29	2	INACTIVE	2	.99989059E+00
3	.11649596E-29	3	INACTIVE	3	.10013870E+01
4	.39985035E+01	4	.15994014E+02	4	0.
5	.11995506E-29	5	INACTIVE	5	.99872237E+00
6	.19976636E+01	6	.79906544E+01	6	.10941300E-03
7	.53293373E-02	7	.21317349E-01	7	.13870418E-02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	49	-.93298335E-10	1	.99989059E+00	1	-.18638860E+01
			2	.10013870E+01	2	.49724512E-02
7	66	-.19529393E-03	1	.99999950E+00	1	.12615736E-04
			2	.10007507E+01	2	.49573347E-06

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 4

THIS IS A DISCRETE SOLUTION

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE.   3 0.   1 .10000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE.   2 .10000000E+01

THE SOLUTION WITH 7 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X      1  .10000000E+01    2  .10000000E+01    3 0.

CONS      1  .10000000E+01    2  .10000000E+01    3  .10000000E+01
          4 0.                5  .10000000E+01    6 0.
          7 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 362

OUT OF THESE 70 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 5

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.10000000E+01	1	.10000000E+01	1	.10000000E+01
			2	.10000000E+01	2	.10000000E+01
11	16	.25455486E+00	1	.31608494E+00	1	.26094082E-07
			2	.90870323E+00	2	.14392533E-07

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 5

THIS SOLUTION IS NONFEASIBLE

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 1 .10000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 3 .10000000E+01 2 .10000000E+01

THE SOLUTION WITH 7 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .31608494E+00    2  .90870323E+00    3  .10000000E+01

CONS     1  .10770283E+01    2  .31608494E+00    3  .90870323E+00
         4  .10000000E+01    5 -.22478816E+00    6  .68391506E+00
         7 -.91296773E-01
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 379

OUT OF THESE 17 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 6

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.66666667E+00	2	.77777777E+00	2	.10000000E+01
			3	.44444445E+00	3	.20000000E+01
2	3	.88888887E-01	2	.28730158E+00	2	-.79204268E-01
			3	.16349207E+00	3	-.37338345E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 6

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.61721089E+00
	2	.10000000E+02	2	.20000000E+01
	3	.10000000E+02	3	.28730158E+00
	4	.10000000E+02	4	.16349207E+00
	5	.10000000E+02	5	.38571429E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	5	.61721089E+00	2	.28730158E+00	2	-.85079368E+00
			3	.16349207E+00	3	.32698413E+00
12	20	.50063389E+00	2	.50000000E+00	2	-.75165179E-09
			3	.22570296E-01	3	-.12247543E-08

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 6

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .50050942E+00

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.50050942E+00
	2	.10000000E+02	2	.20000000E+01
	3	.10000000E+02	3	.50000000E+00
	4	.10000000E+02	4	.22570296E-01
	5	.10000000E+02	5	.45485941E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	22	-.99998232E-10	2	.50000000E+00	2	-.74993523E-09
			3	.22570296E-01	3	.45140592E-01
6	34	-.50940839E-03	2	.50000000E+00	2	.27638593E-08
			3	.87027274E-04	3	.20734285E-06

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 6

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .50000001E+00

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.50000001E+00
2	.29261078E-49	2	INACTIVE	2	.20000000E+01
3	.12262675E-42	3	INACTIVE	3	.50000000E+00
4	.17384720E-03	4	.34769440E-03	4	.87027274E-04
5	.12309725E-42	5	INACTIVE	5	.49982594E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	38	-.99998232E-10	2	.50000000E+00	2	.27638976E-08
			3	.87027274E-04	3	.17405455E-03
5	47	-.75787222E-08	2	.50000000E+00	2	-.47857777E-08
			3	.72849047E-05	3	.36575737E-06

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 6

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

```
X.GE.    1  .20000000E+01

THE SOLUTION WITH   5 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

   X     1  .20000000E+01    2  .50000000E+00    3  .72849047E-05

CONS     1  .50000000E+00    2  .20000000E+01    3  .50000000E+00
         4  .72849047E-05    5  .49998543E+00
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 430

OUT OF THESE 51 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 7

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.52963856E-10	1	.20000000E+01	1	.14569809E-04
			3	.72849047E-05	3	.14569809E-04

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 7

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

```
MULTIPLIER            ALPHA                  CONSTRAINT
VECTOR RMULT(I)       VECTOR AL(I)           VECTOR CONS(I)

NOT CALCULATED     1  OBJECTIVE          1  .10000000E+01
                   2  .10000000E+02      2  .20000000E+01
                   3  .10000000E+02      3 0.
                   4  .10000000E+02      4  .72849047E-05
                   5  .10000000E+02      5  .99998543E+00
                   6  .10000000E+02      6 0.
```

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

```
ITER. FUNC. LEAST PTH          VARIABLE              GRADIENT
NO.   EVAL. FUNCTION           VECTOR X(I)           VECTOR G(I)

  0     3   .11486774E+01   1  .20000000E+01     1 -.28721998E+01
                            3  .72849047E-05     3 -.28703172E+01


 11    18   .10795185E+01   1  .20304293E+01     1  .70968842E-09
                            3  .33399616E-01     3  .70612281E-09
```

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 7

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10050001E+01

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.10050001E+01
	2	.10000000E+02	2	.20304293E+01
	3	.10000000E+02	3	0.
	4	.10000000E+02	4	.33399616E-01
	5	.10000000E+02	5	.90277145E+00
	6	.10000000E+02	6	.30429322E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	20	-.93298335E-10	1	.20304293E+01	1	.17589213E+00
			3	.33399616E-01	3	.11910901E+00
13	44	-.46636242E-02	1	.20004823E+01	1	.41285727E-06
			3	.51805758E-03	3	-.71255293E-06

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 7

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10000012E+01

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.10000012E+01
2	.11820716E-38	2	INACTIVE	2	.20004823E+01
3	.49975167E+01	3	.19990067E+02	3	0.
4	.20015491E-02	4	.80061963E-02	4	.51805758E-03
5	.24614314E-35	5	INACTIVE	5	.99848155E+00
6	.29650120E-02	6	.11860048E-01	6	.48233489E-03

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	48	-.93298335E-10	1	.20004823E+01	1	.27668671E-02
			3	.51805758E-03	3	.18667984E-02
8	60	-.11367681E-05	1	.20000345E+01	1	-.64944963E-07
			3	.50192967E-04	3	.19487942E-07

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 7

THIS IS A DISCRETE SOLUTION

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 2 0.

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 1 .20000000E+01

THE SOLUTION WITH 6 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .20000000E+01   2 0.                 3 0.

CONS     1  .10000000E+01   2  .20000000E+01     3 0.
         4 0.               5  .10000000E+01     6 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 494

OUT OF THESE 64 WERE PERFORMED AT THIS NODE

FEASIBILITY CHECK AT NODE 8

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	.14569809E-04	1	.20000000E+01	1	.10000073E+01
			3	.72849047E-05	3	.20000000E+01

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 8

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.10000000E+01
	2	.10000000E+02	2	.20000000E+01
	3	.10000000E+02	3	.10000000E+01
	4	.10000000E+02	4	.72849047E-05
	5	.10000000E+02	5	-.14569809E-04
	6	.10000000E+02	6	0.

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	.11487193E+01	1	.20000000E+01	1	.23011806E+01
			3	.72849047E-05	3	.28807077E+01
6	13	.11430524E+01	1	.19964769E+01	1	-.37383559E-07
			3	-.12992859E-02	3	.88105253E-07

THE ABOVE ITERATION HAS RESULTED IN A NONFEASIBLE SOLUTION. THE CONSTRAINTS AT THIS POINT ARE GIVEN AS FOLLOWS. IT MAY BE NOTED THAT THE STARTING POINT FOR THE NEXT ITERATION IS NOT THE ABOVE SOLUTION BUT THE BEST FEASIBLE POINT OBTAINED SO FAR

CONS	1	.99298957E+00	2	.19964769E+01	3	.10000000E+01
	4	-.12992859E-02	5	.61216224E-02	6	-.35230506E-02

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 8

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.10000000E+01
	2	.10000000E+03	2	.20000000E+01
	3	.10000000E+03	3	.10000000E+01
	4	.10000000E+03	4	.72849047E-05
	5	.10000000E+03	5	-.14569809E-04
	6	.10000000E+03	6	0.

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	16	.11489108E+01	1	.20000000E+01	1	.26755462E+01
			3	.72849047E-05	3	.29613417E+02
7	25	.11438919E+01	1	.20002990E+01	1	.13693891E-04
			3	-.35570137E-03	3	.29574785E-06

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 8

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10361683E+01

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.10005982E+01
	2	.10000000E+03	2	.20002990E+01
	3	.10000000E+03	3	.10000000E+01
	4	.10000000E+03	4	-.35570137E-03
	5	.10000000E+03	5	.41235801E-03
	6	.10000000E+03	6	.29904472E-03

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	27	-.99988462E-10	1	.20002990E+01	1	.20004848E+01
			3	-.35570137E-03	3	-.10000011E+03
8	45	-.31594360E-01	1	.20000087E+01	1	-.20403248E-03
			3	-.10521519E-04	3	.32483524E-02

ITERATION NUMBER 4 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 8

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10015323E+01

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.10000175E+01
2	.17466675E-39	2	INACTIVE	2	.20000087E+01
3	.35702431E-36	3	INACTIVE	3	.10000000E+01
4	.35993559E+02	4	.14397424E+03	4	-.10521519E-04
5	.17998641E+02	5	.71994562E+02	5	.12303536E-04
6	.19998888E+02	6	.79995552E+02	6	.87395017E-05

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	49	-.99991633E-10	1	.20000087E+01	1	.20000139E+01
			3	-.10521519E-04	3	-.14397424E+03
6	62	-.13339506E-02	1	.20000000E+01	1	.90536805E-02
			3	.41434525E-09	3	.32372711E-01

ITERATION NUMBER 5 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 8

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .10000001E+01

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.10000000E+01
2	.17466675E-39	2	INACTIVE	2	.20000000E+01
3	.35702431E-36	3	INACTIVE	3	.10000000E+01
4	.35978219E+02	4	.14391288E+03	4	.41434525E-09
5	.18007703E+02	5	.72030811E+02	5	-.97013178E-09
6	.19997303E+02	6	.79989211E+02	6	.14145485E-09

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	66	-.99992599E-10	1	.20000000E+01	1	.74030811E+02
			3	.41434525E-09	3	.14406162E+03
1	74	-.99992599E-10	1	.20000000E+01	1	.74030811E+02
			3	.41434528E-09	3	.14406162E+03

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 8

THIS IS A DISCRETE SOLUTION

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 2 .10000000E+01 1 .20000000E+01

THE SOLUTION WITH 6 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X      1  .20000000E+01   2  .10000000E+01   3 0.

CONS      1  .10000000E+01   2  .20000000E+01   3  .10000000E+01
          4 0.               5 0.               6 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 572

OUT OF THESE 78 WERE PERFORMED AT THIS NODE

Example 3: The voltage divider problem [5, 13]

Minimize

$$f = 1/x_1 + 1/x_2$$

subject to

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$0.53-(x_4+0.01x_2x_4)/(x_3-0.01x_1x_3+x_4+0.01x_2x_4) \geq 0$$

$$(x_4-0.01x_2x_4)/(x_3+0.01x_1x_3+x_4-0.01x_2x_4)-0.46 \geq 0$$

$$2.15 - x_4 - 0.01\ x_2\ x_4 - x_3 - 0.01\ x_1\ x_3 \geq 0$$

$$x_4 - 0.01\ x_2\ x_4 + x_3 - 0.01\ x_1\ x_3 - 1.85 \geq 0$$

where x_1 and x_2 both belong to the discrete set {1.0, 3.0, 5.0, 10.0, 15.0}.

The optimal solution is

$$f = 0.4$$

$$x_1 = 5.0$$

$$x_2 = 5.0$$

$$x_3 = 1.0130514$$

$$x_4 = 0.9901098$$

The tree generated by the branch and bound algorithm is shown in Figure 3 and the results are summarized in Table IV. A listing of the main program, subroutine FUN and the output is also presented.

TABLE IV SUMMARY OF RESULTS FOR EXAMPLE 3

Node number	Upper bound	Objective function	Solution x_1, x_2, x_3, x_4	Description
0	10^{10}	0.28	7.00, 7.00, 1.01, 0.99	continuous
1	--	0.31	8.99, 5.00, 1.01, 0.99	feasible
2	--	0.40	5.00, 5.00, 1.01, 0.99	discrete
3	0.40	0.35	10.00, 3.99, 1.02, 0.99	feasible
4	--	0.41	12.29, 3.00, 1.01, 0.99	nonfeasible
5	--	0.30	10.00, 5.00, 1.01, 0.99	nonfeasible
6	--	0.35	3.99, 10.00, 1.01, 0.99	feasible
7	--	0.41	3.00, 12.43, 1.01, 0.99	nonfeasible
8	--	0.30	5.00, 10.00, 1.01, 0.99	nonfeasible

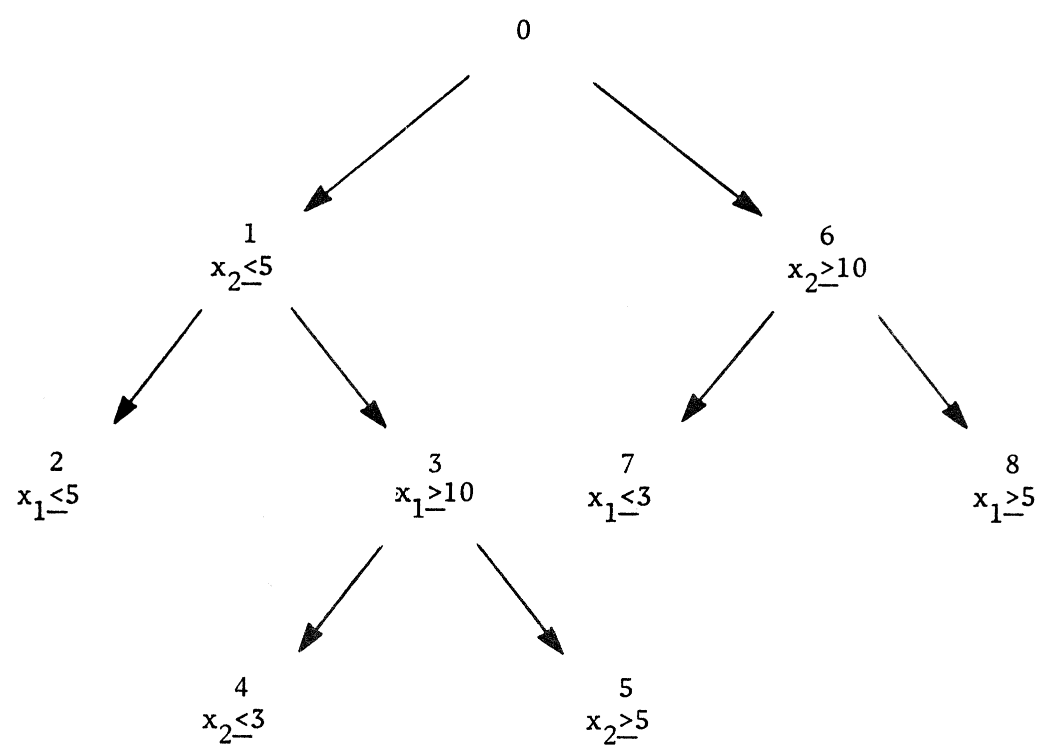


Fig. 3 Tree structure for Example 3.

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 3
C
      DIMENSION DIS(100), IAR(100), X(150), XD(4)
C
      LOGICAL GRADCHK,ONESOL,REVERSE,VERTCHK
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /10/ GRADCHK,HOLDVAR,ONESOL,REVERSE,VERTCHK
C
      DATA DIS/5.,1.,3.,5.,10.,15.,5.,1.,3.,5.,10.,15.,0./
      DATA X(1),X(2),X(3),X(4)/4*1./
C
      ALMIN=100.
      GRADCHK=.FALSE.
      IPT=50
      N=4
      NORCONS=7
      PRINTID=2HNO
      REVERSE=.TRUE.
      VERTCHK=.FALSE.
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE VOLTAGE DIVIDER EXAMPLE
C
C     THIS SUBROUTINE DEFINES THE CONSTRAINTS AND THEIR GRADIENT VECTORS
C     ACCORDING TO THE CONVENTION FOLLOWED IN THIS PROGRAM THE OBJECTIVE
C     FUNCTION IS CALLED THE FIRST CONSTRAINT
C
      DIMENSION CONS(7), DE(4), E(2), GCONS(4,7), IDCONS(1), IDVAR(1), X
     1(4)
C
      COMMON /7/ IFN,IND1,IND2
C
      TM=1./X(1)
      TN=1./X(2)
      DE(1)=X(1)*0.01
      DE(2)=X(2)*0.01
      E(1)=DE(1)*X(3)
      E(2)=DE(2)*X(4)
      TA=X(3)+E(1)
      TB=X(3)-E(1)
      TC=X(4)+E(2)
      TD=X(4)-E(2)
      TE=TB+TC
      TF=TA+TD
C
      DO 80 I=1,7
      J=IDCONS(I)
      IF (J.GE.8) GO TO 80
      GO TO (10,20,30,40,50,60,70), J
10    CONTINUE
C
C     DEFINE THE OBJECTIVE FUNCTION
C
      CONS(1)=TM+TN
      IF (IND1.EQ.0) RETURN
      GO TO 80
C
C     DEFINE THE OTHER CONSTRAINTS
C
20    CONTINUE
      CONS(2)=X(1)
      GO TO 80
30    CONTINUE
      CONS(3)=X(2)
      GO TO 80
40    CONTINUE
      CONS(4)=0.53-TC/TE
      GO TO 80
50    CONTINUE
      CONS(5)=TD/TF-0.46
      GO TO 80
60    CONTINUE
      CONS(6)=2.15-TC-TA
      GO TO 80
70    CONTINUE
      CONS(7)=TD+TB-1.85
80    CONTINUE
C
      IF (IND2.EQ.0) RETURN
C
C     DEFINE THE GRADIENT VECTORS
C
      DE(3)=X(3)*0.01
      DE(4)=X(4)*0.01
      TG=TE*TE
      TH=TF*TF
      TI=1.+DE(1)
      TJ=1.-DE(1)
      TK=1.+DE(2)
      TL=1.-DE(2)
      TP=TC/TG
      TQ=-TD/TH
```

```
      TR=TA/TH
      TS=-TB/TG
C
      DO 130 I=1,4
      J=IDVAR(I)
      IF (J.GE.5) GO TO 130
      GO TO (90,100,110,120), J
 90   CONTINUE
      GCONS(1,1)=-(TM**2)
      GCONS(1,2)=1.0
      GCONS(1,3)=0.
      GCONS(1,4)=-TP*DE(3)
      GCONS(1,5)=TQ*DE(3)
      GCONS(1,6)=-DE(3)
      GCONS(1,7)=GCONS(1,6)
      GO TO 130
 100  CONTINUE
      GCONS(2,1)=-(TN**2)
      GCONS(2,2)=0.
      GCONS(2,3)=1.0
      GCONS(2,4)=TS*DE(4)
      GCONS(2,5)=-TR*DE(4)
      GCONS(2,6)=-DE(4)
      GCONS(2,7)=GCONS(2,6)
      GO TO 130
 110  CONTINUE
      GCONS(3,1)=0.
      GCONS(3,2)=0.
      GCONS(3,3)=0.
      GCONS(3,4)=TP*TJ
      GCONS(3,5)=TQ*TI
      GCONS(3,6)=-TI
      GCONS(3,7)=TJ
      GO TO 130
 120  CONTINUE
      GCONS(4,1)=0.
      GCONS(4,2)=0.
      GCONS(4,3)=0.
      GCONS(4,4)=TS*TK
      GCONS(4,5)=TR*TL
      GCONS(4,6)=-TK
      GCONS(4,7)=TL
 130  CONTINUE
C
      IFN=IFN+1
      RETURN
      END
```

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

THE SOLUTION WITH 7 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .70024624E+01    2  .70024625E+01    3  .10115760E+01
         4  .99144712E+00

CONS     1  .28561381E+00    2  .70024624E+01    3  .70024625E+01
         4 -.88045775E-05    5 -.87828078E-05    6  .67159319E-02
         7  .12762184E-01
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 77

OUT OF THESE 77 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 1

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 2 .50000000E+01

THE SOLUTION WITH 7 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .89957880E+01    2  .50000000E+01    3  .10130514E+01
         4  .99010598E+00

CONS     1  .31116314E+00    2  .89957880E+01    3  .50000000E+01
         4  .68610575E-08    5  .68398069E-08    6  .62053502E-02
         7  .12520136E-01
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 152

OUT OF THESE 75 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 2

THIS IS A DISCRETE SOLUTION

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

```
X.LE.    1  .50000000E+01   2  .50000000E+01

THE SOLUTION WITH   8 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

   X     1  .50000000E+01   2  .50000000E+01   3  .10130514E+01
         4  .99010598E+00

CONS     1  .40000000E+00   2  .50000000E+01   3  .50000000E+01
         4  .10716274E-01   5  .92902703E-02   6  .46684737E-01
         7  .52999523E-01   8 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 220

OUT OF THESE 68 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 3

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 2 .50000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 1 .10000000E+02

THE SOLUTION WITH 8 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.10000000E+02	2	.39886960E+01	3	.10152383E+01
	4	.99083734E+00				
CONS	1	.35070850E+00	2	.10000000E+02	3	.39886960E+01
	4	.99014130E-11	5	.97699626E-13	6	.28790935E-02
	7	.15030277E-01	8	.10113040E+01		

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 305

OUT OF THESE 85 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 4

THIS SOLUTION IS NONFEASIBLE

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 2 .30000000E+01 2 .50000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 1 .10000000E+02

THE SOLUTION WITH 9 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.12286792E+02	2	.30000000E+01	3	.10119974E+01
	4	.98769671E+00				
CONS	1	.41472154E+00	2	.12286792E+02	3	.30000000E+01
	4	-.40350123E-02	5	-.25594790E-02	6	-.36670062E-02
	7	-.42788207E-02	8	.20000000E+01	9	.22867919E+01

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 323

OUT OF THESE 18 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 5

THIS IS A DISCRETE SOLUTION

THIS SOLUTION IS NONFEASIBLE

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 2 .50000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 2 .50000000E+01 1 .10000000E+02

THE SOLUTION WITH 9 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .10000000E+02    2  .50000000E+01    3  .10118094E+01
         4  .98814629E+00

CONS     1  .30000000E+00    2  .10000000E+02    3  .50000000E+01
         4 -.25752688E-02    5 -.24645202E-02    6 -.54398358E-03
         7 -.63253260E-03    8 0.                9 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 334

OUT OF THESE 11 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 6

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 2 .10000000E+02

THE SOLUTION WITH 7 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .39886960E+01   2  .10000000E+02   3  .10072803E+01
         4  .99142085E+00

CONS     1  .35070850E+00   2  .39886960E+01   3  .10000000E+02
         4 -.26290081E-12   5  .58317795E-11   6  .11979395E-01
         7  .93817361E-02
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 420

OUT OF THESE 86 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 7

THIS SOLUTION IS NONFEASIBLE

THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.LE. 1 .30000000E+01

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

X.GE. 2 .10000000E+02

THE SOLUTION WITH 8 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .30000000E+01    2  .12429826E+02    3  .10076222E+01
         4  .99210557E+00

CONS     1  .41378498E+00    2  .30000000E+01    3  .12429826E+02
         4 -.29768718E-02    5 -.43351433E-02    6 -.32734137E-02
         7 -.38179057E-02    8  .24298258E+01
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 436

OUT OF THESE 16 WERE PERFORMED AT THIS NODE

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 8

THIS IS A DISCRETE SOLUTION

THIS SOLUTION IS NONFEASIBLE

THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS AT THIS NODE ARE

