INTERNAL REPORTS IN

SIMULATION, OPTIMIZATION AND CONTROL

No. SOC-70

MINOPT - AN OPTIMIZATION PROGRAM BASED ON RECENT MINIMAX RESULTS

J.W. Bandler, C. Charalambous and J.H.K. Chen

December 1974

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

MINOPT - AN OPTIMIZATION PROGRAM BASED ON RECENT MINIMAX RESULTS

J.W. Bandler, C. Charalambous and J.H.K. Chen

1. Purpose

MINOPT is a package of subroutines for solving minimax problems. That is, it minimizes the function

$$M_a(\underset{\sim}{x}) \triangleq \max_{i \in I} a_i(\underset{\sim}{x}), \quad I \triangleq \{1,2,\ldots,m\}$$

where the a_i's are differentiable functions of $\underset{\sim}{x} \triangleq [x_1 \; x_2 \; \ldots \; x_n]^T$.

The minimax problem is formulated as a least pth objective due to Bandler and Charalambous [1] - [2]. An algorithm recently proposed by Charalambous [3] and the Fletcher minimization program [4] are then adapted to solve the resulting least pth optimization problem.

2. The Algorithm

(1) Set $r = 1$, $k = \beta$, where β is an integer.

(2) Define $\xi^1 = \min[\hat{\xi}^1, M_a(\underset{\sim}{x}^0)]$, where $\underset{\sim}{x}^0$ is the starting point and $\hat{\xi}^1$ is an initial estimate of ξ^1.

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

J.W. Bandler is with the Group on Simulation, Optimization and Control and Department of Electrical Engineering, McMaster University, Hamilton, Canada.

C. Charalambous is with the Department of Combinatorics and Optimization, University of Waterloo, Waterloo, Canada.

J.H.K. Chen was with the Group on Simulation, Optimization and Control, McMaster University. He is now with Bell-Northern Research, Ottawa, Canada.

If $k \leq 0$ set $I^1 = \{i | a_i(\underset{\sim}{x}^0) \geq \xi^0,\ i \in I\}$ otherwise set $I^1 = I$, where ξ^0 is a preset margin.

(3) Minimize with respect to $\underset{\sim}{x}$ the function

$$U_\xi(\underset{\sim}{x},\xi^r) = (M_\xi(\underset{\sim}{x},\xi^r)-\varepsilon)\left[\sum_{i\varepsilon J}\left(\frac{a_i(\underset{\sim}{x})-\xi^r-\varepsilon}{M_\xi(\underset{\sim}{x},\xi^r)-\varepsilon}\right)^q\right]^{\frac{1}{q}}$$

where

$$M_\xi(\underset{\sim}{x},\xi^r) = M_a(\underset{\sim}{x}) - \xi^r$$

$$\varepsilon = \begin{cases} 0 & \text{for } M_\xi(\underset{\sim}{x},\xi^r) \neq 0 \\ \text{small positive number for } M_\xi(\underset{\sim}{x},\xi^r) = 0 \end{cases}$$

$$q = p \operatorname{sgn} M_\xi(\underset{\sim}{x},\xi^r)$$

and

$$\text{if } M_\xi(\underset{\sim}{x},\xi^r) \begin{cases} > 0, \text{ then } 1 < p < \infty,\ J = \{i | a_i(\underset{\sim}{x}) \geq \xi^r,\ i\varepsilon I^r\} \\ \leq 0, \text{ then } 1 \leq p < \infty,\ J = I^r. \end{cases}$$

(4) If $r \geq r_{max}$, where r_{max} is the maximum permissible number of optimizations, stop.