```
X.GE.    1  .50000000E+01    2  .10000000E+02
```

THE SOLUTION WITH 8 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .50000000E+01    2  .10000000E+02    3  .10083695E+01
         4  .99201105E+00

CONS     1  .30000000E+00    2  .50000000E+01    3  .10000000E+02
         4 -.25159962E-02    5 -.25236151E-02    6 -.96316661E-07
         7  .76093945E-03    8 0.
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 447

OUT OF THESE 11 WERE PERFORMED AT THIS NODE

CHAPTER 4

UNDERSTANDING DISOPT3

The objective of this Chapter is to familiarize the user with the main concepts used in developing DISOPT3.

There are eight subroutines in this program, in addition to the main program and subroutine FUN which are supplied by the user. Figure 4 shows the calling sequence for these subroutines.

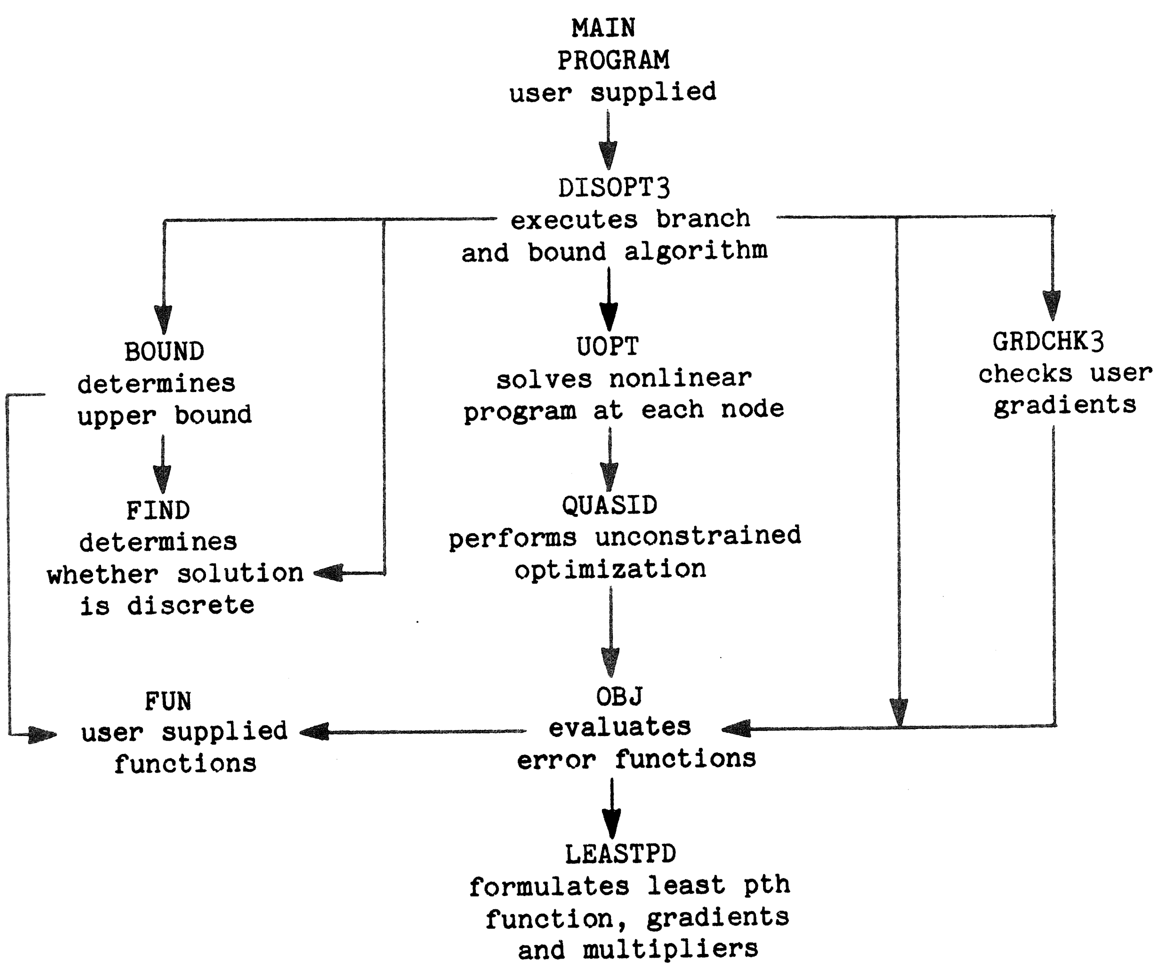


Fig. 4 Calling sequence for all the subroutines. A → B implies that subroutine B is called from subroutine A.

Having taken an overall view of the program, the next step is to understand what each subroutine is doing. This may be achieved by going through the program listing in Appendix 1. A large number of comments have been included in the program listing to facilitate an easy grasp of the logic. In addition, many of the key ideas are further discussed in this Chapter.

1. <u>Use of pointers to arrays DIS, IAR and X</u>

Each subroutine of this program uses many arrays but the user is not burdened with the task of dimensioning each one in the main program. Apart from avoiding inconvenience, this reduces the risk of making an error. Enough storage is reserved in the main program by suitably dimensioning arrays DIS, IAR and X. All the other arrays are accomodated into this space. Consider the following example.

```
MAIN PROGRAM                      SUBROUTINE XYZ (B)
DIMENSION A(100)                  DIMENSION B(1)
     .                                 .
     .
CALL XYZ (A(50))                       .
     .
     .                                 .
STOP                              RETURN
END                               END
```

When CALL XYZ (A(50)) is executed in the main program the control passes to the subroutine and the following equivalence is established between arrays A and B:

$A(50) \equiv B(1)$, $A(51) \equiv B(2)$, etc.

The number 50 may be thought of as a pointer for array B. In DISOPT3, the pointers are easily identified; for array RMULT, the pointer is LRMULT, for AL the pointer is LAL, etc. The idea of pointers has been extensively used. LASTDIS and LASTIAR are special pointers because they point to the first available elements in the respective arrays.

2. Storage of the essential information about the discrete variables

For each discrete variable two elements of array IAR are used to store (1) the number of available values for this discrete variable and (2) the pointer to the first value in array DIS. For example, the number of available values for the first discrete variable is IAR(1) and DIS (IAR(2)) is the first such value; for the second discrete variable these figures are IAR(3) and DIS(IAR(4)); and so on. This information is generated to make the data in array DIS readily accessible to subroutine FIND.

3. Implementation of the branch and bound algorithm

The use of Dakin's branch and bound algorithm involves addition and fathoming of nodes. Addition of a node to the tree is necessitated by the fact that X(I), a variable of the problem, is required to be discrete in the optimal solution but is currently not discrete. Adding a node is equivalent to adding another constraint to the original problem. This constraint is either X(I) $\leq$ XL or X(I) $\geq$ XU where XL and XU are respectively the nearest lower and the nearest upper discrete values of X(I).

In this program, whenever a new node is to be added to the tree, first the constraint X(I) $\leq$ XL is added to the original problem. This is accompanied by the following three steps:

(1) four consecutive elements of IAR are defined. See Figure 5,

(2) NODE is incremented by 1, and

(3) LASTIAR is incremented by 4.

The information, thus generated, is used whenever the new constraint or its gradient vector are evaluated in the subsequent optimizations.

If the node added above is to be fathomed, the 0 in the first element of IAR is changed to 1, indicating that constraint X(I) $\geq$ XU is added, and another optimization is performed. To fathom this kind of node, simply, the following three steps are performed:

(1) NODE is decremented by 1,

(2) pointer LASTIAR is decremented by 4, and

(3) pointer LASTDIS is decremented by (N+2).

Variable NODE, at any time, equals the number of additional constraints in the problem. The record of those constraints which have been discarded is not preserved.

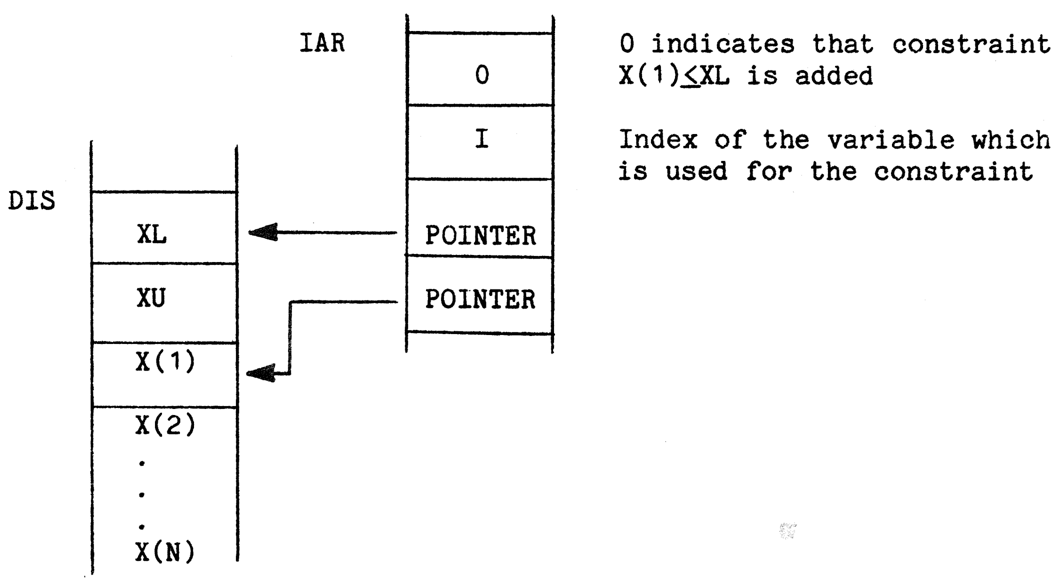


Fig. 5 Updating IAR to add a node

4. <u>Optimization with some variables held fixed</u>

Subroutines LEASTPD, QUASID and UOPT are set up in such a way that it is possible to perform an optimization with fewer variables than in the original problem. Two steps are needed for it:

(1) N, the number of variables in the problem, is suitably reinitialized before performing the optimization, and

(2) IDVAR(1), ..., IDVAR(N) contain the indices of the variables to be included in the optimization.

In this program, this particular feature is being used to hold only one variable fixed. When HOLDVAR is TRUE, the variable X(I) of the additional constraint is held constant at value XL (or XU) in the subsequent optimization at that node. N and NODE are temporarily decremented by 1 and their values are restored after the optimization. This point is clarified as one goes through the program listing.

5. <u>The feature of ONE or ALL discrete optimal solutions</u>

If only one optimal discrete solution is required, this program, after finding one discrete solution, tries not to search those nodes that are likely to yield, at the best, an equally good solution only. This saves effort because many nodes which otherwise would have been searched are not searched now. It is actually accomplished by decreasing the upper bound, once it has been found, by a small quantity of the order 10^{-6} for the purposes of checking feasibility or fathoming nodes.

If all the optimal discrete solutions are required, a precaution is taken against fathoming a node that might yield an equally good solution after one discrete solution has been found. This is achieved by increasing the upper bound by a small quantity (-TOLCONS) for the purposes of checking feasibility or fathoming nodes.

6. Subroutine DISOPT3

This is the main subroutine of this package. It executes the branch and bound algorithm and performs the necessary optimizations in order to find the optimal discrete solutions.

The logic followed in the subroutine is best explained by the flow charts in Figure 6 and Figure 7.

7. Feasibility check

The algorithm used in this program to solve the nonlinear programming problem at every node does not stop without expending a lot of effort if there is no feasible solution to the problem. Hence, to use it efficiently, prior to its application, a feasibility check is made to ensure the existence of a feasible point.

The feasibility check involves a least pth optimization with p = 2 as done by Chen [5]. It is based on the argument that if the optimal value of the least pth function is positive for some value of p then it can not be negative for some other value of p. The aim of the feasibility check is to determine the existence of a feasible point with objective function less than its current upper bound.

8. Continuous solution of the nonlinear programming problem

This solution is required at every node. The algorithm employed in subroutine UOPT to obtain this solution is the one proposed by Charalambous [2]. The implementation of this algorithm in subroutine

UOPT can best be understood by the flow chart in Figure 8.

Some features of the implementation which differ from the proposed original algorithm [2] are as follows.

(1) Before attempting to solve the problem, a feasibility check is made as explained in the previous section.

(2) If the result of an iteration is a nonfeasible point, the starting point for the next iteration is not this point but the best available feasible point.

(3) The scheme for choosing active constraints based on the multipliers is used only after two iterations have successfully led to a feasible solution.

(4) Should the above scheme, at any stage, lead to a nonfeasible point it (the above scheme) is not used again and the alpha parameters corresponding to all the constraints revert to their values just prior to the initiation of the reduction scheme.

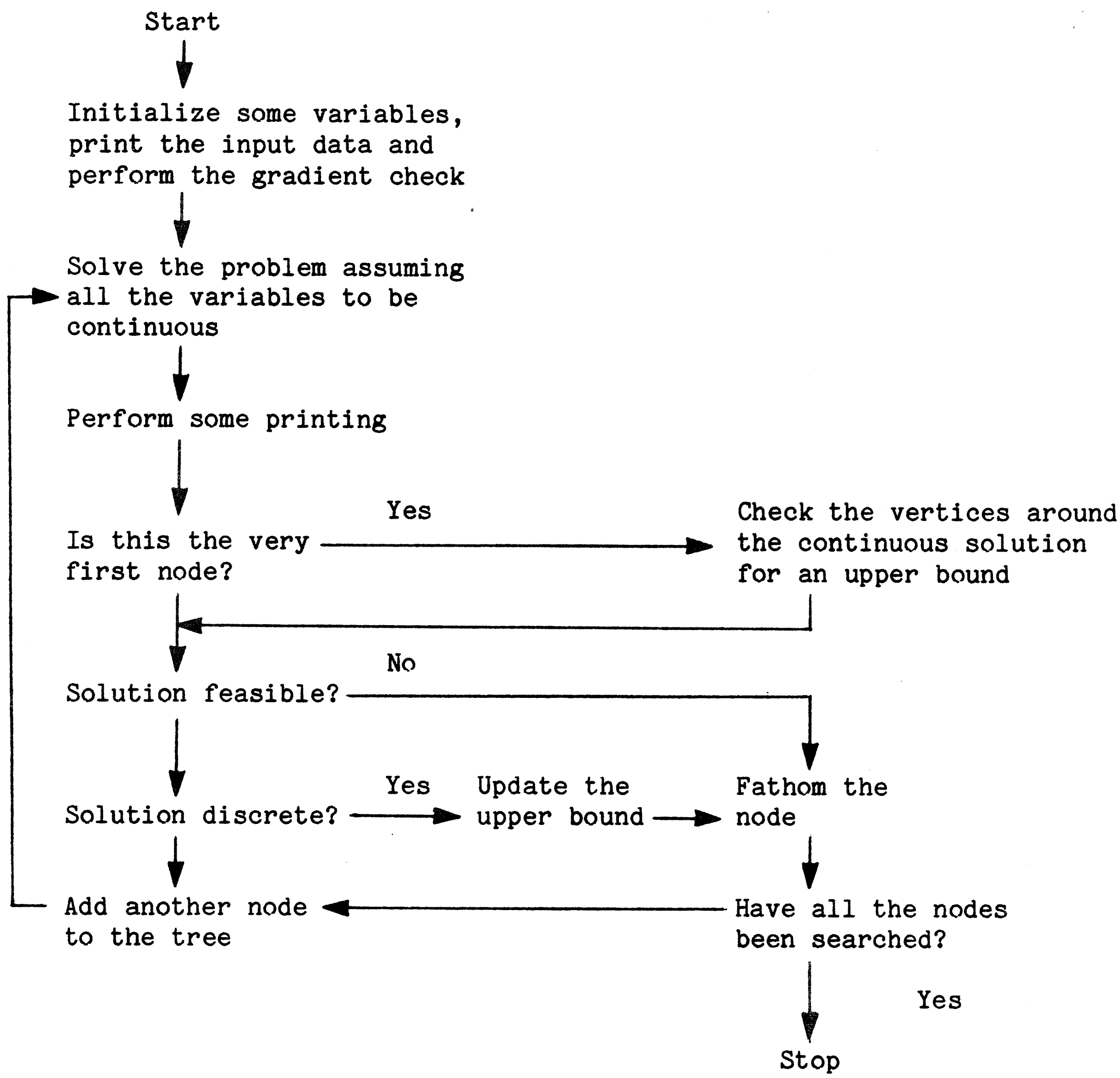
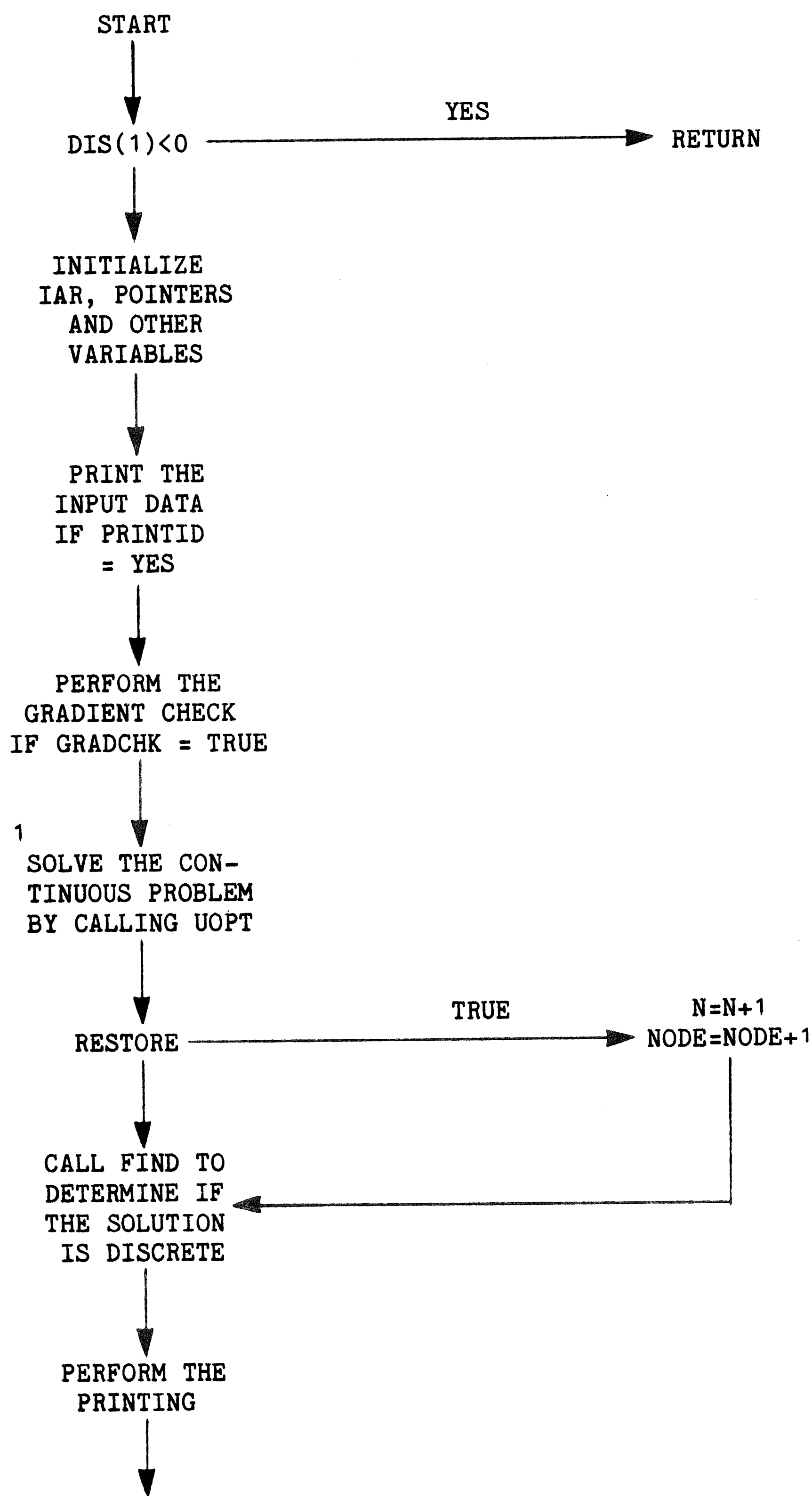
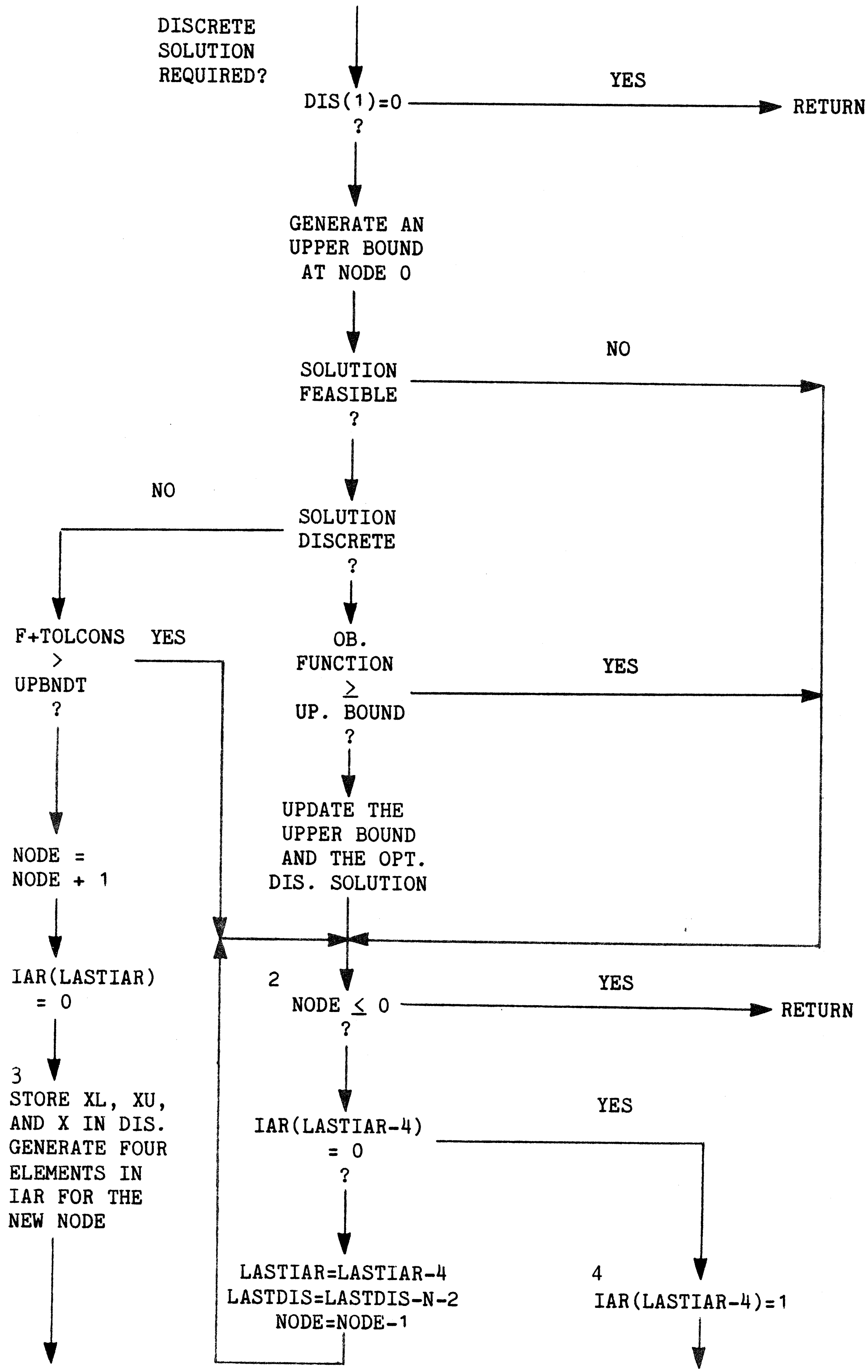


Fig. 6 Basic logical structure of subroutine DISOPT3.


START
YES
DIS(1)<0
RETURN
INITIALIZE
IAR, POINTERS
AND OTHER
VARIABLES
PRINT THE
INPUT DATA
IF PRINTID
= YES
PERFORM THE
GRADIENT CHECK
IF GRADCHK = TRUE
1
SOLVE THE CON-
TINUOUS PROBLEM
BY CALLING UOPT
TRUE
N=N+1
NODE=NODE+1
RESTORE
CALL FIND TO
DETERMINE IF
THE SOLUTION
IS DISCRETE
PERFORM THE
PRINTING


DISCRETE SOLUTION REQUIRED?
DIS(1)=0 ?
YES
RETURN
GENERATE AN UPPER BOUND AT NODE 0
SOLUTION FEASIBLE ?
NO
SOLUTION DISCRETE ?
NO
F+TOLCONS > UPBNDT ?
YES
OB. FUNCTION ≥ UP. BOUND ?
YES
NODE = NODE + 1
UPDATE THE UPPER BOUND AND THE OPT. DIS. SOLUTION
IAR(LASTIAR) = 0
2
NODE ≤ 0 ?
YES
RETURN
3
STORE XL, XU, AND X IN DIS. GENERATE FOUR ELEMENTS IN IAR FOR THE NEW NODE
IAR(LASTIAR-4) = 0 ?
YES
LASTIAR=LASTIAR-4
LASTDIS=LASTDIS-N-2
NODE=NODE-1
4
IAR(LASTIAR-4)=1

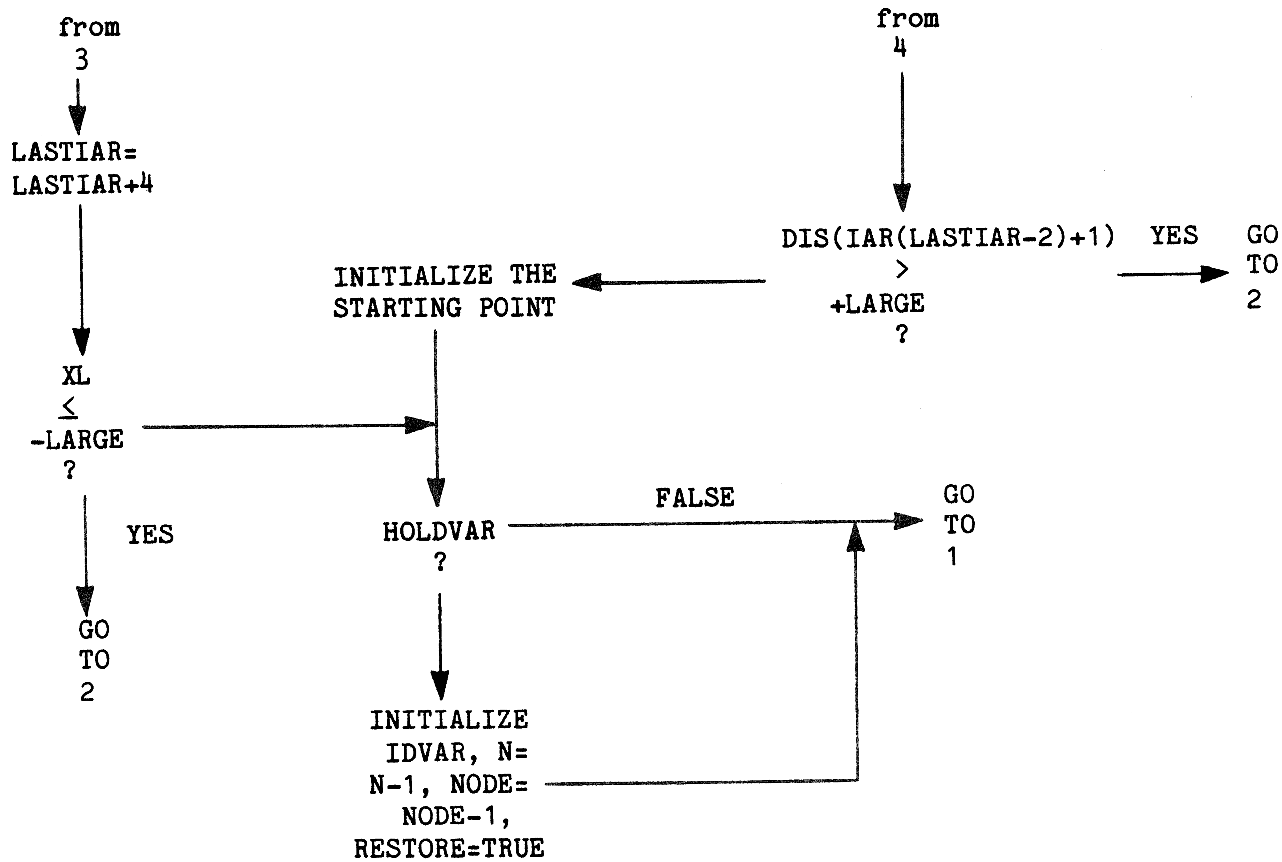


Fig. 7 Flow chart for subroutine DISOPT3.

Start

↓

Check the existence of a feasible point with objective function less than or equal to the current upper bound. This is achieved by a least pth optimization with p = 2 and error functions: f-UPBND, $-g_1$, $-g_2$, etc.

↓

Does a feasible point exist? — No → Return

↓

ALMAX = ALMIN
ALMIN is user supplied alpha parameter

↓

AL(1) = 0 and for I = 2 to N AL(I) = ALMAX.
Vector AL is used for converting the constrained problem into a minimax problem, e.g., min f s.t. $g_1 \geq 0$, $g_2 \geq 0$, is equivalent to min max (f, f-AL(2)g_1, f-AL(3)g_2)

↓

Initialize the starting point to the best feasible point

↓

Determine HEXI, the artificial margin ξ of the Charalambous algorithm.
HEXI=min(0,ERMAX+10^{-10}) where ERMAX is the maximum of the error functions at this starting point

↓

1
Call Fletcher's subroutine for the unconstrained optimization

↓

Are all the constraints satisfied? — No (Loop 1) → ALMAX=ALMAX*10 only if loop 2 has not been executed more than once (→ back to AL(1) = 0 ...)

↓ (Loop 2)

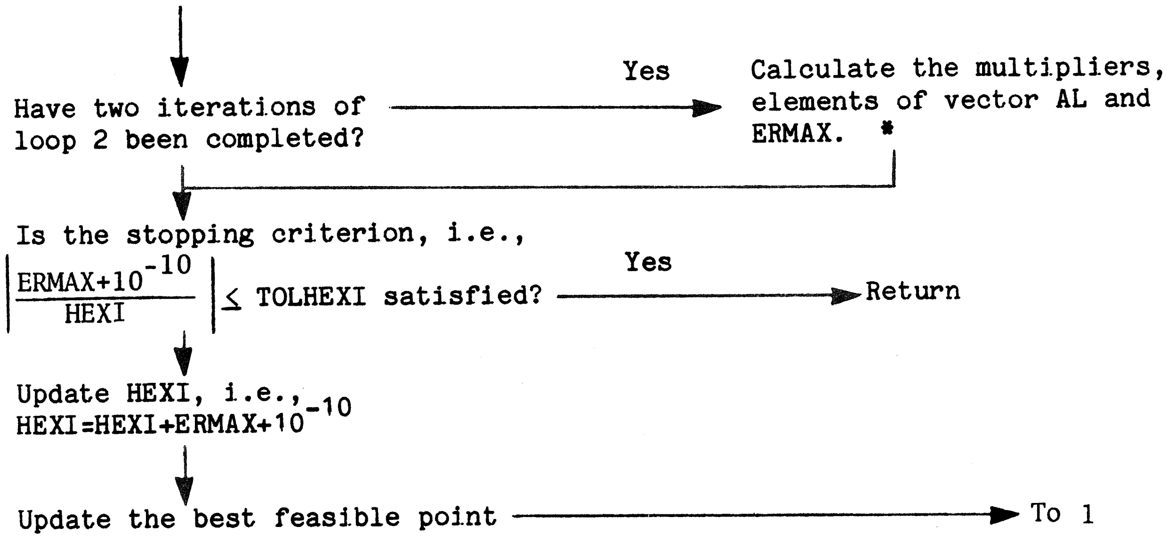


Fig. 8 Charalambous algorithm as implemented in subroutine UOPT.

* The objective function is always active. Other constraints with multipliers exceeding TOLMULT are active. AL(I) for a constraint is its multiplier times the number of active functions.

CHAPTER 5

SOME RESULTS WITH DISOPT3

Discrete optimization involves solution of the nonlinear programming problem at many nodes. Hence, it is of crucial importance to correctly code the many details of the algorithm which finds this solution. This kind of verification has been made by repeating here the example problems presented by Charalambous [2]. Examples 4, 5, 6 and 7 at the end of this Chapter present a comparison between the results obtained by Charalambous and those obtained by DISOPT3. A complete listing of the main program, subroutine FUN and the output is also included for each example.

Examples 1, 2 and 3 were also solved by DISOPT, the old program [5]. A comparison between the results is made in Table V.

The results obtained with Examples 1, 2 and 3 using the different options of DISOPT3 are summarized in Table VI.

Before deciding whether certain features, which do not exactly have a theoretical basis, should be chosen for this program or rejected, a test was made with Examples 1, 2 and 3. The results are summarized in Table VII.

TABLE V COMPARISON BETWEEN THE RESULTS OF DISOPT3/DISOPT

Description	Example 1	Example 2	Example 3
Number of function evaluations	396/518[1]	572/-[2]	447/590
Execution time with full printing	1.9/1.8	3.2/-	3.3/4.0
Execution time with no printing	0.8/1.4	1.6/-	1.9/3.2

1 Algorithm 3 of DISOPT was used. This example could not be solved with Algorithm 4 of DISOPT.

2 This example, for some reason, could not be solved with any algorithm.

TABLE VI PERFORMANCE OF DISOPT3 WITH DIFFERENT OPTIONS

Feature	Number of function evaluations		
	Example 1	Example 2	Example 3
HOLDVAR = TRUE/FALSE	368/368	572/808	447/774
ONESOL = TRUE/FALSE	370/368	515/572	452/447
REVERSE = TRUE/FALSE	655/368	384/572	447/494
VERTCHK = TRUE/FALSE	368/581	572/788	447/447

TABLE VII RESULTS OF SOME TESTS WITH DISOPT3

Feature	Number of function evaluations		
	Example 1	Example 2	Example 3
Original program	368	593	447
With HEXI supplied at each node except node 0	367	599	443
Starting point not initialized at each node	527	607	544

Example 4: The Beale problem [12]

Minimize

$$f = 9 - 8x_1 - 6x_2 - 4x_3 + 2x_1^2 + 2x_2^2 + x_3^2 + 2x_1x_2 + 2x_1x_3$$

subject to

$$x_1 \geq 0$$

$$x_2 \geq 0$$

$$x_3 \geq 0$$

$$3 - x_1 - x_2 - 2\,x_3 \geq 0$$

The optimal solution is

$$f = 0.1111\ 1111$$

$$x_1 = 1.3333\ 3333$$

$$x_2 = 0.7777\ 7777$$

$$x_3 = 0.4444\ 4444$$

The results obtained by DISOPT3 are consistent with the results presented by Charalambous [2]. A comparison is made in Table VIII. A complete listing of the main program, subroutine FUN and the output is also presented.

TABLE VIII COMPARISON BETWEEN THE RESULTS OF CHARALAMBOUS/DISOPT3 ON EXAMPLE 4

Iteration number	1	2	3
Objective function	0.114392 0.114392	0.1111967 0.1111967	0.11111111 0.11111111
x_1	1.338218 1.338219	1.333462 1.333462	1.333333 1.333333
x_2	0.7745206 0.7745207	0.7776922 0.7776922	0.777778 0.777777
x_3	0.4363018 0.4363018	0.4442303 0.4442304	0.4444446 0.4444444
Function evaluations	15 14	13 20	16 15
RMULT(2)	0 X	0 0	0 0
RMULT(3)	0 X	0 0	0 0
RMULT(4)	0 X	0 0	0 0
RMULT(5)	0.2255 X	0.2223 0.2223	0.2222 0.2222

Total number of function evaluations 44/49

X: Not calculated by DISOPT3

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 4
C
      DIMENSION IAR(60), X(100), XD(3)
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
C
      DATA X/3*.5/,DIS,N,NORCONS/0.,3,5/
      DATA ALMIN,IP/1.,10/,TOLCONS,TOLHEXI,TOLMULT/-1.E-2,.001,1.E-4/
C
      PRINTP=3HALL
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE BEALE PROBLEM
C
      DIMENSION CONS(5), GCONS(3,5), IDCONS(1), X(3)
C
      COMMON /7/ IFN,IND1,IND2
C
      DO 60 I=1,5
      J=IDCONS(I)
      GO TO (10,20,30,40,50,60), J
 10   CONS(1)=9.-8.*X(1)-6.*X(2)-4.*X(3)+2.*(X(1)**2+X(2)**2)+X(3)**2+2.
     1*X(1)*(X(2)+X(3))
      GCONS(1,1)=-8.+4.*X(1)+2.*(X(2)+X(3))
      GCONS(2,1)=-6.+4.*X(2)+2.*X(1)
      GCONS(3,1)=-4.+2.*X(3)+2.*X(1)
      GO TO 60
 20   CONS(2)=X(1)
      GCONS(1,2)=1.
      GCONS(2,2)=0.
      GCONS(3,2)=0.
      GO TO 60
 30   CONS(3)=X(2)
      GCONS(1,3)=0.
      GCONS(2,3)=1.
      GCONS(3,3)=0.
      GO TO 60
 40   CONS(4)=X(3)
      GCONS(1,4)=0.
      GCONS(2,4)=0.
      GCONS(3,4)=1.
      GO TO 60
 50   CONS(5)=3.-X(1)-X(2)-2.*X(3)
      GCONS(1,5)=-1.
      GCONS(2,5)=-1.
      GCONS(3,5)=-2.
 60   CONTINUE
C
      IFN=IFN+1
      RETURN
      END
```

INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISOPT3

```
INITIAL VALUE OF THE ELEMENTS OF AL ... ALMIN =  .10000000E+01
OPTIMAL OBJECTIVE AT NODE 0 (GUESS) ..... EST = 0.
VALUE OF PARAMETER P .................... IP =     10
(-LARGE,LARGE) BRACKETS ALL VARIABLES . LARGE =  .10000000E+11
ALLOWED FUNCTION CALLS AT EACH NODE .. MAXIFN =  1000
ALLOWED QUASID CALLS AT EACH NODE .... MAXITN =    15
ALLOWED NUMBER OF NODES ............. MAXNODE =  1000
NUMBER OF DISCRETE VARIABLES ........... NDIS =     0
NUMBER OF CONSTRAINTS IN THE PROBLEM  NORCONS =     5
NUMBER OF UNIFORM STEP VARIABLES ....... NUNI =     0
TOLERANCE FOR THE CONSTRAINTS ....... TOLCONS = -.10000000E-01
TOLERANCE FOR THE DISCRETE VARIABLES . TOLDIS =  .10000000E-02
STOPPING CRITERION FOR UOPT ......... TOLHEXI =  .10000000E-02
TOLERANCE FOR THE MULTIPLIERS ....... TOLMULT =  .10000000E-03
STOPPING CRITERION FOR QUASID .......... TOLX =  .10000000E-06
INITIAL VALUE OF THE UPPER BOUND ...... UPBND =  .10000000E+11
STARTING POINT FOR THIS PROBLEM ........... X 1  .50000000E+00
                                              2  .50000000E+00
                                              3  .50000000E+00
```

OPTIONS IN EFFECT

GRADIENT CHECK AT THE STARTING POINT

ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION

VERTICES AROUND NODE 0 SOLUTION EXAMINED

DETAILED PRINTING REQUESTED

GRADIENT CHECK AT THE STARTING POINT

ANALYTICAL GRADIENT VECTOR G(I)		NUMERICAL GRADIENT VECTOR Y(I)		PERCENTAGE ERROR VECTOR PERCENT(I)	
1	-.44054831E+01	1	-.44054831E+01	1	.28357304E-07
2	-.33244442E+01	2	-.33244442E+01	2	.44638948E-07
3	-.22392689E+01	3	-.22392689E+01	3	.31051971E-07

THE GRADIENTS APPEAR TO BE CORRECT

FEASIBILITY CHECK AT NODE 0

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	-.27735010E+00	1	.50000000E+00	1	-.14934236E+00
			2	.50000000E+00	2	-.14934236E+00
			3	.50000000E+00	3	-.12800774E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.22500000E+01
	2	.10000000E+01	2	.50000000E+00
	3	.10000000E+01	3	.50000000E+00
	4	.10000000E+01	4	.50000000E+00
	5	.10000000E+01	5	.10000000E+01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	.23000048E+01	1	.50000000E+00	1	-.44054831E+01
			2	.50000000E+00	2	-.33244442E+01
			3	.50000000E+00	3	-.22392689E+01
11	16	.11700904E+00	1	.13382190E+01	1	.76122855E-08
			2	.77452070E+00	2	-.10786708E-07
			3	.43630175E+00	3	.17279807E-07

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .11439206E+00

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.11439206E+00
	2	.10000000E+01	2	.13382190E+01
	3	.10000000E+01	3	.77452070E+00
	4	.10000000E+01	4	.43630175E+00
	5	.10000000E+01	5	.14656849E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	18	-.99999564E-10	1	.13382190E+01	1	-.22547930E+00
			2	.77452070E+00	2	-.22547931E+00
			3	.43630175E+00	3	-.45095859E+00
8	37	-.31077454E-02	1	.13334618E+01	1	.42725818E-06
			2	.77769216E+00	2	.42647787E-06
			3	.44423040E+00	3	.86862583E-06

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .11119674E+00

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.11119674E+00
2	.11330150E-28	2	INACTIVE	2	.13334618E+01
3	.41877684E-26	3	INACTIVE	3	.77769216E+00
4	.19165727E-23	4	INACTIVE	4	.44423040E+00
5	.22230830E+00	5	.44461660E+00	5	.38527610E-03

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	41	-.99999564E-10	1	.13334618E+01	1	-.22230785E+00
			2	.77769216E+00	2	-.22230785E+00
			3	.44423040E+00	3	-.44461568E+00
7	55	-.79898885E-04	1	.13333333E+01	1	.88590388E-10
			2	.77777777E+00	2	.88165929E-10
			3	.44444444E+00	3	.17670989E-09

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

THE SOLUTION WITH 5 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

```
   X     1  .13333333E+01   2  .77777777E+00   3  .44444444E+00

CONS     1  .11111111E+00   2  .13333333E+01   3  .77777777E+00
         4  .44444444E+00   5  .13558342E-07
```

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 66

OUT OF THESE 59 WERE PERFORMED AT THIS NODE

Example 5: The Rozen-Suzuki problem [12]

Minimize

$$f = x_1^2 + x_2^2 + 2x_3^2 + x_4^2 - 5x_1 - 5x_2 - 21x_3 + 7x_4$$

subject to

$$-x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 + 8 \geq 0$$

$$-x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 + 10 \geq 0$$

$$-2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 + 5 \geq 0$$

The optimal solution is

$$f = -44.0$$

$$x_1 = 0.0$$

$$x_2 = 1.0$$

$$x_3 = 2.0$$

$$x_4 = -1.0$$

The results obtained by DISOPT3 are consistent with the results presented by Charalambous [2]. A comparison is made in Table IX. A complete listing of the main program, subroutine FUN and the output is also presented.

TABLE IX COMPARISON BETWEEN THE RESULTS OF CHARALAMBOUS/DISOPT3 ON EXAMPLE 5

Iteration number	1	2	3	4	5
Objective function	-65.84928 -65.84928	-42.174263 -42.174270	-43.924003 -43.924003	-43.999954 -43.999954	-44.000000 -44.000000
x_1	0.930565 0.930564	-0.007559 -0.007559	-0.0006666 -0.0006666	0.000002276 0.000002144	0.000000 0.000000
x_2	1.277804 1.277803	0.9498981 0.9498988	0.9991067 0.9991073	0.999999 1.000000	1.000000 1.000000
x_3	3.469368 3.469368	1.926183 1.926184	1.996514 1.996514	1.999999 1.999999	2.000000 2.000000
x_4	-1.569321 -1.569320	-0.8704463 -0.8704450	-0.9950777 -0.9950778	-0.999991 -0.999991	-1.000000 -1.000000
Function evaluations	11 15	21 21	19 29	19 19	29 25
RMULT(2)	0.39480 X	1.02973 X	1.00630 1.00630	0.99996 1.00002	1.000620 0.999733
RMULT(3)	0.26327 X	0.08116 X	0 0	0 0	0 0
RMULT(4)	0.31152 X	2.07300 X	2.00086 2.00086	1.99994 1.99998	1.99657 2.00046

Total number of function evaluations 99/109

X: Not calculated by DISOPT3

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 5
C
      DIMENSION IAR(75), X(115), XD(4)
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
C
      DATA X/4*0./,DIS,N,NORCONS/0.,4,4/
      DATA ALMIN,IP/1.,10/,TOLCONS,TOLHEXI,TOLMULT/-1.E-2,.001,1.E-4/
C
      PRINTP=3HALL
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE ROSEN-SUZUKI PROBLEM
C
      DIMENSION CONS(4), GCONS(4,4), IDCONS(1), X(4)
C
      COMMON /7/ IFN,IND1,IND2
C
      A=X(1)**2+X(3)**2
      B=X(2)**2+X(4)**2
C
      DO 50 I=1,4
      J=IDCONS(I)
      GO TO (10,20,30,40,50), J
 10   CONS(1)=A+B+X(3)**2-5.*(X(1)+X(2))-21.*X(3)+7.*X(4)
      GCONS(1,1)=2.*X(1)-5.
      GCONS(2,1)=2.*X(2)-5.
      GCONS(3,1)=4.*X(3)-21.
      GCONS(4,1)=2.*X(4)+7.
      GO TO 50
 20   CONS(2)=-(A+B)-X(1)+X(2)-X(3)+X(4)+8.
      GCONS(1,2)=-(2.*X(1)+1.)
      GCONS(2,2)=-2.*X(2)+1.
      GCONS(3,2)=-(2.*X(3)+1.)
      GCONS(4,2)=-2.*X(4)+1.
      GO TO 50
 30   CONS(3)=-A-B*2.+X(1)+X(4)+10.
      GCONS(1,3)=-2.*X(1)+1.
      GCONS(2,3)=-4.*X(2)
      GCONS(3,3)=-2.*X(3)
      GCONS(4,3)=-4.*X(4)+1.
      GO TO 50
 40   CONS(4)=-A-(X(1)**2+X(2)**2)-X(1)*2.+X(2)+X(4)+5.
      GCONS(1,4)=(2.*X(1)+1.)*(-2.)
      GCONS(2,4)=-2.*X(2)+1.
      GCONS(3,4)=-2.*X(3)
      GCONS(4,4)=1.
 50   CONTINUE
C
      IFN=IFN+1
      RETURN
      END
```

INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISOPT3

```
INITIAL VALUE OF THE ELEMENTS OF AL ... ALMIN =   .10000000E+01
OPTIMAL OBJECTIVE AT NODE 0 (GUESS) ..... EST = 0.
VALUE OF PARAMETER P ..................... IP =     10
(-LARGE,LARGE) BRACKETS ALL VARIABLES . LARGE =   .10000000E+11
ALLOWED FUNCTION CALLS AT EACH NODE .. MAXIFN =   1000
ALLOWED QUASID CALLS AT EACH NODE .... MAXITN =     15
ALLOWED NUMBER OF NODES ............. MAXNODE =   1000
NUMBER OF DISCRETE VARIABLES ........... NDIS =      0
NUMBER OF CONSTRAINTS IN THE PROBLEM  NORCONS =      4
NUMBER OF UNIFORM STEP VARIABLES ....... NUNI =      0
TOLERANCE FOR THE CONSTRAINTS ....... TOLCONS = -.10000000E-01
TOLERANCE FOR THE DISCRETE VARIABLES . TOLDIS =   .10000000E-02
STOPPING CRITERION FOR UOPT ......... TOLHEXI =   .10000000E-02
TOLERANCE FOR THE MULTIPLIERS ....... TOLMULT =   .10000000E-03
STOPPING CRITERION FOR QUASID .......... TOLX =   .10000000E-06
INITIAL VALUE OF THE UPPER BOUND ...... UPBND =   .10000000E+11
STARTING POINT FOR THIS PROBLEM ........... X 1 0.
                                              2 0.
                                              3 0.
                                              4 0.
```

OPTIONS IN EFFECT

GRADIENT CHECK AT THE STARTING POINT

ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION

VERTICES AROUND NODE 0 SOLUTION EXAMINED

DETAILED PRINTING REQUESTED

GRADIENT CHECK AT THE STARTING POINT

ANALYTICAL GRADIENT VECTOR G(I)		NUMERICAL GRADIENT VECTOR Y(I)		PERCENTAGE ERROR VECTOR PERCENT(I)	
1	-.50000000E+01	1	-.50000000E+01	1	.17053026E-11
2	-.50000000E+01	2	-.50000000E+01	2	.17053026E-11
3	-.21000000E+02	3	-.21000000E+02	3	.16240977E-11
4	.70000000E+01	4	.70000000E+01	4	.12180733E-11

THE GRADIENTS APPEAR TO BE CORRECT

FEASIBILITY CHECK AT NODE 0

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION	VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)	
0	1	-.39036003E+01	1	0.	1	.10084301E+01
			2	0.	2	-.59204604E+00
			3	0.	3	.11617858E+00
			4	0.	4	-.65152948E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	0.
	2	.10000000E+01	2	.80000000E+01
	3	.10000000E+01	3	.10000000E+02
	4	.10000000E+01	4	.50000000E+01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	-.10000000E-09	1	0.	1	-.50000000E+01
			2	0.	2	-.50000000E+01
			3	0.	3	-.21000000E+02
			4	0.	4	.70000000E+01
10	17	-.47415087E+02	1	.93056449E+00	1	.21316345E-09
			2	.12778030E+01	2	-.80748610E-09
			3	.34693676E+01	3	-.55626242E-09
			4	-.15693200E+01	4	.14464961E-08

THE ABOVE ITERATION HAS RESULTED IN A NONFEASIBLE SOLUTION. THE CONSTRAINTS AT THIS POINT ARE GIVEN AS FOLLOWS. IT MAY BE NOTED THAT THE STARTING POINT FOR THE NEXT ITERATION IS NOT THE ABOVE SOLUTION BUT THE BEST FEASIBLE POINT OBTAINED SO FAR

CONS 1 -.65849278E+02 2 -.13689457E+02 3 -.11732309E+02
4 -.12553838E+02

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	0.
	2	.10000000E+02	2	.80000000E+01
	3	.10000000E+02	3	.10000000E+02
	4	.10000000E+02	4	.50000000E+01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	20	-.10000000E-09	1	0.	1	-.50000000E+01
			2	0.	2	-.50000000E+01
			3	0.	3	-.21000000E+02
			4	0.	4	.70000000E+01
16	40	-.40402268E+02	1	-.75588980E-02	1	-.10069569E-05
			2	.94989878E+00	2	-.78902214E-06
			3	.19261840E+01	3	-.25455740E-05
			4	-.87044502E+00	4	.59874652E-06

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = -.42174270E+02

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	-.42174270E+02
	2	.10000000E+02	2	.79060447E+00
	3	.10000000E+02	3	.20917897E+01
	4	.10000000E+02	4	.48196477E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	42	-.10004442E-09	1	-.75588980E-02	1	-.50151178E+01
			2	.94989878E+00	2	-.31002024E+01
			3	.19261840E+01	3	-.13295264E+02
			4	-.87044502E+00	4	.52591100E+01
13	70	-.16810786E+01	1	-.66664066E-03	1	-.53847399E-06
			2	.99910734E+00	2	-.46040363E-08
			3	.19965135E+01	3	-.71224483E-06
			4	-.99507780E+00	4	.35558330E-07

ITERATION NUMBER 4 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = -.43924003E+02

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	-.43924003E+02
2	.10062983E+01	2	.30188950E+01	2	.33721020E-01
3	.38160508E-08	3	INACTIVE	3	.10413986E+01
4	.20008557E+01	4	.60025672E+01	4	.21080498E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	74	-.10004442E-09	1	-.66664066E-03	1	-.50013333E+01
			2	.99910734E+00	2	-.30017853E+01
			3	.19965135E+01	3	-.13013946E+02
			4	-.99507780E+00	4	.50098444E+01
12	92	-.68090337E-01	1	.21436360E-05	1	-.33346764E-05
			2	.10000004E+01	2	-.13851056E-05
			3	.19999991E+01	3	-.73984691E-05
			4	-.99999084E+00	4	.13328128E-05