(5) Set $\xi^{r+1} = \sum_{i\varepsilon\check{J}} u_i a_i(\check{\underset{\sim}{x}}^r)$

where

$$u_i = \frac{v_i}{\sum_{i\varepsilon\check{J}} v_i}$$

$$v_i = \begin{cases} \left(\dfrac{a_i(\check{\underset{\sim}{x}}^r)-\xi^r}{M_\xi(\check{\underset{\sim}{x}}^r,\ \xi^r)}\right)^{q-1} & \text{for } i\varepsilon\check{J} \\ 0 \quad \text{for } i \notin \check{J} \end{cases}$$

where $\check{J}$ is the set J corresponding to the rth optimum and $\check{x}^r$ is the optimum parameter vector of the rth optimization.

If $M_a(\check{x}^r) - \xi^{r+1} < \eta$, where η is a small positive number, stop.

(6) Set k = k - 1. If $k \leq 0$ set $I^{r+1} = \{ i | a_i(\check{x}^r) \geq \xi^r, i \in I\}$ otherwise set $I^{r+1} = I^r$.

(7) Set r = r + 1 and go to (3).

3. Comments

The algorithm recently proposed by Charalambous [3] differs from the two previous algorithms of Bandler and Charalambous [1]-[2] only in the method of determining the artificial margin ξ^r. After the first optimization, the value of ξ^r used in the new algorithm is, under appropriate conditions [3], a lower bound on $M_a(\check{x})$, where $\check{x}$ is the minimax optimum. Therefore, the index set for the least pth formulation is reduced to

$$J = \{i | a_i(x) > \xi^r, i \in I\}$$

and some computation effort may be saved.

In implementing the new algorithm, an option is introduced whereby the index set for the evaluation of the function for the (r+1)st optimization may be reduced to

$$I^{r+1} = \{i | a_i(\check{x}^r) \geq \xi^r, i \in I\}.$$

Therefore, in an approximation problem, say, the user can afford to start the optimization with a large number of sampling points in order to minimize the possibility of missing some crucial points.

After each optimization, the complete original set of functions will be evaluated to determine $K = \{i | a_i(\check{x}^r) = M_a(\check{x}^r), i \in I\}$. If $K \cap I^r = \emptyset$, the program will halt and output an error message.

The accuracy in the estimation of ξ^r, the lower bound, depends on the accuracy of the optimum, $\check{x}^{r-1}$, obtained. If ξ^r is exceeded by more than one percent in the rth optimization, the index set for the evaluation of the functions for the (r+1)st optimization will be reset to

$$I^{r+1} = I.$$

If the problem involves meeting certain performance specifications, the first optimization will indicate whether such specifications can be satisfied [1]. An option is provided whereby the optimization process can be halted if the specifications cannot be met.

The program has been written in such a way that the optimization can be restarted from any point instead of having to repeat the entire process.

A small value of p, such as 2, is recommended. If a large value of p is used, much effort will be spent in the initial optimizations which involve more functions.

All input data is entered through the argument of MINOPT, hence, the program can be easily incorporated into other automated computer-aided design packages.

MINOPT is written in standard FORTRAN IV and has a total of 512 cards.

4. The Argument List

CALL MINOPT (USER, N,NA,P,SI,SIO,NOM,MAX,EST,ETA,EPS,IGC,ISP,IFC,IP,X,Z,GU,A, PY,Y,H,W,GA,T1,T1P,B,VI,ID,IE)

The arguments are as follows

USER the identifier of the user subroutine-see Section 5.

N an integer set to the number of variables ($N \geq 2$).

NA an integer set to the number of functions.

P a real number set to the value of p used in the least pth formulation.

SI a single element in which the value of the current artificial margin is stored. SI should be set to an estimate of the initial value of the artificial margin or zero on entry.

SIO a single element in which the value of the previous artificial margin is stored. SIO should be set to zero or an estimate of the margin for the reduction of the number of functions on entry.

NOM an integer set to the maximum permissible number of optimizations.

MAX an integer set to the maximum permissible number of function evaluations.

EST a real number set to the estimated minimum value of the least pth objective.

ETA a real number set to the stopping test quantity for the algorithm.

EPS a real array of N elements set to the test quantities used in the Fletcher program. The value of the elements will be reduced by a factor of 10 after each optimization.

IGC an integer set to 1 if the derivatives at the starting point are to be checked by numerical perturbation. Otherwise, set to any other value.

ISP an integer set to i if the scheme for the reduction of the number of functions is to be applied after the ith optimization.

IFC an integer set to 1 if the optimization is to be terminated when the specifications cannot be satisfied.

IP an integer controlling output printing to be set as follows:

IP > 0, printing out every IP iterations

IP = 0, printing after each optimization

IP < 0, printing suppressed.

X a real array of N elements in which the current estimate of the solution is stored. An initial approximation must be set in X on entry.

Z a real array of NA elements set to the values of the independent variable at which the functions are to be evaluated.

GU a real array of N elements in which the derivatives of the least pth objective corresponding to X above will be returned.

A a real array of NA elements in which the values of the current set of functions minus the artificial margin is stored.

PY,Y arrays of N elements.

H an array of N(N+1)/2 elements.

W an array of 4N elements.

GA a two suffix array of N rows and NA columns.

T1,T1P,B,VI,ID,IE

arrays of NA elements.

5. The User Subroutine

The user must provide a subroutine headed

```
SUBROUTINE XXX(Z,A,NA,GA,X,N,ID,IG)
DIMENSION Z(1), A(1), GA(N,1), X(1), ID(1)
```

where XXX is an identifier chosen by the user.

This subroutine should use the variables $\mathbf{x}$ supplied in array X, the number of variables supplied in N, the values of the independent variable z supplied in array Z, the current index set for z supplied in array ID and the current number of functions supplied in NA to evaluate the functions and their corresponding partial derivatives and place them in arrays A and GA, respectively. XXX must be passed to MINOPT as MINOPT's first argument - see Section 4, and appear in an EXTERNAL statement in the program that calls MINOPT.

A zero value of the input parameter IG indicates that the partial derivatives are not required. Hence, IG may be used to bypass the evaluation of the partial derivatives.

6. Other Subroutines

The following is a brief description of the subroutines called by MINOPT.

LPOBJ formulates the least pth objective.

GDCHK checks the derivatives at the starting point by numerical perturbation.

OUTPUT outputs the optimum solution or the current estimate of the solution.

VA09A is the Fletcher minimization program.

The overall structure of MINOPT is shown in Figure 1.

7. Illustrative Example

Find a second-order model of a fourth-order system, when the input to the system is an impulse, in the minimax sense.

The transfer function of the fourth-order system is

$$G(s) = \frac{(s+4)}{(s+1)(s^2+4s+8)(s+5)}$$

and the transfer function of the second-order model is

$$H(s) = \frac{x_3}{(s+x_1)^2 + x_2^2}$$

The problem is therefore equivalent to finding the optimum point $\check{\underset{\sim}{x}}$ such that the function

$$F(\underset{\sim}{x},t) = \frac{x_3}{x_2} \exp(-x_1 t) \sin x_2 t$$

best approximates the function

$$S(t) = \frac{3}{20} \exp(-t) + \frac{1}{52} \exp(-5t) - \frac{\exp(-2t)}{65} (3 \sin 2t + 11 \cos 2t)$$

in the minimax sense.

The problem was discretized into 51 uniformly spaced points in the time interval 0 to 10 seconds and the function to be minimized is given by

$$U = \max_{i \in I} |e_i(\underset{\sim}{x})| \, , \; I = \{1,2,\ldots,51\}$$

where

$$e_i(\underset{\sim}{x}) = F(\underset{\sim}{x},t_i) - S(t_i).$$

The minimax optimum is

$$\check{U} = 0.794706 \times 10^{-2}$$

and

$$\check{\underset{\sim}{x}} = \begin{bmatrix} 0.684418 \\ 0.954093 \\ 0.122864 \end{bmatrix}$$

A typical calling program, user subroutine and printout of results are shown in Figures 2,3 and 4, respectively. Four optimizations and 119 function evaluations are required. Figures 5 and 6 illustrate the calling program and the corresponding printout of results when the same problem was restarted from the optimum of the third optimization.