```
ITERATION NUMBER  5 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)
-------------------------------------------------------------------
FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE   0
-------------------------------------------------

VALUE OF HEXI FOR THIS ITERATION ....... HEXI = -.43999954E+02

            MULTIPLIER            ALPHA              CONSTRAINT
          VECTOR RMULT( I)     VECTOR AL( I)       VECTOR CONS( I)

         1  ACTIVE           1  OBJECTIVE        1 -.43999954E+02
         2  .10000200E+01    2  .30000599E+01    2  .29557083E-04
         3  .38160508E-08    3  INACTIVE         3  .10000500E+01
         4  .19999753E+01    4  .59999259E+01    4  .81662286E-05

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD
-----------------------------------------------------------------

ITER. FUNC. LEAST PTH            VARIABLE            GRADIENT
 NO.  EVAL. FUNCTION           VECTOR X( I)         VECTOR G( I)

  0     96 -.10004442E-09   1  .21436360E-05    1 -.49999957E+01
                            2  .10000004E+01    2 -.29999992E+01
                            3  .19999991E+01    3 -.13000004E+02
                            4 -.99999084E+00    4  .50000183E+01

 11    120 -.41115391E-04   1  .33501431E-09    1  .58560547E-03
                            2  .10000000E+01    2  .17309175E-03
                            3  .20000000E+01    3  .45251095E-03
                            4 -.10000000E+01    4  .30608379E-03
```

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

```
THE SOLUTION WITH   4 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

   X      1  .33501431E-09    2  .10000000E+01    3  .20000000E+01
          4 -.10000000E+01

CONS      1 -.44000000E+02    2  .46179593E-09    3  .99999997E+00
          4 -.13756107E-09

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS  133

OUT OF THESE  124 WERE PERFORMED AT THIS NODE
```

Example 6: The Wong problem 1 [12]

Minimize

$$f = (x_1 - 10)^2 + 5(x_2 - 12)^2 + x_3^4 + 3(x_4 - 11)^2 + 10\, x_5^6 + 7\, x_6^2 + x_7^4 - 4\, x_6\, x_7 - 10\, x_6 - 8\, x_7$$

subject to

$$-2\, x_1^2 - 3\, x_2^4 - x_3 - 4\, x_4^2 - 5\, x_5 + 127 \geq 0$$

$$-7\, x_1 - 3\, x_2 - 10\, x_3^2 - x_4 + x_5 + 282 \geq 0$$

$$-23\, x_1 - x_2^2 - 6\, x_6^2 + 8\, x_7 + 196 \geq 0$$

$$-4\, x_1^2 - x_2^2 + 3\, x_1\, x_2 - 2\, x_3^2 - 5\, x_6 + 11\, x_7 \geq 0$$

The optimal solution is

$$f = 680.630$$

$x_1 = 2.3305$	$x_2 = 1.9514$
$x_3 = -0.47754$	$x_4 = 4.3657$
$x_5 = -0.62449$	$x_6 = 1.0381$
$x_7 = 1.5942$	

The results obtained by DISOPT3 are consistent with the results presented by Charalambous [2]. A comparison is made in Table X. A complete listing of the main program, subroutine FUN and the output is also presented.

TABLE X COMPARISON BETWEEN THE RESULTS OF CHARALAMBOUS/DISOPT3 ON EXAMPLE 6

Iteration number	1	2	3	4	5
Objective function	578.7389 578.7388	704.9796 704.9796	681.4550 681.4550	680.6358 680.6358	680.6301 680.6301
Function evaluations	24 26	29 25	30 58	30 27	42 27
Total number of function evaluations				155/163	

```
      PROGRAM TST( INPUT,OUTPUT,TAPE5= INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 6
C
      DIMENSION IAR(125), X(210), XD(7)
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
C
      DATA X/1.,2.,0.,4.,0.,1.,1./,DIS,N,NORCONS/0.,7,5/
      DATA ALMIN,IP/1.,10/,TOLCONS,TOLHEXI,TOLMULT/-1.E-2,.001,1.E-4/
C
      PRINTP=3HALL
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE FIRST WONG PROBLEM
C
      DIMENSION CONS(5), GCONS(7,5), IDCONS(1), X(7)
C
      COMMON /7/ IFN,IND1,IND2
C
      DO 60 I=1,5
      J=IDCONS(I)
      GO TO (10,20,30,40,50,60), J
10    CONS(1)=(X(1)-10.)**2+5.*(X(2)-12.)**2+X(3)**4+3.*(X(4)-11.)**2+10
     1.*X(5)**6+7.*X(6)**2+X(7)**4-4.*X(6)*X(7)-10.*X(6)-8.*X(7)
      GCONS(1,1)=2.*(X(1)-10.)
      GCONS(2,1)=10.*(X(2)-12.)
      GCONS(3,1)=4.*X(3)**3
      GCONS(4,1)=6.*(X(4)-11.)
      GCONS(5,1)=60.*X(5)**5
      GCONS(6,1)=14.*X(6)-4.*X(7)-10.
      GCONS(7,1)=4.*X(7)**3-4.*X(6)-8.
      GO TO 60
20    CONS(2)=-2.*X(1)**2-3.*X(2)**4-X(3)-4.*X(4)**2-5.*X(5)+127.
      GCONS(1,2)=-4.*X(1)
      GCONS(2,2)=-12.*X(2)**3
      GCONS(3,2)=-1.
      GCONS(4,2)=-8.*X(4)
      GCONS(5,2)=-5.
      GCONS(6,2)=0.
      GCONS(7,2)=0.
      GO TO 60
30    CONS(3)=-7.*X(1)-3.*X(2)-10.*X(3)**2-X(4)+X(5)+282.
      GCONS(1,3)=-7.
      GCONS(2,3)=-3.
      GCONS(3,3)=-20.*X(3)
      GCONS(4,3)=-1.
      GCONS(5,3)=1.
      GCONS(6,3)=0.
      GCONS(7,3)=0.
      GO TO 60
40    CONS(4)=-23.*X(1)-X(2)**2-6.*X(6)**2+8.*X(7)+196.
      GCONS(1,4)=-23.
      GCONS(2,4)=-2.*X(2)
      GCONS(3,4)=0.
      GCONS(4,4)=0.
      GCONS(5,4)=0.
      GCONS(6,4)=-12.*X(6)
      GCONS(7,4)=8.
      GO TO 60
50    CONS(5)=-4.*X(1)**2-X(2)**2+3.*X(1)*X(2)-2.*X(3)**2-5.*X(6)+11.*X(
     17)
      GCONS(1,5)=-8.*X(1)+3.*X(2)
      GCONS(2,5)=-2.*X(2)+3.*X(1)
      GCONS(3,5)=-4.*X(3)
      GCONS(4,5)=0.
      GCONS(5,5)=0.
      GCONS(6,5)=-5.
      GCONS(7,5)=11.
60    CONTINUE
C
      IFN=IFN+1
      RETURN
      END
```

INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISOPT3

```
INITIAL VALUE OF THE ELEMENTS OF AL ... ALMIN =  .10000000E+01
OPTIMAL OBJECTIVE AT NODE 0 (GUESS) ..... EST = 0.
VALUE OF PARAMETER P ..................... IP =    10
(-LARGE,LARGE) BRACKETS ALL VARIABLES . LARGE =  .10000000E+11
ALLOWED FUNCTION CALLS AT EACH NODE .. MAXIFN =  1000
ALLOWED QUASID CALLS AT EACH NODE .... MAXITN =    15
ALLOWED NUMBER OF NODES ............. MAXNODE =  1000
NUMBER OF DISCRETE VARIABLES ........... NDIS =     0
NUMBER OF CONSTRAINTS IN THE PROBLEM  NORCONS =     5
NUMBER OF UNIFORM STEP VARIABLES ....... NUNI =     0
TOLERANCE FOR THE CONSTRAINTS ....... TOLCONS = -.10000000E-01
TOLERANCE FOR THE DISCRETE VARIABLES . TOLDIS =  .10000000E-02
STOPPING CRITERION FOR UOPT ......... TOLHEXI =  .10000000E-02
TOLERANCE FOR THE MULTIPLIERS ....... TOLMULT =  .10000000E-03
STOPPING CRITERION FOR QUASID .......... TOLX =  .10000000E-06
INITIAL VALUE OF THE UPPER BOUND ...... UPBND =  .10000000E+11
STARTING POINT FOR THIS PROBLEM ........... X 1  .10000000E+01
                                              2  .20000000E+01
                                              3 0.
                                              4  .40000000E+01
                                              5 0.
                                              6  .10000000E+01
                                              7  .10000000E+01
```

OPTIONS IN EFFECT

GRADIENT CHECK AT THE STARTING POINT

ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION

VERTICES AROUND NODE 0 SOLUTION EXAMINED

DETAILED PRINTING REQUESTED

GRADIENT CHECK AT THE STARTING POINT

ANALYTICAL GRADIENT VECTOR G(I)		NUMERICAL GRADIENT VECTOR Y(I)		PERCENTAGE ERROR VECTOR PERCENT(I)	
1	-.17452567E+02	1	-.17452567E+02	1	.36068278E-07
2	-.80668954E+02	2	-.80668953E+02	2	.15850950E-05
3	.33003301E+00	3	.32741809E+00	3	.79864706E+00
4	-.36839538E+02	4	-.36839538E+02	4	.22212107E-06
5	.16441766E+01	5	.16552804E+01	5	.67080719E+00
6	.22485577E+01	6	.22485577E+01	6	.13501760E-05
7	-.13366959E+02	7	-.13366959E+02	7	.43201089E-06

THE GRADIENTS APPEAR TO BE CORRECT

FEASIBILITY CHECK AT NODE 0

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION	I	VARIABLE VECTOR X(I)	I	GRADIENT VECTOR G(I)
0	1	-.38217634E+01	1	.10000000E+01	1	.18462889E+01
			2	.20000000E+01	2	.33113609E+01
			3	0.	3	.25407465E-01
			4	.40000000E+01	4	.81304187E+00
			5	0.	5	.12703432E+00
			6	.10000000E+01	6	.43610871E+01
			7	.10000000E+01	7	-.95941862E+01

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.71400000E+03
	2	.10000000E+01	2	.13000000E+02
	3	.10000000E+01	3	.26500000E+03
	4	.10000000E+01	4	.17100000E+03
	5	.10000000E+01	5	.40000000E+01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	.79288763E+03	1	.10000000E+01	1	-.17452567E+02
			2	.20000000E+01	2	-.80668954E+02
			3	0.	3	.33003301E+00
			4	.40000000E+01	4	-.36839538E+02
			5	0.	5	.16441766E+01
			6	.10000000E+01	6	.22485577E+01
			7	.10000000E+01	7	-.13366959E+02
17	28	.71557033E+03	1	.34166291E+01	1	.31430092E-07
			2	.23014021E+01	2	-.31680140E-06
			3	-.44287010E+00	3	.16584093E-07
			4	.58126321E+01	4	-.32506392E-07
			5	-.56138621E+00	5	.41327210E-07
			6	.10726683E+01	6	.30908934E-07
			7	.15317032E+01	7	.18541799E-06

THE ABOVE ITERATION HAS RESULTED IN A NONFEASIBLE SOLUTION. THE CONSTRAINTS AT THIS POINT ARE GIVEN AS FOLLOWS. IT MAY BE NOTED THAT THE STARTING POINT FOR THE NEXT ITERATION IS NOT THE ABOVE SOLUTION BUT THE BEST FEASIBLE POINT OBTAINED SO FAR

CONS	1	.57873885E+03	2	-.11240087E+03	3	.24284403E+03
	4	.11747100E+03	5	-.17307631E+02		

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.71400000E+03
	2	.10000000E+02	2	.13000000E+02
	3	.10000000E+02	3	.26500000E+03
	4	.10000000E+02	4	.17100000E+03
	5	.10000000E+02	5	.40000000E+01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	31	.75272643E+03	1	.10000000E+01	1	-.82089809E+01
			2	.20000000E+01	2	-.78706625E+01
			3	0.	3	.10185471E+01
			4	.40000000E+01	4	-.13334000E+02
			5	0.	5	.50927356E+01
			6	.10000000E+01	6	.18500121E+02
			7	.10000000E+01	7	-.49448364E+02
17	55	.72279067E+03	1	.15736285E+01	1	.34121678E-07
			2	.19207655E+01	2	.48349497E-06
			3	-.21294121E+00	3	-.20517089E-08
			4	.42321180E+01	4	.17061525E-06
			5	-.63089198E+00	5	-.48806078E-07
			6	.76103566E+00	6	-.20186444E-07
			7	.18670823E+01	7	-.16437603E-07

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .70497960E+03

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.70497960E+03
	2	.10000000E+02	2	.12937806E+02
	3	.10000000E+02	3	.25990585E+03
	4	.10000000E+02	4	.16757881E+03
	5	.10000000E+02	5	.12115186E+02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	57	-.98225428E-10	1	.15736285E+01	1	-.16852743E+02
			2	.19207655E+01	2	-.10079234E+03
			3	-.21294121E+00	3	-.38622391E-01
			4	.42321180E+01	4	-.40607292E+02
			5	-.63089198E+00	5	-.59968929E+01
			6	.76103566E+00	6	-.68138300E+01
			7	.18670823E+01	7	.14990426E+02
20	114	-.23052973E+02	1	.22860816E+01	1	.14386276E-05
			2	.19512499E+01	2	.33096387E-04
			3	-.46276332E+00	3	.71379587E-06
			4	.43639135E+01	4	.12323952E-04
			5	-.62457299E+00	5	.95302214E-06
			6	.10292990E+01	6	-.73532706E-06
			7	.16050320E+01	7	.13762745E-05

ITERATION NUMBER 4 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .68145499E+03

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.68145499E+03
2	.11405053E+01	2	.34215160E+01	2	.46999076E+00
3	.34291981E-21	3	INACTIVE	3	.25301369E+03
4	.13385916E-18	4	INACTIVE	4	.14609626E+03
5	.40198827E+00	5	.12059648E+01	5	.75065347E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	118	-.98225428E-10	1	.22860816E+01	1	-.15427837E+02
			2	.19512499E+01	2	-.10048750E+03
			3	-.46276332E+00	3	-.39640286E+00
			4	.43639135E+01	4	-.39816519E+02
			5	-.62457299E+00	5	-.57025256E+01
			6	.10292990E+01	6	-.20099421E+01
			7	.16050320E+01	7	.44218724E+01
15	144	-.73925579E+00	1	.23299094E+01	1	.36385162E-06
			2	.19513899E+01	2	.34167386E-05
			3	-.47732792E+00	3	.64835468E-07
			4	.43657823E+01	4	.13312559E-05
			5	-.62448431E+00	5	.32113497E-06
			6	.10380101E+01	6	-.27524283E-07
			7	.15943760E+01	7	-.54908419E-07

ITERATION NUMBER 5 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .68063583E+03

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.68063583E+03
2	.11396957E+01	2	.34190871E+01	2	.17490347E-02
3	.34291981E-21	3	INACTIVE	3	.25256778E+03
4	.13385916E-18	4	INACTIVE	4	.14489438E+03
5	.36907245E+00	5	.11072174E+01	5	.10252320E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	148	-.98225428E-10	1	.23299094E+01	1	-.15340181E+02
			2	.19513899E+01	2	-.10048610E+03
			3	-.47732792E+00	3	-.43502129E+00
			4	.43657823E+01	4	-.39805306E+02
			5	-.62448431E+00	5	-.56984782E+01
			6	.10380101E+01	6	-.18453623E+01
			7	.15943760E+01	7	.40597969E+01
15	174	-.51740615E-02	1	.23304993E+01	1	-.83917482E-05
			2	.19513724E+01	2	-.30012795E-04
			3	-.47754136E+00	3	.42997929E-06
			4	.43657262E+01	4	-.12412704E-04
			5	-.62448697E+00	5	-.17353296E-05
			6	.10381310E+01	6	-.19494934E-05
			7	.15942267E+01	7	.44064735E-05

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

THE SOLUTION WITH 5 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.23304993E+01	2	.19513724E+01	3	-.47754136E+00
	4	.43657262E+01	5	-.62448697E+00	6	.10381310E+01
	7	.15942267E+01				
CONS	1	.68063006E+03	2	.18413175E-06	3	.25256172E+03
	4	.14487818E+03	5	.11677357E-05		

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 193

OUT OF THESE 178 WERE PERFORMED AT THIS NODE

<u>Example 7: The Wong problem 2 [12]</u>

Minimize

$$f = x_1^2 + x_2^2 + x_1 x_2 - 14 x_1 - 16 x_2 + (x_3 - 10)^2 + 4(x_4 - 5)^2 + (x_5 - 3)^2 + 2(x_6 - 1)^2 + 5 x_7^2 + 7(x_8 - 11)^2 + 2(x_9 - 10)^2 + (x_{10} - 7)^2 + 45$$

subject to

$$-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2 x_3^2 + 7x_4 + 120 \geq 0$$

$$-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0$$

$$-0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \geq 0$$

$$-x_1^2 - 2(x_2 - 2)^2 + 2x_1 x_2 - 14x_5 + 6x_6 \geq 0$$

$$-4x_1 - 5x_2 + 3x_7 - 9x_8 + 105 \geq 0$$

$$-10x_1 + 8x_2 + 17x_7 - 2x_8 \geq 0$$

$$3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0$$

$$8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0$$

The optimal solution is

$$f = 24.306209$$

$$x_1 = 2.171996 \qquad x_2 = 2.363683$$

$$x_3 = 8.773926 \qquad x_4 = 5.095985$$

$$x_5 = 0.990655 \qquad x_6 = 1.430574$$

$$x_7 = 1.321644 \qquad x_8 = 9.828726$$

$$x_9 = 8.280092 \qquad x_{10} = 8.375927$$

The results obtained by DISOPT3 are consistent with the results presented by Charalambous [2]. A comparison is made in Table XI. A complete listing of the main program, subroutine FUN and the output is also presented.

TABLE XI COMPARISON BETWEEN THE RESULTS OF CHARALAMBOUS/DISOPT3 ON EXAMPLE 7

Iteration number	1	2	3	4	5
Objective function	-18.22450 -18.22498	26.15797 26.15797	24.42958 24.42958	24.30638 24.30638	24.306209 24.306209
Function evaluations	48 50	57 55	49 72	54 52	65 50
Total number of function evaluations				273/279	

```
      PROGRAM TST(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
C     MAIN PROGRAM FOR EXAMPLE 7
CC
      DIMENSION IAR(180), X(360), XD(10)
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
C
      DATA DIS,N,NORCONS/0.,10,9/
      DATA X/2.,3.,5.,5.,1.,2.,7.,3.,6.,10./
      DATA ALMIN,IP/1.,10/,TOLCONS,TOLHEXI,TOLMULT/-1.E-2,.001,1.E-4/
C
      PRINTP=3HALL
      CALL DISOPT3 (DIS,IAR,X,XD)
      STOP
      END
```

```
      SUBROUTINE FUN (CONS,GCONS,IDCONS,IDVAR,X)
C
C     THE SECOND WONG PROBLEM
C
      DIMENSION CONS(9), GCONS(10,9), IDCONS(1), X(10)
C
      COMMON /7/ IFN,IND1,IND2
C
      DO 100 I=1,9
      J=IDCONS(I)
      GO TO (10,20,30,40,50,60,70,80,90,100), J
10    CONS(1)=X(1)**2+X(2)**2+X(1)*X(2)-14.*X(1)-16.*X(2)+(X(3)-10.)**2+
     14.*(X(4)-5.)**2+(X(5)-3.)**2+2.*(X(6)-1.)**2+5.*X(7)**2+7.*(X(8)-1
     21.)**2+2.*(X(9)-10.)**2+(X(10)-7.)**2+45.
      GCONS(1,1)=2.*X(1)+X(2)-14.
      GCONS(2,1)=2.*X(2)+X(1)-16.
      GCONS(3,1)=2.*(X(3)-10.)
      GCONS(4,1)=8.*(X(4)-5.)
      GCONS(5,1)=2.*(X(5)-3.)
      GCONS(6,1)=4.*(X(6)-1.)
      GCONS(7,1)=10.*X(7)
      GCONS(8,1)=14.*(X(8)-11.)
      GCONS(9,1)=4.*(X(9)-10.)
      GCONS(10,1)=2.*(X(10)-7.)
      GO TO 100
20    CONS(2)=-3.*(X(1)-2.)**2-4.*(X(2)-3.)**2-2.*X(3)**2+7.*X(4)+120.
      GCONS(1,2)=-6.*(X(1)-2.)
      GCONS(2,2)=-8.*(X(2)-3.)
      GCONS(3,2)=-4.*X(3)
      GCONS(4,2)=7.
      GCONS(5,2)=0.
      GCONS(6,2)=0.
      GCONS(7,2)=0.
      GCONS(8,2)=0.
      GCONS(9,2)=0.
      GCONS(10,2)=0.
      GO TO 100
30    CONS(3)=-5.*X(1)**2-8.*X(2)-(X(3)-6.)**2+2.*X(4)+40.
      GCONS(1,3)=-10.*X(1)
      GCONS(2,3)=-8.
      GCONS(3,3)=-2.*(X(3)-6.)
      GCONS(4,3)=2.
      GCONS(5,3)=0.
      GCONS(6,3)=0.
      GCONS(7,3)=0.
      GCONS(8,3)=0.
      GCONS(9,3)=0.
      GCONS(10,3)=0.
      GO TO 100
40    CONS(4)=-.5*(X(1)-8.)**2-2.*(X(2)-4.)**2-3.*X(5)**2+X(6)+30.
      GCONS(1,4)=8.-X(1)
      GCONS(2,4)=-4.*(X(2)-4.)
      GCONS(3,4)=0.
      GCONS(4,4)=0.
      GCONS(5,4)=-6.*X(5)
      GCONS(6,4)=1.
      GCONS(7,4)=0.
      GCONS(8,4)=0.
      GCONS(9,4)=0.
      GCONS(10,4)=0.
      GO TO 100
50    CONS(5)=-X(1)**2-2.*(X(2)-2.)**2+2.*X(1)*X(2)-14.*X(5)+6.*X(6)
      GCONS(1,5)=-2.*X(1)+2.*X(2)
      GCONS(2,5)=-4.*(X(2)-2.)+2.*X(1)
      GCONS(3,5)=0.
      GCONS(4,5)=0.
      GCONS(5,5)=-14.
      GCONS(6,5)=6.
      GCONS(7,5)=0.
      GCONS(8,5)=0.
      GCONS(9,5)=0.
      GCONS(10,5)=0.
      GO TO 100
```