Acknowledgement

The authors are indebted to W.Y. Chu for his assistance in this work.

References

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

[2] C. Charalambous and J.W. Bandler, "Nonlinear minimax optimization as a sequence of least pth optimization with finite values of p", McMaster University, Hamilton, Canada, Internal Report in Simulation, Optimization and Control, No. SOC-3, June 1973 (revised March 1974).

[3] C. Charalambous, "Minimax optimization of recursive digital filters using recent minimax results", _IEEE Trans. Acoust. Speech and Signal Processing_, vol. ASSP-23, pp. 333-345, Aug. 1975.

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

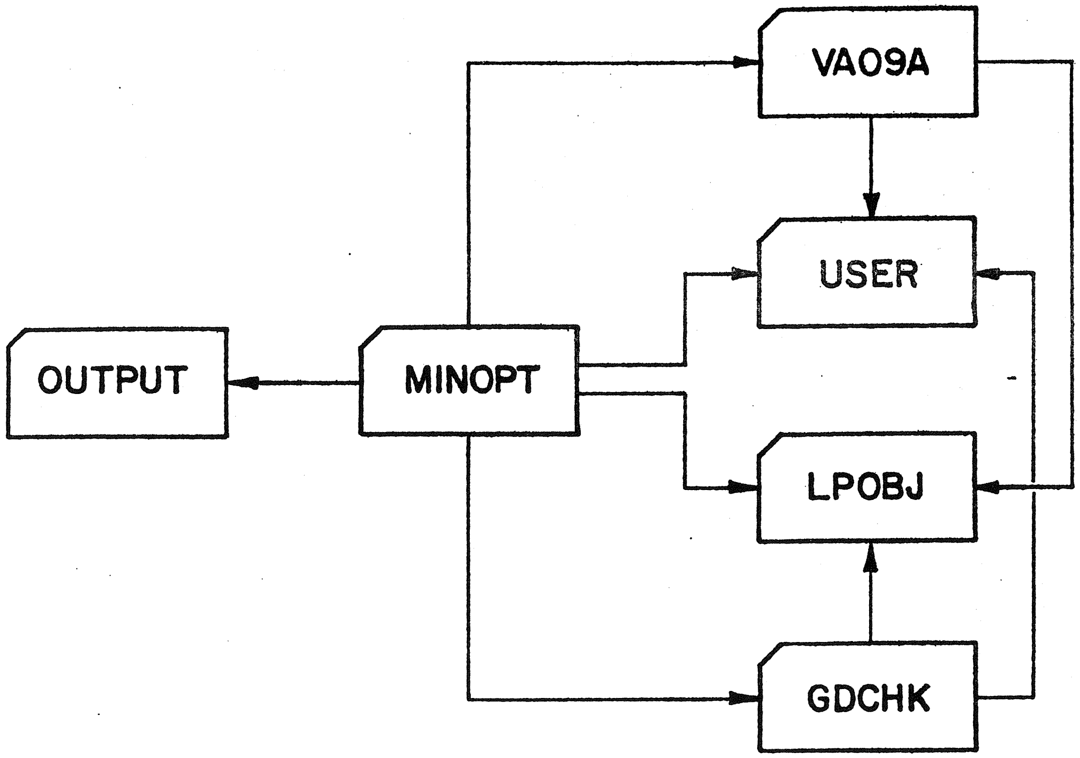

Figure 1. Overall structure of MINOPT.

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      DIMENSION X(3), GU(3), EPS(3), H(6), W(12), A(51), GA(3,51), T1(51
     1), T1P(51), B(51), VI(51), PY(3), Y(3), Z(51), ID(51), IF(51)
      EXTERNAL FCT
      COMMON /ABC/ SPE(51)
C
      N=3
      NA=51
      P=2.
      SI=4.E-3
      SIO=0.
      NOM=8
      MAX=300
      EST=0.
      ETA=1.E-6
      DO 1 I=1,N
      EPS(I)=1.E-5
    1 CONTINUE
      IGC=1
      ISP=1
      IFC=0
      IP=20
      X(1)=1.
      X(2)=1.
      X(3)=1.
      DO 2 I=1,NA
      Z(I)=0.2*FLOAT(I-1)
    2 CONTINUE
      D1=3./20.
      DO 3 I=1,NA
      T=Z(I)
      D2=EXP(-T)
      TT=T+T
      SPE(I)=D1*D2+D2**5/52.-D2*D2*(3.*SIN(TT)+11.*COS(TT))/65.
    3 CONTINUE
      CALL MINOPT (FCT,N,NA,P,SI,SIO,NOM,MAX,EST,ETA,EPS,IGC,ISP,IFC,IP,
     1X,Z,GU,A,PY,Y,H,W,GA,T1,T1P,B,VI,ID,IF)
      STOP
      END
```

Figure 2. Calling program for the system modelling example. Starting point $\underset{\sim}{x}^0 = [1\ 1\ 1]^T$.

```
      SUBROUTINE FCT (Z,A,NA,GA,X,N,ID,IG)
C
      DIMENSION Z(1), A(1), GA(N,1), X(1), ID(1)
      COMMON /ABC/ SPE(51)
C
      DO 1 I=1,NA
      J=ID(I)
      T=Z(J)
      D2=EXP(-T)
      D4=D2**X(1)/X(2)
      D7=X(2)*T
      D5=D4*SIN(D7)
      D6=D4*COS(D7)
      APE=X(3)*D5
      D3=APE-SPE(J)
      A(J)=ABS(D3)
      IF (IG.EQ.0) GO TO 1
      D8=D3/A(J)
      GA(1,J)=-APE*T*D8
      GA(2,J)=(-APE/X(2)+X(3)*T*D6)*D8
      GA(3,J)=D5*D8
    1 CONTINUE
      RETURN
      END
```

Figure 3. User subroutine for the system modelling example.

GRADIENTS CHECKING

GRADIENTS HAVE BEEN CHECKED AT THE FOLLOWING POINT