```
60    CONS(6)=-4.*X(1)-5.*X(2)+3.*X(7)-9.*X(8)+105.              FUN 740
      GCONS(1,6)=-4.                                             FUN 750
      GCONS(2,6)=-5.                                             FUN 760
      GCONS(3,6)=0.                                              FUN 770
      GCONS(4,6)=0.                                              FUN 780
      GCONS(5,6)=0.                                              FUN 790
      GCONS(6,6)=0.                                              FUN 800
      GCONS(7,6)=3.                                              FUN 810
      GCONS(8,6)=-9.                                             FUN 820
      GCONS(9,6)=0.                                              FUN 830
      GCONS(10,6)=0.                                             FUN 840
      GO TO 100                                                  FUN 850
70    CONS(7)=-10.*X(1)+8.*X(2)+17.*X(7)-2.*X(8)                 FUN 860
      GCONS(1,7)=-10.                                            FUN 870
      GCONS(2,7)=8.                                              FUN 880
      GCONS(3,7)=0.                                              FUN 890
      GCONS(4,7)=0.                                              FUN 900
      GCONS(5,7)=0.                                              FUN 910
      GCONS(6,7)=0.                                              FUN 920
      GCONS(7,7)=17.                                             FUN 930
      GCONS(8,7)=-2.                                             FUN 940
      GCONS(9,7)=0.                                              FUN 950
      GCONS(10,7)=0.                                             FUN 960
      GO TO 100                                                  FUN 970
80    CONS(8)=3.*X(1)-6.*X(2)-12.*(X(9)-8.)**2+7.*X(10)          FUN 980
      GCONS(1,8)=3.                                              FUN 990
      GCONS(2,8)=-6.                                             FUN1000
      GCONS(3,8)=0.                                              FUN1010
      GCONS(4,8)=0.                                              FUN1020
      GCONS(5,8)=0.                                              FUN1030
      GCONS(6,8)=0.                                              FUN1040
      GCONS(7,8)=0.                                              FUN1050
      GCONS(8,8)=0.                                              FUN1060
      GCONS(9,8)=-24.*(X(9)-8.)                                  FUN1070
      GCONS(10,8)=7.                                             FUN1080
      GO TO 100                                                  FUN1090
90    CONS(9)=8.*X(1)-2.*X(2)-5.*X(9)+2.*X(10)+12.               FUN1100
      GCONS(1,9)=8.                                              FUN1110
      GCONS(2,9)=-2.                                             FUN1120
      GCONS(3,9)=0.                                              FUN1130
      GCONS(4,9)=0.                                              FUN1140
      GCONS(5,9)=0.                                              FUN1150
      GCONS(6,9)=0.                                              FUN1160
      GCONS(7,9)=0.                                              FUN1170
      GCONS(8,9)=0.                                              FUN1180
      GCONS(9,9)=-5.                                             FUN1190
      GCONS(10,9)=2.                                             FUN1200
100   CONTINUE                                                   FUN1210
C                                                                FUN1220
      IFN=IFN+1                                                  FUN1230
      RETURN                                                     FUN1240
      END                                                        FUN1250-
```

INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISOPT3

```
INITIAL VALUE OF THE ELEMENTS OF AL ... ALMIN =  .10000000E+01
OPTIMAL OBJECTIVE AT NODE 0 (GUESS) ..... EST = 0.
VALUE OF PARAMETER P ..................... IP =     10
(-LARGE,LARGE) BRACKETS ALL VARIABLES . LARGE =  .10000000E+11
ALLOWED FUNCTION CALLS AT EACH NODE .. MAXIFN =  1000
ALLOWED QUASID CALLS AT EACH NODE .... MAXITN =    15
ALLOWED NUMBER OF NODES ............ MAXNODE =  1000
NUMBER OF DISCRETE VARIABLES ........... NDIS =     0
NUMBER OF CONSTRAINTS IN THE PROBLEM  NORCONS =     9
NUMBER OF UNIFORM STEP VARIABLES ....... NUNI =     0
TOLERANCE FOR THE CONSTRAINTS ....... TOLCONS = -.10000000E-01
TOLERANCE FOR THE DISCRETE VARIABLES . TOLDIS =  .10000000E-02
STOPPING CRITERION FOR UOPT ......... TOLHEXI =  .10000000E-02
TOLERANCE FOR THE MULTIPLIERS ....... TOLMULT =  .10000000E-03
STOPPING CRITERION FOR QUASID .......... TOLX =  .10000000E-06
INITIAL VALUE OF THE UPPER BOUND ...... UPBND =  .10000000E+11
STARTING POINT FOR THIS PROBLEM ........... X 1  .20000000E+01
                                              2  .30000000E+01
                                              3  .50000000E+01
                                              4  .50000000E+01
                                              5  .10000000E+01
                                              6  .20000000E+01
                                              7  .70000000E+01
                                              8  .30000000E+01
                                              9  .60000000E+01
                                             10  .10000000E+02
```

OPTIONS IN EFFECT

GRADIENT CHECK AT THE STARTING POINT

ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION

VERTICES AROUND NODE 0 SOLUTION EXAMINED

DETAILED PRINTING REQUESTED

GRADIENT CHECK AT THE STARTING POINT

	ANALYTICAL GRADIENT VECTOR G(I)		NUMERICAL GRADIENT VECTOR Y(I)		PERCENTAGE ERROR VECTOR PERCENT(I)
1	-.75256281E+01	1	-.75256281E+01	1	.28875029E-06
2	-.77036885E+01	2	-.77036885E+01	2	.96973452E-07
3	-.11678879E+02	3	-.11678879E+02	3	.50200769E-07
4	-.71024624E+00	4	-.71024624E+00	4	.48847533E-06
5	-.13250269E+01	5	-.13250269E+01	5	.50134075E-06
6	.36530360E+01	6	.36530360E+01	6	.43300044E-06
7	.85246476E+02	7	.85246476E+02	7	.18825451E-08
8	-.13714403E+03	8	-.13714403E+03	8	.62205117E-09
9	-.27046790E+02	9	-.27046790E+02	9	.16254318E-05
10	.58615790E+01	10	.58615790E+01	10	.15268917E-08

THE GRADIENTS APPEAR TO BE CORRECT

FEASIBILITY CHECK AT NODE 0

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	1	-.27510862E+01	1	.20000000E+01	1	.23508603E+01
			2	.30000000E+01	2	.13674920E+01
			3	.50000000E+01	3	-.33278472E+00
			4	.50000000E+01	4	-.33327035E+00
			5	.10000000E+01	5	.47260799E+01
			6	.20000000E+01	6	-.19805800E+01
			7	.70000000E+01	7	-.36330206E-03
			8	.30000000E+01	8	.45288908E-03
			9	.60000000E+01	9	-.93918587E+00
			10	.10000000E+02	10	-.16984969E+00

ITERATION NUMBER 1 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.75300000E+03
	2	.10000000E+01	2	.10500000E+03
	3	.10000000E+01	3	.50000000E+01
	4	.10000000E+01	4	.90000000E+01
	5	.10000000E+01	5	.40000000E+01
	6	.10000000E+01	6	.76000000E+02
	7	.10000000E+01	7	.11700000E+03
	8	.10000000E+01	8	.10000000E+02
	9	.10000000E+01	9	.12000000E+02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	3	.90443858E+03	1	.20000000E+01	1	-.75256281E+01
			2	.30000000E+01	2	-.77036885E+01
			3	.50000000E+01	3	-.11678879E+02
			4	.50000000E+01	4	-.71024624E+00
			5	.10000000E+01	5	-.13250269E+01
			6	.20000000E+01	6	.36530360E+01
			7	.70000000E+01	7	.85246476E+02
			8	.30000000E+01	8	-.13714403E+03
			9	.60000000E+01	9	-.27046790E+02
			10	.10000000E+02	10	.58615790E+01
31	52	.83087861E+00	1	.22568303E+01	1	.81008085E-06
			2	.41306565E+01	2	.38716761E-07
			3	.91750958E+01	3	-.27418885E-05
			4	.50515773E+01	4	.67836075E-06
			5	.21595763E+01	5	-.21778055E-05
			6	.11800908E+01	6	.10224182E-05
			7	.21969076E+00	7	-.12012407E-05
			8	.10529234E+02	8	.20552927E-05
			9	.10000000E+02	9	.17542098E-07
			10	.70000000E+01	10	-.11513731E-07

THE ABOVE ITERATION HAS RESULTED IN A NONFEASIBLE SOLUTION. THE CONSTRAINTS AT THIS POINT ARE GIVEN AS FOLLOWS. IT MAY BE NOTED THAT THE STARTING POINT FOR THE NEXT ITERATION IS NOT THE ABOVE SOLUTION BUT THE BEST FEASIBLE POINT OBTAINED SO FAR

CONS	1	-.18224975E+02	2	-.18777411E+02	3	-.18889747E+02
	4	.63150933E+00	5	-.18887267E+02	6	-.19034638E+02
	7	-.64467760E+01	8	-.17313448E+02	9	-.14306670E+02

ITERATION NUMBER 2 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = 0.

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.75300000E+03
	2	.10000000E+02	2	.10500000E+03
	3	.10000000E+02	3	.50000000E+01
	4	.10000000E+02	4	.90000000E+01
	5	.10000000E+02	5	.40000000E+01
	6	.10000000E+02	6	.76000000E+02
	7	.10000000E+02	7	.11700000E+03
	8	.10000000E+02	8	.10000000E+02
	9	.10000000E+02	9	.12000000E+02

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	55	.83403970E+03	1	.20000000E+01	1	.12220232E+02
			2	.30000000E+01	2	.10989327E+02
			3	.50000000E+01	3	-.16074643E+02
			4	.50000000E+01	4	-.42948443E+01
			5	.10000000E+01	5	.37032160E+02
			6	.20000000E+01	6	-.11186646E+02
			7	.70000000E+01	7	.82458588E+02
			8	.30000000E+01	8	-.13193374E+03
			9	.60000000E+01	9	-.67731597E+02
			10	.10000000E+02	10	-.23412785E+01
34	109	.27460677E+02	1	.21560750E+01	1	-.41073479E-08
			2	.23437964E+01	2	.67819377E-08
			3	.87376438E+01	3	.46197753E-09
			4	.51001570E+01	4	.86245086E-09
			5	.93844466E+00	5	-.14931227E-08
			6	.14417619E+01	6	-.60392648E-09
			7	.13575276E+01	7	-.23646503E-08
			8	.98225645E+01	8	.23678161E-07
			9	.82103568E+01	9	-.26784530E-08
			10	.84317146E+01	10	.23228993E-08

ITERATION NUMBER 3 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .26157974E+02

MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
NOT CALCULATED	1	OBJECTIVE	1	.26157974E+02
	2	.10000000E+02	2	.12127709E+01
	3	.10000000E+02	3	.71195157E+00
	4	.10000000E+02	4	.62379765E+01
	5	.10000000E+02	5	.73409619E+00
	6	.10000000E+02	6	.32622001E+00
	7	.10000000E+02	7	.62246040E+00
	8	.10000000E+02	8	.50896449E+02
	9	.10000000E+02	9	.37265278E+00

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	111	-.99930730E-10	1	.21560750E+01	1	-.73440535E+01
			2	.23437964E+01	2	-.91563322E+01
			3	.87376438E+01	3	-.25247125E+01
			4	.51001570E+01	4	.80125634E+00
			5	.93844466E+00	5	-.41231107E+01
			6	.14417619E+01	6	.17670474E+01
			7	.13575276E+01	7	.13575276E+02
			8	.98225645E+01	8	-.16484097E+02
			9	.82103568E+01	9	-.71585729E+01
			10	.84317146E+01	10	.28634291E+01
22	182	-.16256787E+01	1	.21708621E+01	1	-.50730796E-05
			2	.23626031E+01	2	.77705643E-06
			3	.87705716E+01	3	.17577970E-05
			4	.50963275E+01	4	-.34707885E-06
			5	.98686990E+00	5	.14586852E-05
			6	.14313851E+01	6	-.28553553E-06
			7	.13240404E+01	7	.43704627E-07
			8	.98283790E+01	8	-.13665007E-05
			9	.82753891E+01	9	.38012546E-05
			10	.83796887E+01	10	-.14781230E-05

ITERATION NUMBER 4 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .24429584E+02

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.24429584E+02
2	.20639586E-01	2	.14447710E+00	2	.11576112E+00
3	.31307169E+00	3	.21915018E+01	3	.52552777E-01
4	.36460515E-16	4	INACTIVE	4	.61580870E+01
5	.28759013E+00	5	.20131309E+01	5	.54299038E-01
6	.17167685E+01	6	.12017380E+02	6	.20246684E-01
7	.47588815E+00	7	.33312170E+01	7	.44133774E-01
8	.46211204E-26	8	INACTIVE	8	.50084718E+02
9	.13796896E+01	9	.96578269E+01	9	.24121940E-01

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	186	-.99930730E-10	1	.21708621E+01	1	-.72956727E+01
			2	.23626031E+01	2	-.91039317E+01
			3	.87705716E+01	3	-.24588569E+01
			4	.50963275E+01	4	.77062008E+00
			5	.98686990E+00	5	-.40262602E+01
			6	.14313851E+01	6	.17255405E+01
			7	.13240404E+01	7	.13240404E+02
			8	.98283790E+01	8	-.16402695E+02
			9	.82753891E+01	9	-.68984435E+01
			10	.83796887E+01	10	.27593774E+01
26	237	-.10155877E+00	1	.21720012E+01	1	.36540935E-05
			2	.23636878E+01	2	.24744698E-05
			3	.87738835E+01	3	.23532304E-06
			4	.50959869E+01	4	-.12116846E-06
			5	.99064958E+00	5	-.35062561E-06
			6	.14305751E+01	6	.16126493E-06
			7	.13216477E+01	7	.18913223E-05
			8	.98287205E+01	8	.22694965E-05
			9	.82800929E+01	9	-.18209734E-05
			10	.83759257E+01	10	.77182968E-06

ITERATION NUMBER 5 OF THE CHARALAMBOUS METHOD (LEAST PTH APPROACH)

FOR THE NONLINEAR PROGRAMMING PROBLEM AT NODE 0

VALUE OF HEXI FOR THIS ITERATION .... HEXI = .24306379E+02

	MULTIPLIER VECTOR RMULT(I)		ALPHA VECTOR AL(I)		CONSTRAINT VECTOR CONS(I)
1	ACTIVE	1	OBJECTIVE	1	.24306379E+02
2	.20549399E-01	2	.14384579E+00	2	.15199347E-02
3	.31202486E+00	3	.21841740E+01	3	.94930773E-04
4	.36460515E-16	4	INACTIVE	4	.61485955E+01
5	.28705003E+00	5	.20093502E+01	5	.95152708E-04
6	.17165416E+01	6	.12015791E+02	6	.14308628E-04
7	.47452060E+00	7	.33216442E+01	7	.60863890E-04
8	.46211204E-26	8	INACTIVE	8	.50023932E+02
9	.13759252E+01	9	.96314765E+01	9	.20824199E-04

UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLETCHERS METHOD

ITER. NO.	FUNC. EVAL.	LEAST PTH FUNCTION		VARIABLE VECTOR X(I)		GRADIENT VECTOR G(I)
0	241	-.99930730E-10	1	.21720012E+01	1	-.72923097E+01
			2	.23636878E+01	2	-.91006231E+01
			3	.87738835E+01	3	-.24522330E+01
			4	.50959869E+01	4	.76789536E+00
			5	.99064958E+00	5	-.40187008E+01
			6	.14305751E+01	6	.17223004E+01
			7	.13216477E+01	7	.13216477E+02
			8	.98287205E+01	8	-.16397913E+02
			9	.82800929E+01	9	-.68796283E+01
			10	.83759257E+01	10	.27518514E+01
24	290	-.14015349E-03	1	.21719964E+01	1	.48677419E-03
			2	.23636830E+01	2	.70040610E-03
			3	.87739257E+01	3	.30838701E-04
			4	.50959845E+01	4	-.34034597E-05
			5	.99065474E+00	5	-.17216402E-03
			6	.14305739E+01	6	.73625402E-04
			7	.13216442E+01	7	-.20417606E-03
			8	.98287258E+01	8	.11421048E-02
			9	.82800916E+01	9	-.88128702E-04
			10	.83759266E+01	10	.35032717E-04

OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE 0

THE SOLUTION WITH 9 CONSTRAINTS (CONS(1)=OBJECTIVE) IS

X	1	.21719964E+01	2	.23636830E+01	3	.87739257E+01
	4	.50959845E+01	5	.99065474E+00	6	.14305739E+01
	7	.13216442E+01	8	.98287258E+01	9	.82800916E+01
	10	.83759266E+01				
CONS	1	.24306209E+02	2	.23905159E-07	3	.83400664E-09
	4	.61485034E+01	5	.13207000E-08	6	.39108272E-10
	7	.67814199E-09	8	.50023961E+02	9	.21123014E-09

THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED SO FAR IS 315

OUT OF THESE 294 WERE PERFORMED AT THIS NODE

Appendix 1

Listing of subroutines BOUND, DISOPT3, FIND, GRDCHK3, LEASTPD, OBJ, QUASID AND UOPT

```
      SUBROUTINE BOUND (CONS,DIS,GCONS,IAR,IDCONS,IDDIS,IDVAR,VL,VU,X,XD
     1,Y)
C
C     THIS SUBROUTINE DETERMINES THE UPPER BOUND BY FINDING THE BEST FEA
C     SIBLE DISCRETE POINT IN THE VICINITY OF THE GIVEN POINT X. TO DO
C     SO IT (1) FINDS THE NEAREST LOWER AND NEAREST UPPER DISCRETE VALUE
C     FOR EACH NONDISCRETE ELEMENT OF X AND THESE ARE STORED IN ARRAYS
C     VL AND VU (2) FOR K NON-DISCRETE ELEMENTS 2**K DISCRETE COMBINA-
C     TIONS ARE POSSIBLE. EACH COMBINATION REPRESENTS A DISCRETE POINT.
C     THE OBJECTIVE FUNCTION IS EVALUATED AT EACH POINT AND, IF LESS
C     THAN THE CURRENT UPPER BOUND, THE FEASIBILITY OF THIS POINT IS
C     ALSO CHECKED (3) THE FEASIBLE DISCRETE POINT YIELDING THE LOWEST
C     OBJECTIVE FUNCTION VALUE IS STORED IN ARRAY XD AND THE OBJECTIVE
C     FUNCTION VALUE IS THE DESIRED UPPER BOUND (4) IF AN UPPER BOUND
C     IS FOUND UPDATED, A LOGICAL VARIABLE, IS TRUE
C
C     INPUT    DISCRET,IAR,IDCONS,II,NNCON,NORCONS,REVERSE,TOLCONS,UPBND,
C              X
C
C     OUTPUT   UPBND,UPDATED,XD
C
      DIMENSION CONS(1), DIS(1), GCONS(1), IAR(1), IDCONS(1), IDDIS(1),
     1IDVAR(1), VL(1), VU(1), X(1), XD(1), Y(1)
C
      REAL LARGE
      LOGICAL DISCRET,REVERSE,UPDATED
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /7/ IFN,IND1,IND2
      COMMON /9/ IEXIT,SKIPOBJ,UOBJ,UPDATED,WRONG
      COMMON /10/ GRADCHK,HOLDVAR,ONESOL,REVERSE,VERTCHK
C
C     INITIALIZE. IF THE SOLUTION IS ALREADY DISCRETE RETURN
C
      IA=II
      XLD=XL
      XUD=XU
      IC=0
      UPDATED=.FALSE.
      IF (DISCRET) GO TO 170
C
      DO 10 I=1,NNCON
      Y(I)=X(I)
10    CONTINUE
C
      IF (.NOT.REVERSE) GO TO 20
      IC=1
      II=0
      K=0
      REVERSE=.FALSE.
      GO TO 50
C
C     TAKE ADVANTAGE OF THE FACT THAT FIND HAS ALREADY BEEN CALLED ONCE
C
20    CONTINUE
      IF (XL.LE.-LARGE) GO TO 30
      IF (XU.GE.+LARGE) GO TO 40
      IDDIS(1)=II
      K=1
      X(II)=XL
      VL(II)=XL
      VU(II)=XU
      GO TO 50
30    K=0
      X(II)=XU
      GO TO 50
40    K=0
      X(II)=XL
C
```

```
C     GENERATE ARRAYS VL AND VU
C
50    IF (II.EQ.NNCON) GO TO 80
      IFIND=II+1
      CALL FIND (DIS,IAR,X)
      IF (DISCRET) GO TO 80
      IF (XL.LE.-LARGE) GO TO 60
      IF (XU.GE.+LARGE) GO TO 70
      K=K+1
      IDDIS(K)=II
      X(II)=XL
      VL(II)=XL
      VU(II)=XU
      GO TO 50
60    X(II)=XU
      GO TO 50
70    X(II)=XL
      GO TO 50
C
C     USE A NUMBERING SCHEME TO IDENTIFY A DISCRETE POINT. EVALUATE THE
C     OBJECTIVE FUNCTION AT THIS POINT. CHECK FEASIBILITY IF NECESSARY
C
80    K2=2**K
C
      DO 150 I=1,K2
      II=1
      IF (K.EQ.0) GO TO 110
C
      DO 100 J=1,K
      M=(K+1)-J
      MP=2**(M-1)
      IB=(I-II)/MP
      II=II+IB*MP
      IF (IB.EQ.0) GO TO 90
      X(IDDIS(M))=VU(IDDIS(M))
      GO TO 100
90    X(IDDIS(M))=VL(IDDIS(M))
100   CONTINUE
C
      IND1=0
110   CALL FUN (CONS,GCONS,IDCONS,IDVAR,X)
      IND1=1
      IF (CONS(1).GE.UPBND) GO TO 150
      IF (NORCONS.EQ.1) GO TO 130
      IND2=0
      CALL FUN (CONS,GCONS,IDCONS,IDVAR,X)
      IND2=1
C
      DO 120 M=2,NORCONS
      IF ((CONS(M)-TOLCONS).LT.0.) GO TO 150
120   CONTINUE
C
130   UPBND=CONS(1)
      UPDATED=.TRUE.
C
      DO 140 M=1,N
      XD(M)=X(M)
140   CONTINUE
C
150   CONTINUE
C
      DO 160 I=1,NNCON
      X(I)=Y(I)
160   CONTINUE
C
      DISCRET=.FALSE.
170   CONTINUE
      II=IA
      XL=XLD
      XU=XUD
      IF (IC.EQ.1) REVERSE=.TRUE.
      RETURN
      END
```