```
X( 1)=  1.00000000E+00
X( 2)=  1.00000000E+00
X( 3)=  1.00000000E+00
```

ANALYTICAL GRADIENTS	NUMERICAL GRADIENTS	PERCENTAGE ERROR
-7.78784935E-01	-7.78784933E-01	2.00411507E-07
-3.78029995E-01	-3.78029995E-01	1.72039197E-08
7.89847237E-01	7.89847238E-01	2.16616212E-07

GRADIENTS ARE O. K.

ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION = 4.00000000E-03

NUMBER OF FUNCTIONS FOR THE NEXT OPTIMIZATION = 51

OPTIMIZATION 1

ITER	FUNCT	OBJECTIVE	VARIABLE	GRADIENT
0	9	6.394211E-01	1.000000E+00	-7.787849E-01
			1.000000E+00	-3.780300E-01
			1.000000E+00	7.898472E-01
20	36	7.778212E-03	8.520020E-01	6.395533E-06
			8.935317E-01	1.384626E-05
			1.422568E-01	-8.362492E-05
22	38	7.778211E-03	8.520350E-01	-9.730915E-08
			8.935018E-01	-2.855661E-08
			1.422609E-01	6.064313E-07

IEXIT = 1

NORMAL EXIT

CURRENT MAXIMUM FUNCTION VALUE = 1.05144148E-02

ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION = 7.27711352E-03

NUMBER OF FUNCTIONS FOR THE NEXT OPTIMIZATION = 13

OPTIMIZATION 2

ITER	FUNCT	OBJECTIVE	VARIABLE	GRADIENT
35	55	1.161221E-03	7.051282E-01	-1.504854E-08
			9.479483E-01	-3.011574E-08
			1.251141E-01	1.234177E-08

IEXIT = 1

NORMAL EXIT

CURRENT MAXIMUM FUNCTION VALUE = 8.24480216E-03

ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION = 7.93591219E-03

NUMBER OF FUNCTIONS FOR THE NEXT OPTIMIZATION = 6

Figure 4. Results for the system modelling example. Starting point $\underset{\sim}{x}^0 = [1\ 1\ 1]^T$.

OPTIMIZATION 3

ITER	FUNCT	OBJECTIVE	VARIABLE	GRADIENT
40	68	5.436247E-05	6.876561E-01	-4.268539E-03
			9.525845E-01	-2.717345E-04
			1.231909E-01	1.732489E-01
49	81	1.915435E-05	6.847436E-01	-3.683231E-07
			9.540264E-01	1.349722E-07
			1.228994E-01	1.685067E-06

IEXIT = 1

NORMAL EXIT

CURRENT MAXIMUM FUNCTION VALUE = 7.95178792E-03

ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION = 7.94705799E-03

NUMBER OF FUNCTIONS FOR THE NEXT OPTIMIZATION = 4

OPTIMIZATION 4

ITER	FUNCT	OBJECTIVE	VARIABLE	GRADIENT
60	111	3.631441E-09	6.844180E-01	-1.622166E-02
			9.540929E-01	-1.916376E-02
			1.228643E-01	1.199815E-01
64	118	1.629539E-09	6.844178E-01	-1.428569E-02
			9.540931E-01	-3.729895E-03
			1.228642E-01	1.650471E-01

IEXIT = 1

NORMAL EXIT

~~CURRENT MAXIMUM FUNCTION VALUE = 7.94705954E-03~~

ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION = 7.94705910E-03

FOLLOWING IS THE OPTIMUM SOLUTION

OBJECTIVE FUNCTION U = 1.62953865E-09

X(1)= 6.84417768E-01 GU(1)= -1.42856936E-02
X(2)= 9.54093084E-01 GU(2)= -3.72989486E-03
X(3)= 1.22864249E-01 GU(3)= 1.65647054E-01

NUMBER OF FUNCTION EVALUATIONS = 119*

*This total includes the number of function evaluations required for gradient checking, minimization and the determination of the artificial margin and index set.

Figure 4. [continued]. Results for the system modelling example.

Starting point $\underset{\sim}{x}^0 = [1\ 1\ 1]^T$.

```
      PROGRAM MAIN(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
C
      DIMENSION X(3), GU(3), EPS(3), H(6), W(12), A(51), GA(3,51), T1(51
     1), T1P(51), B(51), VI(51), PY(3), Y(3), Z(51), ID(51), IE(51)
      EXTERNAL FCT
      COMMON /ABC/ SPE(51)
C
      N=3
      NA=51
      P=2.
       SI=7.94705801E-03
       SIO=7.93581201E-03
      NOM=8
      MAX=300
      EST=0.
      ETA=1.E-6
      DO 1 I=1,N
       EPS(I)=1.E-08
    1 CONTINUE
       IGC=0
       ISP=0
      IEC=0
      IP=20
       X(1)=6.847436E-01
       X(2)=9.540264E-01
       X(3)=1.228994E-01
      DO 2 I=1,NA
      Z(I)=0.2*FLOAT(I-1)
    2 CONTINUE
      D1=3./20.
      DO 3 I=1,NA
      T=Z(I)
      D2=EXP(-T)
      TT=T+T
      SPE(I)=D1*D2+D2**5/52.-D2*D2*(3.*SIN(TT)+11.*COS(TT))/65.
    3 CONTINUE
      CALL MINOPT (FCT,N,NA,P,SI,SIO,NOM,MAX,EST,ETA,EPS,IGC,ISP,IEC,IP,
     1X,Z,GU,A,PY,Y,H,W,GA,T1,T1P,B,VI,ID,IE)
      STOP
      END
```

Figure 5. Calling program for the system modelling example.

Starting point $\underset{\sim}{x}^0 = [0.6847436, 0.9540264, 0.1228994]^T$