```
      SUBROUTINE DISOPT3 (DIS,IAR,X,XD)
C
C     THIS SUBROUTINE FINDS THE DISCRETE SOLUTIONS BY EMPLOYING THE
C     BRANCH AND BOUND ALGORITHM. REFER TO REPORT SOC-XXX FOR DETAILS
C
C     INPUT   DIS,N,NORCONS,X
C
C     OUTPUT  XD
C
C         DESCRIPTION OF THE IMPORTANT VARIABLES IN THIS PROGRAM
C         ------------------------------------------------------
C
C     ********************** HOLLERITH VARIABLES ***********************
C
C     PRINTP  OFFERS THESE OPTIONS FOR PRINTING THE RESULTS
C             NONE    NO PRINTING AT ALL. OTHER PARAMETERS INEFFECTIVE
C             ONLYDIS ONLY DISCRETE SOLUTIONS WILL BE PRINTED
C             NODEOPT OPTIMAL SOLUTION AT EACH NODE WILL BE PRINTED
C             ALL      IN ADDITION TO THE OPTIMAL SOLUTION AT EACH NODE
C                     THE DETAILS OF EACH ITERATION IN SUBROUTINE UOPT
C                     WILL ALSO BE PRINTED
C
C     PRINTID YES IF THE INPUT DATA IS TO BE PRINTED. OTHERWISE NO
C
C     *********************** INTEGER VARIABLES ***********************
C
C     IAR     AN ARRAY OF 6*IEXTRA+4*N+2*NORCONS ELEMENTS USED AS WORK-
C             ING SPACE
C
C     IDCONS  AN ARRAY IDENTIFYING THE ACTIVE CONSTRAINTS. ACTIVE CONS-
C             TRAINTS ARE THOSE CONSTRAINTS WHICH ARE ACTUALLY USED IN
C             THE OPTIMIZATION. OTHERS ARE IGNORED
C
C     IDDIS   USED BY SUBROUTINE BOUND TO STORE THE INDICES OF THOSE
C             DISCRETE VARIABLES WHICH ARE NOT DISCRETE IN THE SOLUTION
C
C     IDVAR   AN ARRAY IDENTIFYING THOSE VARIABLES WHICH ARE ALLOWED TO
C             VARY IN THE OPTIMIZATION
C
C     IEXIT   RETURNED BY SUBROUTINE QUASID. IEXIT=1 INDICATES NORMAL
C             EXECUTION. 2 IMPLIES THAT THE PROGRAM IS UNABLE TO FIND A
C             DOWNHILL DIRECTION AND, THEREFORE, NO OPTIMIZATION IS POSS
C             IBLE. REASONS COULD BE - EPS IS TOO SMALL, GRADIENTS ARE
C             INCORRECT, DIMENSIONS ARE WRONG, OR ANY OTHER PROGRAMMING
C             ERROR. 3 IMPLIES AN INTERRUPTION BECAUSE MAXIFN HAS BEEN
C             EXCEEDED
C
C     IEXTRA  IT IS USED TO ESTIMATE THE REQUIREMENT OF WORKING SPACE
C             FOR THE PROGRAM. ITS DEFAULT VALUE IS 2*N, WHICH IS SUFFI-
C             CIENT FOR MOST OF THE PROBLEMS. IN THE RARE CASE WHEN
C             IEXTRA IS NOT LARGE ENOUGH THE PROGRAM STOPS WITH A MESS-
C             AGE. IEXTRA MAY BE INITIALIZED IN THE MAIN PROGRAM
C
C     IFIND   INDEX OF THE FIRST VARIABLE EXAMINED BY SUBROUTINE FIND
C             FOR A DISCRETE VALUE
C
C     IFN     COUNTS THE FUNCTION EVALUATIONS
C
C     II      RETURNED BY SUBROUTINE FIND ALONG WITH XL AND XU. IT IS
C             THE INDEX OF THAT SOLUTION VARIABLE WHOSE NEAREST LOWER
C             DISCRETE VALUE IS XL AND THE NEAREST UPPER DISCRETE VALUE
C             IS XU
C
C     IND1    EQUALS 0 WHEN THE OBJECTIVE FUNCTION VALUE ALONE IS RE-
C             QUIRED. OTHERWISE, ALL THE CONSTRAINTS MUST BE EVALUATED
C             BY SUBROUTINE FUN
C
C     IND2    THE GRADIENTS NEED NOT BE EVALUATED BY SUBROUTINE FUN
C             WHEN IND2=0
C
C     INT     THAT PART OF ARRAY IAR WHICH HAS FOUR ELEMENTS FOR EACH
C             NODE IS REFERRED TO AS INT BY SOME SUBROUTINES
C
```

```
C     IP      THE PARAMETER P OF THE LEAST PTH OPTIMIZATION
C
C     IPT     THE RESULTS OF THE OPTIMIZATION ARE PRINTED AFTER EVERY
C             IPT ITERATIONS IF IT IS POSITIVE
C
C     JDCONS  IDENTIFIES ALL THE CONSTRAINTS IN A SEQUENTIAL ORDER. ONCE
C             INITIALIZED IT IS NEVER CHANGED. USED INSTEAD OF IDCONS
C             WHENEVER NECESSARY
C
C     JPT     INITIALIZED BY SUBROUTINE DISOPT3. IT CONTROLS PRINTING
C
C     LASTDIS A POINTER TO THE FIRST UNOCCUPIED ELEMENT OF ARRAY DIS
C
C     LASTIAR A POINTER TO THE FIRST UNOCCUPIED ELEMENT OF ARRAY IAR
C
C     LCONS, LER, LGCONS, LGRADU, LH, LRMULT, LW, LX, LY
C             LCONS POINTS TO THAT ELEMENT OF ARRAY X WHICH STORES THE
C             FIRST ELEMENT OF ARRAY CONS. LER POINTS TO THAT ELEMENT OF
C             ARRAY X WHICH STORES THE FIRST ELEMENT OF ARRAY ER. OTHERS
C             MAY BE INTERPRETED SIMILARLY
C
C     LIDCONS, LIDDIS, LIDVAR, LINT, LJDCONS
C             LIDCONS POINTS TO THAT ELEMENT OF ARRAY IAR WHICH STORES
C             THE FIRST ELEMENT OF ARRAY IDCONS. LIDDIS POINTS TO THAT
C             ELEMENT OF ARRAY IAR WHICH STORES THE FIRST ELEMENT OF
C             ARRAY IDDIS. OTHERS MAY BE INTERPRETED SIMILARLY
C
C     MAXIFN  THE MAXIMUM NUMBER OF FUNCTION EVALUATIONS PERMITTED AT
C             EACH NODE
C
C     MAXITN  THE MAXIMUM NUMBER OF ITERATIONS PERMITTED WITHIN SUB-
C             ROUTINE UOPT AT EACH NODE. EACH ITERATION WITHIN UOPT IN-
C             VOLVES A CALL TO QUASID WHICH PERFORMS THE UNCONSTRAINED
C             OPTIMIZATION
C
C     MAXNODE THE MAXIMUM PERMISSIBLE NUMBER OF NODES THAT MAY BE SEAR-
C             CHED FOR A DISCRETE SOLUTION
C
C     MODE    EQUALS 1 IF THE HESSIAN IN SUBROUTINE QUASID, INITIALLY,
C             IS REQUIRED TO BE AN IDENTITY MATRIX. OTHERWISE, THE HESS-
C             IAN GENERATED BY THE LAST CALL TO QUASID, WHICH IS ALREADY
C             IN LDL(TRANSPOSE) FORM, IS USED
C
C     N       THE NUMBER OF VARIABLES IN THE PROBLEM. ALWAYS GREATER
C             THAN 1
C
C     NA      THE NUMBER OF ACTIVE CONSTRAINTS
C
C     NCONS   THE TOTAL NUMBER OF CONSTRAINTS IN THE PROBLEM AT ANY TIME
C
C     NNCON   THE NUMBER OF NON-CONTINUOUS VARIABLES IN THE PROBLEM
C
C     NODE    THE NUMBER OF ADDITIONAL CONSTRAINTS IN THE PROBLEM AT ANY
C             TIME. EACH OF THESE CONSTRAINTS CORRESPONDS TO A NODE IN
C             THE BRANCH AND BOUND ALGORITHM. THOSE NODES WHICH HAVE
C             BEEN FATHOMED ARE NOT INCLUDED IN THIS NUMBER
C
C     NODES   AS OPPOSED TO NODE, NODES EQUALS THE CUMULATIVE NUMBER OF
C             NODES THAT HAVE BEEN ADDED SO FAR
C
C     NORCONS THE NUMBER OF CONSTRAINTS IN THE ORIGINAL PROBLEM. THE OB-
C             JECTIVE FUNCTION IS CALLED THE FIRST CONSTRAINT AND MUST
C             BE COUNTED WITH THEM
C
C     ************************ LOGICAL VARIABLES ***********************
C
C     DISCRET RETURNED BY SUBROUTINE FIND. IT IS TRUE IF THE OPTIMIZA-
C             TION AT ANY NODE RESULTS IN A DISCRETE SOLUTION
C
C     FEASBLE TRUE IF THE OPTIMIZATION AT ANY NODE RESULTS IN A FEASIBLE
C             SOLUTION
C
C     FEASCHK TRUE IF SUBROUTINE UOPT IS PERFORMING A FEASIBILITY CHECK
```

```
C          TO ENSURE THE EXISTENCE OF A FEASIBLE SOLUTION
C
C  GRADCHK IF IT IS TRUE SUBROUTINE GRDCHK3 IS CALLED BY DISOPT3 TO
C          VERIFY THE GRADIENTS AS DEFINED BY THE USER IN SUBROUTINE
C          FUN
C
C  HOLDVAR IF IT IS TRUE THEN ONE SOLUTION VARIABLE IS ALWAYS HELD
C          CONSTANT IN THE OPTIMIZATION AT ALL THE NODES EXCEPTING 0
C
C  MULTS   TRUE IF THE MULTIPLIERS FOR THE ERROR FUNCTIONS ARE TO BE
C          CALCULATED BY SUBROUTINE LEASTPD. ARRAY GRADU IS NOT CAL-
C          CALCULATED IN THIS CASE
C
C  ONESOL  IF IT IS TRUE ONLY ONE OPTIMAL DISCRETE SOLUTION IS FOUND
C
C  REVERSE IF IT IS TRUE THEN THE ORDER IN WHICH THE VARIABLES ARE
C          EXAMINED BY SUBROUTINE FIND FOR A DISCRETE VALUE IS
C          REVERSE
C
C  SKIPOBJ IF IT IS TRUE ONLY THE CONSTRAINTS ARE EVALUATED BY SUB-
C          ROUTINE OBJ AND NOTHING ELSE
C
C  UONLY   WHEN IT IS TRUE ONLY UOBJ IS CALCULATED BY SUBROUTINE
C          LEASTPD AND NOTHING ELSE
C
C  UPDATED RETURNED BY SUBROUTINE BOUND. TRUE IMPLIES THAT THE UPPER
C          BOUND HAS INDEED BEEN UPDATED
C
C  VERTCHK IF IT IS TRUE THEN THE DISCRETE POINTS SURROUNDING THE
C          SOLUTION AT NODE 0 ARE EXAMINED TO YIELD AN UPPER BOUND
C
C  WRONG   RETURNED BY SUBROUTINE GRDCHK3. IF IT IS TRUE, THE GRAD-
C          IENTS AS DEFINED BY THE USER IN SUBROUTINE FUN ARE WRONG
C
C  ************************ REAL VARIABLES **************************
C
C  AL      AN ARRAY OF NORCONS+NODE ELEMENTS USED BY SUBROUTINE OBJ
C          TO CONVERT THE CONSTRAINED PROBLEM INTO A MINIMAX PROBLEM
C
C  ALMIN   EACH ELEMENT OF VECTOR AL INITIALLY EQUALS ALMIN
C
C  CONS    AN ARRAY OF NORCONS+NODE ELEMENTS STORING THE CONSTRAINTS
C          EVALUATED BY SUBROUTINE OBJ. THE FIRST CONSTRAINT STORES
C          THE OBJECTIVE FUNCTION
C
C  DIS     AN ARRAY OF M+IEXTRA*(N+2) ELEMENTS. M IS THE NUMBER OF
C          ELEMENTS USED FOR STORING THE AVAILABLE VALUES FOR THE DIS
C          CRETE VARIABLES. REST OF THE ARRAY IS USED AS WORKING SPA-
C          CE. THE FIRST M ELEMENTS OF DIS ARE INITIALIZED IN THE
C          MAIN PROGRAM ACCORDING TO THE CONVENTION DESCRIBED IN
C          CHAPTER 2 OF REPORT SOC-XXX
C
C  ER      AN ARRAY OF NORCONS+NODE ELEMENTS STORING THE ERROR FUNC-
C          TIONS. EVALUATED BY SUBROUTINE OBJ
C
C  ERMAX   THE MAXIMUM OF THE ERROR FUNCTIONS. EVALUATED BY SUB-
C          ROUTINE LEASTPD
C
C  EST     AN ESTIMATE OF THE OPTIMAL OBJECTIVE FUNCTION VALUE FOR A
C          CONTINUOUS SOLUTION. USED BY SUBROUTINE QUASID
C
C  G OR GRADU
C          AN ARRAY OF N ELEMENTS STORING THE GRADIENT VECTOR OF THE
C          UNCONSTRAINED LEAST PTH OBJECTIVE FUNCTION, UOBJ
C
C  GCONS   AN ARRAY OF N*NORCONS ELEMENTS STORING THE GRADIENT VEC-
C          TORS OF ALL THE ORIGINAL CONSTRAINTS IN THE PROBLEM
C
C  H       THE HESSIAN OF THE UNCONSTRAINED LEAST PTH OBJECTIVE FUNC-
C          TION USED IN SUBROUTINE QUASID. STORED AS LDL(TRANSPOSE)
C
C  HEXI    THE ARTIFICIAL MARGIN IN THE MINIMAX ALGORITHM PROPOSED BY
C          CHARALAMBOUS AND USED IN THIS PROGRAM
```

```
C
C      LARGE   A NUMBER LARGE ENOUGH TO LIE BEYOND THE RANGE OF VALUES
C              THAT THE SOLUTION VARIABLES CAN EVER ASSUME
C
C      RMULT   AN ARRAY OF NORCONS+NODE ELEMENTS STORING THE MULTIPLIERS
C              FOR THE ERROR FUNCTIONS. USED IN SUBROUTINE UOPT TO SELECT
C              ACTIVE FUNCTIONS. EVALUATED BY SUBROUTINE LEASTPD
C
C      TOLCONS A SMALL NEGATIVE NUMBER. IF A CONSTRAINT IS SMALLER THAN 0
C              BUT LARGER THAN TOLCONS IT IS CONSIDERED AS SATISFIED
C
C      TOLDIS  A SMALL POSITIVE NUMBER. IF A NUMBER LIES WITHIN TOLDIS
C              NEIGHBOURHOOD OF A DISCRETE VALUE IT IS ASSUMED TO BE DIS-
C              CRETE
C
C      TOLHEXI A SMALL POSITIVE NUMBER. USED IN SUBROUTINE UOPT AS A
C              STOPPING CRITERION
C
C      TOLMULT A SMALL POSITIVE NUMBER. USED IN SUBROUTINE UOPT TO SELECT
C              ACTIVE FUNCTIONS. IF THE MULTIPLIER OF A FUNCTION EXCEEDS
C              TOLMULT IT IS CONSIDERED ACTIVE
C
C      TOLX    A SMALL POSITIVE NUMBER. USED IN SUBROUTINE QUASID TO TEST
C              THE CONVERGENCE OF THE SOLUTION
C
C      UOBJ    THE UNCONSTRAINED LEAST PTH OBJECTIVE FUNCTION
C
C      UPBND   THE UPPER BOUND ON THE OBJECTIVE FUNCTION. THE INITIAL
C              VALUE IS AN ARBITRARILY LARGE NUMBER. IT IS UPDATED AS
C              SOON AS A DISCRETE SOLUTION IS FOUND
C
C      W       A WORKING ARRAY OF 4*N ELEMENTS USED BY SUBROUTINE QUASID
C
C      X       A WORKING ARRAY OF (N**2+15*N+2*N*NORCONS+10*NORCONS+10*
C              IEXTRA)/2 ELEMENTS. THE FIRST N ELEMENTS STORE THE START-
C              ING POINT AND MUST BE INITIALIZED IN THE MAIN PROGRAM
C
C      XD      AN ARRAY OF N ELEMENTS STORING THE BEST DISCRETE SOLUTION
C
C      XL AND XU
C              SEE II
C
C     *****************************************************************
C
      DIMENSION DIS(1), IAR(1), X(1), XD(1)
C
      REAL LARGE
      LOGICAL DISCRET,FEASBLE,FEASCHK,GRADCHK,HOLDVAR,MULTS,ONESOL,RESTO
     1RE,REVERSE,SKIPOBJ,UONLY,UPDATED,VERTCHK,WRONG
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /3/ IEXTRA,LASTDIS,LASTIAR,NODE,NODES
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /7/ IFN,IND1,IND2
      COMMON /8/ LAL,LCONS,LER,LGCONS,LGRADU,LH,LINT,LRMULT,LW,LX,LY
      COMMON /9/ IEXIT,SKIPOBJ,UOBJ,UPDATED,WRONG
      COMMON /10/ GRADCHK,HOLDVAR,ONESOL,REVERSE,VERTCHK
      COMMON /11/ LIDCONS,LIDDIS,LIDVAR,LJDCONS
C
      DATA PRINTID,PRINTP/3HYES,7HNODEOPT/
      DATA ALMIN,EST,UPBND/10.,0.,1.E10/,IEXTRA,MAXNODE/1,1000/
      DATA IP,IPT,LARGE,MAXIFN,MAXITN/10,500,1.0E+10,1000,15/
      DATA TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX/-.001,2*.001,.1E-7,.1E-6/
      DATA GRADCHK,HOLDVAR,ONESOL,REVERSE,VERTCHK/.T.,.T.,.F.,.F.,.T./
C
      IF (DIS(1).LT.0.) GO TO 250
C
C     INITIALIZE ARRAY IAR WITH THE NECESSARY INFORMATION FOR EACH
C     DISCRETE VARIABLE
C
```

```
      I=0
      IZ=0
      J=-1
      NDIS=0
      NUNI=0
10    CONTINUE
      I=I+IZ+1
      J=J+2
      IZ=IFIX(DIS(I))
      IF (IZ.EQ.0) GO TO 20
      IAR(J)=IZ
      IAR(J+1)=I+1
      IF (IZ.EQ.1) NUNI=NUNI+1
      IF (IZ.GT.1) NDIS=NDIS+1
      GO TO 10
20    CONTINUE
      LASTDIS=I+1
      LASTIAR=J
      NNCON=NDIS+NUNI
C
C     CALCULATE THE POINTERS FOR THE DIFFERENT VARIABLES
C
      IF (IEXTRA.EQ.1) IEXTRA=2*N
      ND=IEXTRA+NORCONS
      LX=1
      LGCONS=LX+N
      LGRADU=LGCONS+N*NORCONS
      LH=LGRADU+N
      LW=LH+N*(N+1)/2
      LY=LW+4*N
      LAL=LY+N
      LCONS=LAL+ND
      LER=LCONS+ND
      LRMULT=LER+2*ND
      LIDVAR=LASTIAR
      LASTIAR=LASTIAR+N
      LINT=LASTIAR
      LIDCONS=LINT+4*IEXTRA
      LJDCONS=LIDCONS+ND
      LIDDIS=LJDCONS+ND
C
C     INITIALIZE ARRAYS AND VARIABLES
C
      IFN=0
      IND1=1
      IND2=1
      MODE=1
      MULTS=.FALSE.
      NODE=0
      NODES=-1
      NX=N
      RESTORE=.FALSE.
      SKIPOBJ=.FALSE.
      UONLY=.FALSE.
      WRONG=.FALSE.
      UPBNDT=UPBND
      IF (ONESOL) UPBNDT=UPBND*(1.-SIGN(1.E-6,UPBND))+TOLCONS
      DO 30 I=1,ND
      IAR((LJDCONS-1)+I)=I
30    CONTINUE
      DO 40 I=1,N
      IAR((LIDVAR-1)+I)=I
40    CONTINUE
C
C     INITIALIZE IPT AND JPT USING PARAMETER PRINTP
C
      JPT=1
      IF (PRINTP.EQ.3HALL) GO TO 50
      IPT=0
      IF (PRINTP.EQ.7HNODEOPT) GO TO 50
      JPT=0
      IF (PRINTP.EQ.7HONLYDIS) GO TO 50
      JPT=-1
```

```
      PRINTID=2HNO
C
C     PRINT THE INPUT DATA
C
50    CONTINUE
      IF (PRINTID.EQ.2HNO) GO TO 100
      PRINT 260, ALMIN,EST,IP,LARGE,MAXIFN,MAXITN,MAXNODE,NDIS,NORCONS,ND
     1UNI,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX,UPBND,(I,X(I),I=1,N)
      I=1
      J=1
60    IF (I.GT.NNCON) GO TO 90
      IF (IAR(J).EQ.1) GO TO 70
      IBEG=IAR(J+1)-1
      IEND=IAR(J)
      PRINT 270, I,(IZ,DIS(IBEG+IZ),IZ=1,IEND)
      GO TO 80
70    PRINT 280, I,DIS(IAR(J+1))
80    I=I+1
      J=J+2
      GO TO 60
90    CONTINUE
      PRINT 290
      IF (GRADCHK) PRINT 300
      IF (HOLDVAR) PRINT 310
      IF (ONESOL) PRINT 320
      IF (REVERSE) PRINT 330
      IF (VERTCHK) PRINT 340
      IF ((IPT.GT.0).AND.(JPT.GT.0)) PRINT 350
      IF ((IPT.EQ.0).AND.(JPT.GT.0)) PRINT 360
      IF ((IPT.EQ.0).AND.(JPT.EQ.0)) PRINT 370
C
C     PERFORM THE GRADIENT CHECK
C
100   CONTINUE
      IF (GRADCHK) CALL GRDCHK3 (X(LAL),X(LCONS),X(LER),X(LGRADU),IAR(LJ
     1DCONS),IAR(LIDVAR),X(LW),X,X(LY))
      IF (WRONG) GO TO 250
C
C     SOLVE THE NONLINEAR PROGRAMMING PROBLEM AT THIS NODE
C
110   CONTINUE
      NODES=NODES+1
      IF (NODES.GT.MAXNODE) GO TO 250
      IFND=IFN
      IFN=0
      Z=UPBND
      UPBND=UPBNDT
      CALL UOPT (X(LAL),X(LCONS),DIS,X(LER),X(LGRADU),X(LH),IAR(LIDCONS)
     1,IAR(LIDVAR),IAR(LINT),IAR(LJDCONS),X(LRMULT),X(LW),X,X(LY))
      UPBND=Z
      IFN=IFN+IFND
      IF (RESTORE) N=N+1
      IF (RESTORE) NODE=NODE+1
C
C     DETERMINE IF THE SOLUTION IS DISCRETE OR NOT
C
      IFIND=1
      CALL FIND (DIS,IAR,X)
      NA=NCONS
      IND2=0
      SKIPOBJ=.TRUE.
      CALL OBJ (1.,X(LCONS),DIS,1.,1.,IAR(LJDCONS),IAR(LIDVAR),IAR(LINT)
     1,1.,X)
      SKIPOBJ=.FALSE.
      IND2=1
C
C     PERFORM THE NECESSARY PRINTING AT THE NODE
C
      IF (JPT.LT.0) GO TO 150
      IF ((JPT.EQ.0).AND..NOT.DISCRET) GO TO 150
      PRINT 390, NODES
      IF (DISCRET) PRINT 400
      IF (.NOT.FEASBLE) PRINT 460
```

```
      IF (NODE.EQ.0) GO TO 140
      IBEG=LRMULT-1
      IEND=LW-1
      J=LASTIAR
      NDIS=LASTIAR
      NUNI=LASTIAR+NODE
      DO 130 I=1,NODE
      J=J-4
      IF (IAR(J).EQ.0) GO TO 120
      IEND=IEND+1
      X(IEND)=DIS(IAR(J+2)+1)
      IAR(NUNI)=IAR(J+1)
      NUNI=NUNI+1
      GO TO 130
120   IBEG=IBEG+1
      X(IBEG)=DIS(IAR(J+2))
      IAR(NDIS)=IAR(J+1)
      NDIS=NDIS+1
130   CONTINUE
      NDIS=LASTIAR-LRMULT
      IF (IBEG.GE.LRMULT) PRINT 410, (IAR(NDIS+I),X(I),I=LRMULT,IBEG)
      NUNI=LASTIAR+NODE-LW
      IF (IEND.GE.LW) PRINT 420, (IAR(NUNI+I),X(I),I=LW,IEND)
140   CONTINUE
      PRINT 430, NCONS,(I,X(I),I=1,N)
      PRINT 440, (I,X((LCONS-1)+I),I=1,NCONS)
      PRINT 450, IFN,IFN-IFND
150   CONTINUE
C
C     IF NO DISCRETE SOLUTION IS REQUIRED RETURN
C
      IF (DIS(1).EQ.0.) GO TO 250
C
C     GENERATE AN UPPER BOUND BY CHECKING THE SURROUNDING VERTICES
C
      F=X(LCONS)
      IF (NODE.NE.0) GO TO 160
      IF (.NOT.VERTCHK) GO TO 160
      CALL BOUND (X(LCONS),DIS,X(LCCONS),IAR,IAR(LJDCONS),IAR(LIDDIS),IA
     1R(LIDVAR),X(LW),X(LW+N),X,XD,X(LY))
      IF (.NOT.UPDATED) GO TO 160
      UPBNDT=UPBND
      IF (ONESOL) UPBNDT=UPBND*(1.-SIGN(1.E-6,UPBND))+TOLCONS
      IF (JPT.LT.0) GO TO 160
      IND2=0
      SKIPOBJ=.TRUE.
      CALL OBJ (1.,X(LCONS),DIS,1.,1.,IAR(LJDCONS),IAR(LIDVAR),IAR(LINT)
     1,1.,XD)
      SKIPOBJ=.FALSE.
      IND2=1
      PRINT 380, (I,XD(I),I=1,N)
      PRINT 440, (I,X((LCONS-1)+I),I=1,NCONS)
160   CONTINUE
C
C     IF THE SOLUTION IS NOT FEASIBLE FATHOM THE NODE
C
      IF (.NOT.FEASBLE) GO TO 180
C
C     IF THE SOLUTION IS DISCRETE UPDATE THE UPPER BOUND AND FATHOM THE
C     NODE ELSE, ADD ANOTHER NODE TO THE TREE
C
      IF (.NOT.DISCRET) GO TO 210
      IF (X(LCONS).GE.UPBND) GO TO 180
      UPBND=X(LCONS)
      UPBNDT=UPBND
      IF (ONESOL) UPBNDT=UPBND*(1.-SIGN(1.E-6,UPBND))+TOLCONS
C
      DO 170 I=1,N
      XD(I)=X(I)
170   CONTINUE
C
C     FATHOM THE NODE
C
```

```
180   CONTINUE
      IF (NODE.LE.0) GO TO 250
      IF (IAR(LASTIAR-4).EQ.0) GO TO 190
      LASTIAR=LASTIAR-4
      LASTDIS=LASTDIS-2-N
      NODE=NODE-1
      GO TO 180
190   IAR(LASTIAR-4)=1
      II=IAR(LASTIAR-3)
      IF (DIS(IAR(LASTIAR-2)+1).GE.+LARGE) GO TO 180
C
      DO 200 I=1,N
      X(I)=DIS(IAR(LASTIAR-1)+I-1)
200   CONTINUE
C
      X(II)=DIS(IAR(LASTIAR-2)+1)
      IF (HOLDVAR) GO TO 230
      GO TO 110
C
C     ADD ANOTHER NODE TO THE TREE
C
210   CONTINUE
      IF ((F+TOLCONS).GT.UPBNDT) GO TO 180
      NODE=NODE+1
      IF (NODE.GT.IEXTRA) GO TO 250
      IAR(LASTIAR)=0
      IAR(LASTIAR+1)=II
      IAR(LASTIAR+2)=LASTDIS
      DIS(LASTDIS)=XL
      LASTDIS=LASTDIS+1
      DIS(LASTDIS)=XU
      LASTDIS=LASTDIS+1
      IAR(LASTIAR+3)=LASTDIS
C
      DO 220 I=1,N
      DIS(LASTDIS)=X(I)
      LASTDIS=LASTDIS+1
220   CONTINUE
C
      LASTIAR=LASTIAR+4
      IF (XL.LE.-LARGE) GO TO 180
      X(II)=XL
      IF (HOLDVAR) GO TO 230
      GO TO 110
C
C     HOLD A VARIABLE CONSTANT BY INITIALIZING IDVAR AND N
C
230   CONTINUE
      IF (N.LE.2) GO TO 110
      J=1
C
      DO 240 I=1,N
      IF (I.EQ.II) GO TO 240
      IAR((LIDVAR-1)+J)=I
      J=J+1
240   CONTINUE
C
      IAR(LIDVAR-1+J)=N+1
      N=N-1
      NODE=NODE-1
      RESTORE=.TRUE.
      GO TO 110
C
250   CONTINUE
      IF ((NODE.GT.IEXTRA).AND.(JPT.GE.0)) PRINT 470
      RETURN
C
260   FORMAT (54H1INPUT DATA FOR THE DISCRETE OPTIMIZATION PROGRAM DISO,
     103HPT3/1X,56(1H-)////33H INITIAL VALUE OF THE ELEMENTS OF,15H AL .
     2.. ALMIN =,E15.8//28H OPTIMAL OBJECTIVE AT NODE 0,20H (GUESS) ....
     3. EST =,E15.8//22H VALUE OF PARAMETER P ,21(1H.),5H IP =,I6//35H (
     4-LARGE,LARGE) BRACKETS ALL VARIAB,13HLES . LARGE =,E15.8//30H ALLO
     5WED FUNCTION CALLS AT EAC,18HH NODE .. MAXIFN =,I6//28H ALLOWED QU
```

```
     6ASID CALLS AT EAC,20HH NODE .... MAXITN =,I6//25H ALLOWED NUMBER O
     7F NODES ,13(1H.),10H MAXNODE =,I6//29H NUMBER OF DISCRETE VARIABLED
     8S,1X,11(1H.),7H NDIS =,I6//29H NUMBER OF CONSTRAINTS IN THE,19H PR
     9OBLEM  NORCONS =,I6//27H NUMBER OF UNIFORM STEP VAR,21HIABLES ....
     $... NUNI =,I6//25H TOLERANCE FOR THE CONSTR,23HAINTS ....... TOLCO
     $NS =,E15.8//20H TOLERANCE FOR THE D,28HISCRETE VARIABLES . TOLDIS
     $=,E15.8//15H STOPPING CRITE,14HRION FOR UOPT ,9(1H.),10H TOLHEXI =
     $,E15.8//9H TOLERANC,39HE FOR THE MULTIPLIERS ....... TOLMULT =,E15
     $.8//5H STOP,26HPING CRITERION FOR QUASID ,10(1H.),7H TOLX =,E15.8/
     $/48H INITIAL VALUE OF THE UPPER BOUND ...... UPBND =,E15.8//33H ST
     $ARTING POINT FOR THIS PROBLEM ,11(1H.),2H X,I2,E15.8/44X,99(I4,E15
     $.8/44X))
C
270   FORMAT (/3H X(,I3,35H) IS DISCRETE WITH AVAILABLE VALUES,I7,E15.8/
     141X,99(I7,E15.8/41X))
C
280   FORMAT (3H X(,I3,32H) IS UNIFORM STEP WITH STEP SIZE,9X,1H=,E15.8/
     1)
C
290   FORMAT (/18H OPTIONS IN EFFECT)
C
300   FORMAT (/37H GRADIENT CHECK AT THE STARTING POINT)
C
310   FORMAT (/47H ONE VARIABLE HELD CONSTANT DURING OPTIMIZATION)
C
320   FORMAT (/37H ONLY ONE DISCRETE SOLUTION REQUESTED)
C
330   FORMAT (/38H BRANCHING STARTS ON THE LAST VARIABLE)
C
340   FORMAT (/41H VERTICES AROUND NODE 0 SOLUTION EXAMINED)
C
350   FORMAT (/28H DETAILED PRINTING REQUESTED)
C
360   FORMAT (/38H OPTIMAL SOLUTION AT EACH NODE PRINTED)
C
370   FORMAT (/27H DISCRETE SOLUTIONS PRINTED)
C
380   FORMAT (///51H THE UPPER BOUND HAS BEEN UPDATED AT THIS NODE. THE,
     109H DISCRETE//40H SOLUTION AND THE CONSTRAINTS (CONS(1)=U,21HPPER
     2BOUND) FOLLOWING//28H A CHECK AT THE VERTICES SUR,32HROUNDING THE
     3NODE 0 SOLUTION ARE//4X,2HX ,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))
C
390   FORMAT (48H1OPTIMAL SOLUTION AT DAKIN BRANCH AND BOUND NODE,I4/1X,
     151(1H-))
C
400   FORMAT (28H THIS IS A DISCRETE SOLUTION/1X,27(1H-))
C
410   FORMAT (/53H THE X LESS THAN OR EQUAL TO KIND OF CONSTRAINTS AT T,
     112HHIS NODE ARE//6H X.LE.,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))
C
420   FORMAT (/53H THE X GREATER THAN OR EQUAL TO KIND OF CONSTRAINTS A,
     115HT THIS NODE ARE//6H X.GE.,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))
C
430   FORMAT (/19H THE SOLUTION WITH ,I3,27H CONSTRAINTS (CONS(1)=OBJEC,
     108HTIVE) IS///4X,2HX ,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))
C
440   FORMAT (/,1X,5HCONS ,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))
C
450   FORMAT (/53H THE TOTAL NUMBER OF FUNCTION EVALUATIONS PERFORMED S,
     18HO FAR IS,I5//13H OUT OF THESE,I5,18H WERE PERFORMED AT,10H THIS
     2NODE)
C
460   FORMAT (29H THIS SOLUTION IS NONFEASIBLE/1X,28(1H-))
C
470   FORMAT (//52H THE PROGRAM HAS STALLED BECAUSE OF INSUFFICIENT SPA,
     115HCE PROVIDED FOR/1X,66(1H=)/24H THE ADDITIONAL CONSTRAI,44HNTS.
     2PLEASE INCREASE THE VALUE OF IEXTRA. IT/1X,67(1H=)/53H MAY BE NOTE
     3D THAT THE DEFAULT VALUE OF IEXTRA IS 2*N/1X,52(1H=))
C
      END
```

```
      SUBROUTINE FIND (DIS,IAR,X)
C
C     THIS SUBROUTINE DETERMINES WHETHER OR NOT A SOLUTION IS DISCRETE.
C     IF NOT DISCRETE, IT FINDS THE NEAREST LOWER AND THE NEAREST UPPER
C     DISCRETE VALUES FOR THE FIRST NONDISCRETE VARIABLE ENCOUNTERED IN
C     THE SOLUTION
C
C     INPUT    DIS,IAR,IFIND,LARGE,NNCON,REVERSE,TOLDIS,X
C
C     OUTPUT   DISCRET,II,XL,XU
C
      DIMENSION DIS(1), IAR(1), X(1)
C
      REAL LARGE
      LOGICAL DISCRET,REVERSE
C
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /10/ GRADCHK,HOLDVAR,ONESOL,REVERSE,VERTCHK
C
      IA=1
      IF (NNCON.EQ.0) GO TO 90
      DISCRET=.TRUE.
C
      DO 80 IB=IFIND,NNCON
      IA=IB
      IF (REVERSE) IA=(NNCON+1)-IB
      I=2*IA-1
      IF (IAR(I).EQ.1) GO TO 40
      K=IAR(I+1)-1
      KD=IAR(I)
C
      DO 10 J=1,KD
      IF (X(IA).LT.DIS(J+K)) GO TO 20
10    CONTINUE
C
C     X(IA) LIES BEYOND THE LAST SPECIFIED DISCRETE VALUE
C
      XL=DIS(KD+K)
      XU=+LARGE
      GO TO 60
C
C     X(IA) LIES BETWEEN TWO SPECIFIED DISCRETE VALUES
C
20    CONTINUE
      IF (J.EQ.1) GO TO 30
      XL=DIS(J+K-1)
      XU=DIS(J+K)
      GO TO 60
C
C     X(IA) LIES BEFORE THE FIRST SPECIFIED DISCRETE VALUE
C
30    CONTINUE
      XL=-LARGE
      XU=DIS(K+1)
      GO TO 60
C
C     X(IA) IS A UNIFORMLY DISCRETE VARIABLE
C
40    Z=DIS(IAR(I+1))
      IF (X(IA).LT.0) GO TO 50
      XL=Z*FLOAT(IFIX(X(IA)/Z))
      XU=XL+Z
      GO TO 60
50    XU=Z*FLOAT(IFIX(X(IA)/Z))
      XL=XU-Z
C
C     CHECK IF X(IA) IS DISCRETE OR NOT
C
60    CONTINUE
      IF ((X(IA)-XL).LE.TOLDIS) GO TO 70
```

```
      IF ((XU-X(IA)).GT.TOLDIS) GO TO 90
      X(IA)=XU
      GO TO 80
70    X(IA)=XL
80    CONTINUE
C
      GO TO 100
90    CONTINUE
      DISCRET=.FALSE.
100   CONTINUE
      II=IA
      RETURN
      END
```

```
      SUBROUTINE GRDCHK3 (AL,CONS,ER,G,IDCONS,IDVAR,PERCENT,X,Y)
C
C     THIS SUBROUTINE IS CALLED ONLY ONCE BY DISOPT3 AT THE BEGINNING
C     TO VERIFY THAT THE GRADIENT VECTOR AS FORMULATED BY THE USER IS
C     CORRECT. THE GRADIENT VECTOR IS CALCULATED AT THE STARTING POINT
C     ONCE BY THE USERS DEFINITION AND AGAIN BY NUMERICALLY PERTURBING
C     POINT X. IF THE DIFFERENCE BETWEEN THE TWO VALUES EXCEEDS 10 P.C.
C     THE PROGRAM IS TERMINATED WITH A MESSAGE
C
C     INPUT    IDCONS,IDVAR,X
C
C     OUTPUT   G,PERCENT,WRONG,Y
C
      DIMENSION AL(1), CONS(1), ER(1), G(1), IDCONS(1), IDVAR(1), PERCEN
     1T(1), X(1), Y(1)
C
      LOGICAL FEASCHK,UONLY,WRONG
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /7/ IFN,IND1,IND2
      COMMON /9/ IEXIT,SKIPOBJ,UOBJ,UPDATED,WRONG
      COMMON /10/ GRADCHK,HOLDVAR,ONESOL,REVERSE,VERTCHK
C
      DO 10 I=1,NORCONS
      AL(I)=ALMIN
10    CONTINUE
C
      AL(1)=0.
      FEASCHK=.FALSE.
      HEXI=0.
      NA=NORCONS
      NCONS=NORCONS
      CALL OBJ (AL,CONS,1.,ER,G,IDCONS,IDVAR,1.,1.,X)
      UONLY=.TRUE.
      IND2=0
      WRONG=.FALSE.
C
C     TO CALCULATE G(I), AN ELEMENT OF THE GRADIENT VECTOR, X(I) IS
C     PERTURBED ONCE BY +DX AND ONCE BY -DX, AND THE FUNCTION EVALUATED
C     AT THESE POINTS. A SIMPLE DIVISION YIELDS THE VALUE OF G(I).
C
      DO 20 I=1,N
      Z=X(I)
      DELX=1.E-4*Z
      IF (ABS(Z).LT.1.E-10) DELX=1.E-10
      X(I)=Z+DELX
      CALL OBJ (AL,CONS,1.,ER,1.,IDCONS,IDVAR,1.,1.,X)
      U2=UOBJ
      X(I)=Z-DELX
      CALL OBJ (AL,CONS,1.,ER,1.,IDCONS,IDVAR,1.,1.,X)
      U1=UOBJ
      X(I)=Z
      Z=.5*(U2-U1)/DELX
      ZZ=G(I)
      IF (ABS(Z).LT.1.E-20) Z=1.E-20
      IF (ABS(ZZ).LT.1.E-20) ZZ=1.E-20
      PERCENT(I)=ABS((Z-ZZ)/Z)*100.
      Y(I)=Z
      IF (PERCENT(I).GT.10.) WRONG=.TRUE.
20    CONTINUE
C
      UONLY=.FALSE.
      IND2=1
      IF (JPT.LT.0) GO TO 30
      PRINT 40, (I,G(I),I,Y(I),I,PERCENT(I),I=1,N)
      IF (WRONG) PRINT 50
      IF (.NOT.WRONG) PRINT 60
30    RETURN
C
40    FORMAT (37H1GRADIENT CHECK AT THE STARTING POINT/1X,36(1H-)//12X,1
```

```
     10HANALYTICAL,9X,9HNUMERICAL,10X,10HPERCENTAGE/13X,8HGRADIENT,11X,8GRD 740
     2HGRADIENT,13X,5HERROR/12X,11HVECTOR G(I),8X,11HVECTOR Y(I),6X,17HVGRD 750
     3ECTOR PERCENT(I)//6X,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))           GRD 760
C                                                                       GRD 770
50    FORMAT (///51H YOUR PROGRAM HAS BEEN TERMINATED BECAUSE THE GRADI,GRD 780
     118HENTS ARE INCORRECT//24H PLEASE CHECK THEM AGAIN)               GRD 790
C                                                                       GRD 800
60    FORMAT (///35H THE GRADIENTS APPEAR TO BE CORRECT)                GRD 810
C                                                                       GRD 820
      END                                                               GRD 830-
```

```
      SUBROUTINE OBJ (AL,CONS,DIS,ER,GRADU,IDCONS,IDVAR,INT,RMULT,X)
C
C     THIS SUBROUTINE GENERATES THE ADDITIONAL CONSTRAINTS REQUIRED FOR
C     DISCRETE OPTIMIZATION, CONVERTS THE CONSTRAINED PROBLEM INTO A
C     MINIMAX PROBLEM USING THE BANDLER-CHARALAMBOUS TECHNIQUE, AND THEN
C     EVALUATES THE ERROR FUNCTIONS FOR THIS MINIMAX PROBLEM
C
C     INPUT   AL,DIS,FEASCHK,HEXI,IDCONS,INT,IP,MULTS,N,NA,NODE,NORCONS,
C             UPBND,X
C
C     OUTPUT CONS,ER,FEASBLE,CCONS,GRADU,RMULT,UOBJ
C
      DIMENSION AL(1), CONS(1), DIS(1), ER(1), GRADU(1), IDCONS(1), IDVA
     1R(1), INT(1), RMULT(1), X(1)
C
      LOGICAL FEASCHK,SKIPOBJ
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /3/ IEXTRA,LASTDIS,LASTIAR,NODE,NODES
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /8/ LAL,LCONS,LER,LCCONS,LGRADU,LH,LINT,LRMULT,LW,LX,LY
      COMMON /9/ IEXIT,SKIPOBJ,UOBJ,UPDATED,WRONG
C
C     EVALUATE THE ORIGINAL CONSTRAINTS AND THEIR GRADIENT VECTORS
C
      CALL FUN (CONS,X(LCCONS),IDCONS,IDVAR,X)
      IF (NODE.EQ.0) GO TO 20
C
C     EVALUATE THE ADDITIONAL CONSTRAINTS FOR THE NODES IN THE TREE
C
      DO 10 I=1,NA
      J=IDCONS(I)
      IF (J.LE.NORCONS) GO TO 10
      JL=J-NORCONS
      JL4=(JL-1)*4+1
      IF (INT(JL4).EQ.0) CONS(J)=DIS(INT(JL4+2))-X(INT(JL4+1))
      IF (INT(JL4).NE.0) CONS(J)=X(INT(JL4+1))-DIS(INT(JL4+2)+1)
10    CONTINUE
C
20    CONTINUE
      IF (SKIPOBJ) GO TO 70
      IF (FEASCHK) GO TO 40
C
C     EVALUATE ERROR FUNCTIONS FOR AN OPTIMIZATION. AL(1)=0
C
      Z=CONS(1)-HEXI
C
      DO 30 I=1,NA
      J=IDCONS(I)
      ER(J)=Z-AL(J)*CONS(J)
30    CONTINUE
C
      GO TO 60
C
C     EVALUATE ERROR FUNCTIONS FOR A FEASIBILITY CHECK. ALL THE CONSTRA-
C     INTS ARE ACTIVE DURING A FEASIBILITY CHECK
C
40    DO 50 I=1,NCONS
      ER(I)=-CONS(I)
50    CONTINUE
C
      ER(1)=CONS(1)-UPBND
C
C     EVALUATE THE LEAST PTH OBJECTIVE FUNCTION AND ITS GRADIENT VECTOR
C
60    CALL LEASTPD (AL,ER,ER(NCONS+1),X(LCCONS),GRADU,IDCONS,IDVAR,INT,R
     1MULT)
70    CONTINUE
      RETURN
      END
```

```
      SUM2=0.
      Z=GCONS(I)
      IF (FEASCHK) GCONS(I)=-GCONS(I)
      ID=-NX+I
C
      DO 80 J=1,NA
      K=IDCONS(J)
      IF (POSITIV.AND.(ER(K).LE.0.)) GO TO 80
      IF (K.GT.NORCONS) GO TO 50
      IF (FEASCHK) SUM2=SUM2-ES(K)*GCONS(K*NX+(ID))
      IF (.NOT.FEASCHK) SUM2=SUM2+ES(K)*(Z-AL(K)*GCONS(K*NX+(ID)))
      GO TO 80
50    KL4=(K-NORCONS)*2*2-3
      IF (INT(KL4+1).NE.I) GO TO 60
      ZZ=+1.
      IF (INT(KL4).EQ.0) ZZ=-1.
      GO TO 70
60    ZZ=0.
70    IF (FEASCHK) SUM2=SUM2-ES(K)*ZZ
      IF (.NOT.FEASCHK) SUM2=SUM2+ES(K)*(Z-AL(K)*ZZ)
80    CONTINUE
C
      GRADU(IA)=SUM1*SUM2
90    CONTINUE
C
      IF (FEASCHK.AND.((ERMAX+TOLCONS).LT.0.)) FEASBLE=.TRUE.
      GO TO 130
100   CONTINUE
C
C     CALCULATE THE MULTIPLIERS FOR THE ACTIVE FUNCTIONS. NA MUST BE
C     .GE. 2
C
      SUM1=ES(1)
C
      DO 110 I=2,NA
      J=IDCONS(I)
      IF (POSITIV.AND.(ER(J).LE.0.)) GO TO 110
      SUM1=SUM1+ES(J)
110   CONTINUE
C
      DO 120 I=2,NA
      J=IDCONS(I)
      RMULT(J)=0.
      IF (POSITIV.AND.(ER(J).LE.0.)) GO TO 120
      RMULT(J)=AL(J)*ES(J)/SUM1
120   CONTINUE
C
      MULTS=.FALSE.
130   CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LEASTPD (AL,ER,ES,GCONS,GRADU,IDCONS,IDVAR,INT,RMULT)
C
C     THIS SUBROUTINE EVALUATES THE CHARALAMBOUS LEAST PTH UNCONSTRAINED
C     FUNCTION AND ITS GRADIENT VECTOR. IF MULTS, A LOGICAL VARIABLE, IS
C     TRUE, THE GRADIENT VECTOR IS NOT EVALUATED AND, INSTEAD, THE MULTI
C     PLIERS FOR THE ERROR FUNCTIONS ARE EVALUATED
C
C      INPUT  AL,ER,FEASCHK,GCONS,IDCONS,INT,IP,MULTS,N,NA,NORCONS
C
C     OUTPUT ERMAX,FEASBLE,GRADU,RMULT,UOBJ
C
      DIMENSION AL(1), ER(1), ES(1), GCONS(1), GRADU(1), IDCONS(1), IDVA
     1R(1), INT(1), RMULT(1)
C
      LOGICAL FEASBLE,FEASCHK,MULTS,POSITIV,UONLY
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /9/ IEXIT,SKIPOBJ,UOBJ,UPDATED,WRONG
C
C     EVALUATE THE MAXIMUM OF THE ERROR FUNCTIONS AND IF IT IS 0 SUB-
C     TRACT 1.E-10 FROM EVERY ERROR FUNCTION
C
      ERMAX=ER(IDCONS(1))
C
      DO 10 I=1,NA
      ERMAX=AMAX1(ERMAX,ER(IDCONS(I)))
10    CONTINUE
C
       IF (ERMAX.NE.0.) GO TO 30
C
      ERMAX=-1.E-10
C
      DO 20 I=1,NA
      J=IDCONS(I)
      ER(J)=ER(J)-1.E-10
20    CONTINUE
C
C     EVALUATE FEASBLE, POSITIV AND IPL
C
30    POSITIV=.FALSE.
       IF (ERMAX.GT.0.) POSITIV=.TRUE.
       IPL=-IP
       IF (POSITIV) IPL=IP
C
C     EVALUATE THE LEAST PTH OBJECTIVE FUNCTION
C
      SUM1=0.
C
      DO 40 I=1,NA
      J=IDCONS(I)
      Z=ER(J)
       IF (POSITIV.AND.(Z.LE.0.)) GO TO 40
      Z=Z/ERMAX
      ZZ=Z**IPL
      SUM1=SUM1+ZZ
      ES(J)=ZZ/Z
40    CONTINUE
C
      Z=SUM1**(1./FLOAT(IPL))
      UOBJ=Z*ERMAX
      SUM1=Z/SUM1
       IF (UONLY) GO TO 130
       IF (MULTS) GO TO 100
C
C     EVALUATE THE GRADIENT VECTOR. THE OBJECTIVE FUNCTION IS ALWAYS
C     THE FIRST ACTIVE FUNCTION
C
      DO 90 IA=1,N
      I=IDVAR(IA)
```

```
      SUBROUTINE QUASID (AL,CONS,DIS,ER,G,H,IDCONS,IDVAR,INT,W,X)
C
C     THIS SUBROUTINE IS BASED ON THE 1972 VERSION OF FLETCHERS
C     METHOD OF UNCONSTRAINED OPTIMIZATION. WITHOUT DISTURBING THE BASIC
C     LOGIC OF THE FLETCHERS ORIGINAL SUBROUTINE, SOME CHANGES HAVE BEEN
C     MADE IN THIS PROGRAM. THESE CHANGES ARE (1) THE PART WHICH DECOM-
C     POSES H INTO LDL(TRANSPOSE) HAS BEEN REMOVED (2) THE PART WHICH
C     FINDS DMIN HAS BEEN REMOVED (3) THIS SUBROUTINE REQUIRES IDVAR AS
C     AN INPUT. IN ADDITION, IT ASSUMES THAT THE GRADIENT VECTOR HAS
C     BEEN SUITABLY CALCULATED. FOR EXAMPLE, LET THE ORIGINAL PROBLEM
C     HAVE THREE VARIABLES X(1), X(2) AND X(3). FOR SOME OPTIMIZATION IT
C     IS DECIDED TO HOLD X(2) CONSTANT. NOW, IN ORDER TO USE QUASID, WE
C     MUST HAVE N=2, IDVAR(1)=1, IDVAR(2)=3, G(1)=DEL UOBJ/DEL X(1) AND
C     G(2)=DEL UOBJ/DEL X(3). IN OTHER WORDS, VECTOR G, AN ARRAY OF N
C     ELEMENTS, STORES THE PARTIAL DERIVATIVES OF UOBJ WITH RESPECT TO
C     THE ACTIVE VARIABLES IN THE SAME ORDER IN WHICH THE INDICES OF
C     THE ACTIVE VARIABLES ARE STORED IN ARRAY IDVAR
C
C     INPUT    AL,DIS,EST,FEASCHK,FEASBLE,IDCONS,IDVAR,IFN,INT,IPT,
C              MAXIFN,MODE,TOLX,X
C
C     OUTPUT   IEXIT,X
C
      DIMENSION AL(1), CONS(1), DIS(1), ER(1), G(1), H(1), IDCONS(1), ID
     1VAR(1), INT(1), W(1), X(1)
C
      LOGICAL FEASBLE,FEASCHK
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /3/ IEXTRA,LASTDIS,LASTIAR,NODE,NODES
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN,DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /7/ IFN,IND1,IND2
      COMMON /9/ IEXIT,SKIPOBJ,UOBJ,UPDATED,WRONG
C
C     INITIALIZATION
C
      IF (FEASCHK.AND.(IPT.GT.0)) PRINT 410, NODES
      IF (IPT.GT.0) PRINT 400
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=0
      IF (MODE.NE.1) GO TO 30
C
C     THE INITIAL ESTIMATE OF H, AN IDENTITY MATRIX, IS GENERATED HERE
C
      IJ=NN+1
C
      DO 20 I=1,N
      DO 10 J=1,I
      IJ=IJ-1
      H(IJ)=0.
10    CONTINUE
      H(IJ)=1.
20    CONTINUE
      DMIN=1.
C
C     INITIAL PRINTING AND INITIALIZATION
C
30    Z=EST
      ITN=0
      CALL OBJ (AL,CONS,DIS,ER,G,IDCONS,IDVAR,INT,1.,X)
      U=UOBJ
      IF (FEASCHK.AND.FEASBLE) GO TO 320
      DF=U-EST
      IF (DF.LE.0.0) DF=1.0
```

```
40    IF (IPT.EQ.0.OR.MOD(ITN,IPT).NE.0) GO TO 50
      PRINT 370, ITN,IFN,U,(IDVAR(I),X(IDVAR(I)),IDVAR(I),G(I),I=1,N)
C
C     AN ITERATION OF QUASID BEGINS. IT INVOLVES SELECTION OF ALPHA,
C     THE LINE SEARCH PARAMETER, AND UPDATING OF H FOR THE NEXT ITERA-
C     TION OF QUASID
C
50    ITN=ITN+1
C
C     THE DIRECTION OF SEARCH, WHICH IS THE PRODUCT OF THE INVERSE OF
C     THE HESSIAN H WITH THE GRADIENT VECTOR G, IS FOUND HERE. THE ELE-
C     MENTS OF THIS VECTOR ARE W(N+1), W(N+2),  ......  , W(2N)
C
      W(1)=-G(1)
C
      DO 70 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 60 J=1,I1
      Z=Z-H(IJ)*W(J)
      IJ=IJ+N-J
60    CONTINUE
      W(I)=Z
70    CONTINUE
C
      W(IS+N)=W(N)/H(NN)
      IJ=NN
C
      DO 90 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 80 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
      IJ=IJ-1
80    CONTINUE
      W(IS+N-I)=W(N-I)/H(IJ)-Z
90    CONTINUE
C
C     THE SCALAR PRODUCT OF G WITH THE DIRECTION OF SEARCH IS NOW FOUND.
C     IT MUST BE NEGATIVE OR ELSE THE FUNCTION CAN NOT BE MINIMIZED ANY
C     FURTHER. GS IS TESTED TO ENSURE THIS
C
      GS=0.
C
      DO 100 I=1,N
      GS=GS+W(IS+I)*G(I)
100   CONTINUE
C
      IEXIT=2
      IF (GS.GE.0.) GO TO 320
C
C     ALPHA, THE LINE SEARCH PARAMETER, WILL NOW BE CALCULATED USING
C     EITHER THE QUADRATIC FIT, THE CUBIC INTERPOLATION, OR THE LINEAR
C     EXTRAPOLATION. AN INEXACT LINE SEARCH IS MADE HERE
C
      GSO=GS
      ALPHA=-2.*DF/GS
      IF (ALPHA.GT.1.) ALPHA=1.
      DF=U
      TOT=0.
110   IEXIT=3
      IF (IFN.GE.MAXIFN) GO TO 320
      ICON=0
      IEXIT=1
C
      DO 120 I=1,N
      Z=ALPHA*W(IS+I)
      II=IDVAR(I)
      IF (ABS(Z).GE.TOLX) ICON=1
      X(II)=X(II)+Z
120   CONTINUE
C
```

```
      CALL OBJ (AL,CONS,DIS,ER,W,IDCONS,IDVAR,INT,1.,X)
      IF (FEASCHK.AND.FEASBLE) GO TO 300
      FY=UOBJ
C
C     ELEMENTS W(1),W(2), .... ,W(N) NOW CONTAIN THE GRADIENT VECTOR.
C     GYS, IN THE FOLLOWING SECTION, IS THE SCALAR PRODUCT OF THE GRAD-
C     IENT AT THE NEXT POINT WITH THE PRESENT DIRECTION OF SEARCH
C
      GYS=0.
C
      DO 130 I=1,N
      GYS=GYS+W(I)*W(IS+I)
130   CONTINUE
C
      IF (FY.GE.U) GO TO 140
      IF (ABS(GYS/GS0).LE..9) GO TO 160
      IF (GYS.GT.0.) GO TO 140
C
C     LINEAR EXTRAPOLATION FOR ALPHA IS PERFORMED HERE
C
      TOT=TOT+ALPHA
      Z=10.
      IF (GS.LT.GYS) Z=GYS/(GS-GYS)
      IF (Z.GT.10.) Z=10.
      ALPHA=ALPHA*Z
      U=FY
      GS=GYS
      GO TO 110
C
C     CUBIC INTERPOLATION TO FIND ALPHA IS PERFORMED HERE
C
140   DO 150 I=1,N
      I1=IDVAR(I)
      X(I1)=X(I1)-ALPHA*W(IS+I)
150   CONTINUE
C
      IF (ICON.EQ.0) GO TO 320
      Z=3.*(U-FY)/ALPHA+GYS+GS
      ZZ=SQRT(Z*Z-GS*GYS)
      GZ=GYS+ZZ
      Z=1.-(GZ-Z)/(ZZ+GZ-GS)
      ALPHA=ALPHA*Z
      GO TO 110
C
C     THE LINE SEARCH HAS BEEN COMPLETED AND A NEW POINT HAS BEEN OB-
C     TAINED. H MUST BE UPDATED NOW
C
160   ALPHA=TOT+ALPHA
      U=FY
      IF (ICON.EQ.0) GO TO 300
      DF=DF-U
      DGS=GYS-GS0
      LINK=1
C
C     IF THE FOLLOWING TEST IS TRUE, THE DFP FORMULA WILL BE USED FOR
C     UPDATING H, OTHERWISE, THE COMPLEMENTARY DFP FORMULA WILL BE USED
C
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 180
C
      DO 170 I=1,N
      W(IU+I)=W(I)-G(I)
170   CONTINUE
C
      SIG=1./(ALPHA*DGS)
      GO TO 250
180   ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.
C
      DO 190 I=1,N
      W(IU+I)=Z*G(I)+W(I)
190   CONTINUE
C
      SIG=1./(ZZ*DGS*DGS)
```

```
      GO TO 250
200   LINK=2
C
      DO 210 I=1,N
      W(IU+I)=G(I)
210   CONTINUE
C
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 220
      SIG=1./GS0
      GO TO 250
220   SIG=-ZZ
      GO TO 250
C
230   DO 240 I=1,N
      G(I)=W(I)
240   CONTINUE
C
      GO TO 40
250   W(IV+1)=W(IU+1)
C
      DO 270 I=2,N
      IJ=I
      I1=I-1
      Z=W(IU+I)
      DO 260 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
      IJ=IJ+N-J
260   CONTINUE
      W(IV+I)=Z
270   CONTINUE
C
      IJ=1
C
      DO 280 I=1,N
      IVI=IV+I
      IBI=IB+I
      Z=H(IJ)+SIG*W(IVI)*W(IVI)
      IF (Z.LE.0.) Z=DMIN
      IF (Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IBI)=W(IVI)*SIG/Z
      SIG=SIG-W(IBI)*W(IBI)*Z
      IJ=IJ+NP-I
280   CONTINUE
C
      IJ=1
C
      DO 290 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 290 J=I1,N
      W(IU+J)=W(IU+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(IU+J)
290   IJ=IJ+1
C
      IF (LINK-2) 200,230,230
C
C     THE UPDATING OF H IS NOW COMPLETE AND THE NEXT ITERATION BEGINS
C
300   DO 310 I=1,N
      G(I)=W(I)
310   CONTINUE
C
320   IF (IPT.EQ.0) GO TO 330
      PRINT 370, ITN,IFN,U,(IDVAR(I),X(IDVAR(I)),IDVAR(I),G(I),I=1,N)
330   IF (IPT.LT.0) GO TO 360
      IF (IEXIT-2) 360,340,350
340   PRINT 380, IEXIT
      GO TO 360
350   PRINT 390, IEXIT
360   RETURN
C
370   FORMAT (1H0,I3,2X,I4,E15.8,99(I4,E15.8,I4,E15.8/25X))
```

```
C                                                                     QUA2930
380   FORMAT (8H1IEXIT =,I2,40H THE PROGRAM IS UNABLE TO FIND A DOWNHIL,QUA2940
     113HL DIRECTION. //36H POSSIBLE CAUSES ARE (1) EXCESSIVE R,36HOUND QUA2950
     2OFF ERROR DUE TO VERY SMALL EPS//21X,9H(2) ERROR,41HIN THE CALCULAQUA2960
     3TION OF THE GRADIENT VECTOR)                                     QUA2970
C                                                                     QUA2980
390   FORMAT (8H1IEXIT =,I2,40H PERMISSIBLE NUMBER OF FUNCTION EVALUATI,QUA2990
     136HONS AT THIS NODE HAVE BEEN PERFORMED)                         QUA3000
C                                                                     QUA3010
400   FORMAT (53H UNCONSTRAINED OPTIMIZATION USING 1972 VERSION OF FLE,1QUA3020
     13HTCHERS METHOD/1X,65(1H-)//1X,21HITER. FUNC. LEAST PTH,10X,8HVARIQUA3030
     2ABLE,11X,8HGRADIENT/2X,19HNO.  EVAL. FUNCTION,10X,11HVECTOR X(I),8QUA3040
     3X,11HVECTOR G(I))                                                QUA3050
C                                                                     QUA3060
410   FORMAT (26H1FEASIBILITY CHECK AT NODE,I4/1X,29(1H-))             QUA3070
C                                                                     QUA3080
      END                                                             QUA3090-
```

```
      SUBROUTINE UOPT (AL,CONS,DIS,ER,G,H,IDCONS,IDVAR,INT,JDCONS,RMULT,
     1W,X,Y)
C
C     THIS SUBROUTINE SOLVES THE NONLINEAR PROGRAMMING PROBLEM AT EVERY
C     NODE. THE ALGORITHM EMPLOYED HERE IS THE ONE THAT HAS BEEN PRESEN-
C     TED BY CHARALAMBOUS IN HIS PAPER ON NONLINEAR LEAST PTH OPTIMIZA-
C     TION AND NONLINEAR PROGRAMMING IN MATH. PROGRAMMING, VOL.12,1977.
C     THIS ALGORITHM, AS IMPLEMENTED IN THIS PROGRAM, HAS SOME NOTABLE
C     FEATURES (1) BEFORE SOLVING THE ACTUAL PROBLEM, A FEASIBILITY
C     CHECK IS MADE TO ENSURE THE EXISTENCE OF A FEASIBLE POINT. IF NO
C     SUCH POINT EXISTS THEN NO ATTEMPT IS MADE TO SOLVE THE PROBLEM
C     (2) WHILE THE ACTUAL PROBLEM IS BEING SOLVED, IF AN ITERATION
C     LEADS TO A NONFEASIBLE POINT, THE STARTING POINT FOR THE NEXT ITE-
C     RATION IS NOT THE SAME POINT BUT A PREVIOUSLY OBTAINED FEASIBLE
C     POINT
C
C     INPUT   AL,DIS,HEXI,INT,IP,N,NODE,NORCONS,TOLCONS,TOLHEXI,TOLMULT,
C             TOLX,UPBND,X
C
C     OUTPUT FEASBLE,X
C
      DIMENSION AL(1), CONS(1), DIS(1), ER(1), G(1), H(1), IDCONS(1), ID
     1VAR(1), INT(1), JDCONS(1), RMULT(1), W(1), X(1), Y(1)
C
      LOGICAL FEASBLE,FEASCHK,FLAG,MULTS,REDUCE,UONLY
C
      COMMON /1/ IP,MAXNODE,N,NORCONS,PRINTID,PRINTP
      COMMON /2/ LARGE,TOLCONS,TOLDIS,TOLHEXI,TOLMULT,TOLX
      COMMON /3/ IEXTRA,LASTDIS,LASTIAR,NODE,NODES
      COMMON /4/ DISCRET,FEASBLE,FEASCHK,MULTS,UONLY
      COMMON /5/ IFIND,II,IPT,JPT,MAXIFN,MAXITN,MODE,NA,NCONS,NNCON,NX
      COMMON /6/ ALMIN.DMIN,ERMAX,EST,HEXI,UPBND,XL,XU
      COMMON /7/ IFN,IND1,IND2
C
C     PERFORM A FEASIBILITY CHECK FIRST TO ENSURE THE EXISTENCE OF A
C     FEASIBLE SOLUTION
C
      FEASBLE=.FALSE.
      FEASCHK=.TRUE.
      NCONS=NORCONS+NODE
      NA=NCONS
C
      DO 10 I=1,NA
      IDCONS(I)=I
10    CONTINUE
C
      IPD=IP
      IP=2
      CALL QUASID (AL,CONS,DIS,ER,G,H,IDCONS,IDVAR,INT,W,X)
      IP=IPD
      FEASCHK=.FALSE.
      IF (.NOT.FEASBLE) GO TO 210
C
C     PERFORM AN OPTIMIZATION ONLY IF A FEASIBLE POINT HAS BEEN FOUND
C
      DO 20 I=1,N
      J=IDVAR(I)
      Y(J)=X(J)
20    CONTINUE
C
      ALMAX=ALMIN
      K=-1
      KOUNT=0
      REDUCE=.TRUE.
30    CONTINUE
C
      DO 40 I=1,NA
      AL(I)=ALMAX
40    CONTINUE
C
      AL(1)=0.
C
C     DO NOT REINITIALIZE THE STARTING POINT IF IT HAPPENS TO BE A FEAS-
```

```
C       IBLE POINT
C
        IF (FEASBLE) GO TO 60
C
        DO 50 I=1,N
        J=IDVAR(I)
        X(J)=Y(J)
50      CONTINUE
C
        IF (IPT.GT.0) PRINT 220, (I,CONS(I),I=1,NCONS)
60      CONTINUE
        HEXI=0.
        UONLY=.TRUE.
        IND2=0
        CALL OBJ (AL,CONS,DIS,ER,1.,IDCONS,IDVAR,INT,1.,X)
        IND2=1
        UONLY=.FALSE.
        HEXI=AMIN1(HEXI,ERMAX+1.E-10)
70      CONTINUE
        IF (KOUNT.GE.MAXITN) GO TO 200
        KOUNT=KOUNT+1
C
C       PERFORM THE NECESSARY PRINTING
C
        IF (IPT.LE.0) GO TO 100
        PRINT 230, KOUNT,NODES,HEXI
        FLAG=.FALSE.
        IF (REDUCE.AND.(K.GT.0)) FLAG=.TRUE.
        IF (FLAG) PRINT 260, CONS(1)
        IF (.NOT.FLAG) PRINT 270, CONS(1)
        IF (NCONS.EQ.1) GO TO 100
        J=2
C
        DO 90 I=2,NCONS
        IF (I.EQ.IDCONS(J)) GO TO 80
        PRINT 280, I,RMULT(I),I,I,CONS(I)
        GO TO 90
80      CONTINUE
        IF (FLAG) PRINT 290, I,RMULT(I),I,AL(I),I,CONS(I)
        IF (.NOT.FLAG) PRINT 250, I,AL(I),I,CONS(I)
        IF (J.LT.NA) J=J+1
90      CONTINUE
C
100     CONTINUE
        PRINT 240
C
C       PERFORM THE UNCONSTRAINED OPTIMIZATION
C
        CALL QUASID (AL,CONS,DIS,ER,G,H,IDCONS,IDVAR,INT,W,X)
C
C       CHECK THE FEASIBILITY OF THE SOLUTION
C
        IF (NCONS.EQ.1) GO TO 140
        NAD=NA
        NA=NCONS
        IND2=0
        UONLY=.TRUE.
        CALL OBJ (AL,CONS,DIS,ER,1.,JDCONS,IDVAR,INT,1.,X)
        UONLY=.FALSE.
        IND2=1
        NA=NAD
        FEASBLE=.TRUE.
C
        DO 110 I=2,NCONS
        IF (CONS(I).LT.TOLCONS) GO TO 120
110     CONTINUE
C
        GO TO 140
120     IF (K.LE.0) ALMAX=ALMAX*10.
        FEASBLE=.FALSE.
        IF ((K.LE.0).OR.(.NOT.REDUCE)) GO TO 30
        REDUCE=.FALSE.
        DO 130 I=1,NCONS
```

```
      IDCONS(I)=I
130   CONTINUE
C
      NA=NCONS
      GO TO 30
C
C     SELECT ACTIVE FUNCTIONS
C
140   CONTINUE
      K=K+1
      IF ((K.LE.0).OR.(.NOT.REDUCE).OR.(NA.EQ.1)) GO TO 180
      IND2=0
      MULTS=.TRUE.
      CALL OBJ (AL,CONS,DIS,ER,1.,IDCONS,IDVAR,INT,RMULT,X)
      NAD=2
C
      DO 150 I=2,NA
      J=IDCONS(I)
      IF (RMULT(J).LE.TOLMULT) GO TO 150
      IDCONS(NAD)=J
      NAD=NAD+1
150   CONTINUE
C
      DO 160 I=NAD,NORCONS
      IDCONS(I)=(NORCONS+1)
160   CONTINUE
C
      NA=NAD-1
C
C     UPDATE ELEMENTS OF VECTOR AL CORRESPONDING TO THE ACTIVE FUNCTIONS
C     AND CALCULATE ERMAX FOR THESE NEW FUNCTIONS
C
      DO 170 I=2,NA
      J=IDCONS(I)
      AL(J)=FLOAT(NA)*RMULT(J)
170   CONTINUE
C
      UONLY=.TRUE.
      CALL OBJ (AL,CONS,DIS,ER,1.,IDCONS,IDVAR,INT,1.,X)
      UONLY=.FALSE.
      IND2=1
C
C     CHECK THE STOPPING CRITERION. IF IT IS NOT SATISFIED UPDATE HEXI
C
180   CONTINUE
      HEXID=HEXI
      IF (HEXID.EQ.0) HEXID=1.E-10
      IF (ABS((ERMAX+1.E-10)/HEXID).LE.TOLHEXI) GO TO 200
      HEXI=HEXI+ERMAX+1.E-10
C
      DO 190 I=1,N
      J=IDVAR(I)
      Y(J)=X(J)
190   CONTINUE
C
      GO TO 70
200   CONTINUE
210   CONTINUE
      RETURN
C
220    FORMAT (/64H THE ABOVE ITERATION HAS RESULTED IN A NONFEASIBLE SOL
     1UTION. THE//64H CONSTRAINTS AT THIS POINT ARE GIVEN AS FOLLOWS. IT
     2 MAY BE NOTED//64H THAT THE STARTING POINT FOR THE NEXT ITERATION
     3IS NOT THE ABOVE//53H SOLUTION BUT THE BEST FEASIBLE POINT OBTAINE
     4D SO FAR//6H CONS ,99(I4,E15.8,I4,E15.8,I4,E15.8/6X))
C
230    FORMAT (17H1ITERATION NUMBER,I3,30H OF THE CHARALAMBOUS METHOD (L,
     118HEAST PTH APPROACH)/1X,67(1H-)/21H FOR THE NONLINEAR PR,25HOGRAM
     2MING PROBLEM AT NODE,I4/1X,49(1H-)//10H VALUE OF ,38HHEXI FOR THIS
     3 ITERATION ....... HEXI =,E15.8//14X,2HMU,8HLTIPLIER,10X,5HALPHA,1
     42X,10HCONSTRAINT/12X,09HVECTOR RM,6HULT(I),4X,12HVECTOR AL(I),7X,1
     54HVECTOR CONS(I)/)
C
```

```
240     FORMAT (1X)
C
250     FORMAT (25X,I4,E15.8,I4,E15.8)
C
260     FORMAT (9X,9H1  ACTIVE,10X,12H1  OBJECTIVE,7X,1H1,E15.8)
C
270     FORMAT (12X,28HNOT CALCULATED  1  OBJECTIVE,7X,1H1,E15.8)
C
280     FORMAT (6X,I4,E15.8,I4,10H  INACTIVE,5X,I4,E15.8)
C
290     FORMAT (6X,3(I4,E15.8))
C
        END
```

Appendix 2

REFERENCES

[1] J.W. Bandler and C. Charalambous, "Nonlinear programming using minimax techniques", J. Optimization Theory and Applications, vol. 13, 1974, pp. 607-619.

[2] C. Charalambous, "Nonlinear least pth optimization and nonlinear programming", Mathematical Programming, vol. 12, 1977, pp. 195-225.

[3] R. Fletcher, "FORTRAN subroutines for minimization by quasi-Newton methods", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R7125, 1972.

[4] R.J. Dakin, "A tree-search algorithm for mixed integer programming problems", Computer J., vol. 8, 1966, pp. 250-255.

[5] J.H.K. Chen, "DISOPT - a general program for continuous and discrete nonlinear programming problems", McMaster University, Hamilton, Canada, Report SOC-29, March 1974 (Revised June 1975).

[6] J.W. Bandler and J.H.K. Chen, "DISOPT - a general program for continuous and discrete nonlinear programming problems", Int. J. Systems Science, vol. 6, 1975, pp. 665-680.

[7] J.W. Bandler and C. Charalambous, "Practical least pth optimization of networks", IEEE Trans. Microwave Theory Tech., vol. MTT-20, 1972, pp. 834-840.

[8] C. Charalambous and J.W. Bandler, "New algorithms for network optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-21, 1973, pp. 815-818.

[9] J.W. Bandler, C. Charalambous, J.H.K. Chen and W.Y. Chu, "New results in the least pth approach to minimax design", IEEE Trans. Microwave Theory Tech., vol. MTT-24, 1976, pp. 116-119.

[10] C. Charalambous and J.W. Bandler, "Nonlinear minimax optimization as a sequence of least pth optimization with finite value of p", Int. J. Systems Science, vol. 7, 1976, pp. 377-391.

[11] C. Charalambous, "A unified review of optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-22, 1974, pp. 289-300.

[12] J. Asaadi, "A computational comparison of some non-linear programs", Mathematical Programming, vol. 4, 1973, pp. 144-154.

[13] B.J. Karafin, "The optimum assignment of component tolerances for electrical networks", BSTJ, vol. 50, 1971, pp. 1255-1242.

SOC-174

DISOPT3 - A USER-ORIENTED PACKAGE FOR NONLINEAR CONTINUOUS AND DISCRETE OPTIMIZATION PROBLEMS

J.W. Bandler and D. Sinha

July 1977, No. of Pages: 188

Revised:

Key Words: Engineering optimization, nonlinear programming, discrete optimization, least pth optimization, branch and bound method, computer programs

Abstract: A package of FORTRAN subroutines called DISOPT3 for solving continuous and discrete, constrained or unconstrained general optimization problems is presented. The method used for arriving at the discrete solution involves conversion of the original constrained problem into a minimax problem by the Bandler-Charalambous technique, solving the continuous minimax problem using the latest (1977) Charalambous least pth algorithm, Fletcher's 1972 method for unconstrained minimization and use of the Dakin branch and bound technique to generate the additional constraints. These steps are iteratively implemented until all the discrete solutions have been found. DISOPT3 is based conceptually on the DISOPT program developed by Bandler and Chen. All of the desirable features of DISOPT have been retained in DISOPT3 and some more have been added. DISOPT has been used as a yardstick against which the performance and validity of DISOPT3 have been measured. A CDC 6400 computer was used for developing and running this program.

Description: Contains Fortran listing, user's manual.
Source deck available for $400.00.
The listing contains 1707 cards, of which 708 are comment cards.

Related Work: As for SOC-29. Represents a complete restructuring and redevelopment of work presented in SOC-29.

Price: $ 150.00.