```
ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION =  7.94705801E-03
NUMBER OF FUNCTIONS FOR THE NEXT OPTIMIZATION =     4

OPTIMIZATION  1
---------------
ITER   FUNCT          OBJECTIVE        VARIABLE         GRADIENT
  0      2          4.725031E-06    6.847436E-01    -4.260840E-03
                                    9.540264E-01    -2.716570E-04
                                    1.228994E-01     1.733467E-01

 14     42          1.541955E-09    6.844178E-01     4.844517E-03
                                    9.540931E-01    -3.682037E-03
                                    1.228642E-01    -8.827802E-02

IEXIT =     1
NORMAL EXIT
CURRENT MAXIMUM FUNCTION VALUE =  7.94705906E-03
ARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION =  7.94705888E-03


          FOLLOWING IS THE OPTIMUM SOLUTION
          ---------------------------------
             OBJECTIVE FUNCTION U =  1.54195549E-09

        X( 1)=  6.84417759E-01 GU( 1)=  4.84451665E-03
        X( 2)=  9.54093077E-01 GU( 2)= -3.68203700E-03
        X( 3)=  1.22864246E-01 GU( 3)= -8.82780232E-02
      NUMBER OF FUNCTION EVALUATIONS =   43
```

Figure 6. Results for the system modelling example.

Starting point $\underset{\sim}{x}^0 = [0.6847436 \quad 0.9540264 \quad 0.1228994]^T$.

FORTRAN Listing for MINOPT

```
      SUBROUTINE MINOPT (USER,N,NA,P,SI,SIO,NOM,MAX,EST,ETA,EPS,IGC,ISP,
     1IFC,IP,X,Z,GU,A,PY,Y,H,W,GA,T1,T1P,B,VI,ID,IF)
C
C     MINOPT PACKAGE FORMULATES A MINIMAX PROBLEM AS A LEAST PTH
C     OBJECTIVE DUE TO BANDLER AND CHARALAMBOUS(1)-(2) WHICH IS THEN
C     SOLVED BY AN ALGORITHM PROPOSED BY CHARALAMBOUS(3) IN CONJUNCTION
C     WITH THE FLETCHER MINIMIZATION PROGRAM(4).
C     REFERENCES
C     (1) C. CHARALAMBOUS AND J.W. BANDLER, NEW ALGORITHMS FOR NETWORK
C         OPTIMIZATION, IEEE TRANS. MICROWAVE THEORY TECH., VOL.MTT-21,
C         PP.815-818, DEC. 1973.
C     (2) C. CHARALAMBOUS AND J.W. BANDLER, NONLINEAR MINIMAX OPTIMIZA-
C         TION AS A SEQUENCE OF LEAST PTH OPTIMIZATION WITH FINITE
C         VALUES OF P, MCMASTER UNIVERSITY, HAMILTON, CANADA, INTERNAL
C         REPORT IN SIMULATION, OPTIMIZATION AND CONTROL, NO.SOC-3. JUNE
C         1973.
C     (3) C. CHARALAMBOUS, MINIMAX OPTIMIZATION OF RECURSIVE DIGITAL
C         FILTERS USING RECENT MINIMAX RESULTS, UNIVERSITY OF WATERLOO,
C         WATERLOO, CANADA, DEPARTMENT OF COMBINATORICS AND OPTIMIZATION
C         , RESEARCH REPORT CORR 74-6, FEBRUARY 1974.
C     (4) R. FLETCHER, FORTRAN SUBROUTINES FOR MINIMIZATION BY QUASI-
C         NEWTON METHODS, ATOMIC ENERGY RESEARCH ESTABLISHMENT, HARWELL,
C         BERKSHIRE, ENGLAND, REPORT AERE-R7125, 1972.
C
      DIMENSION X(1), GU(1), EPS(1), H(1), W(1), A(1), GA(N,1), T1(1), T
     11P(1), B(1), VI(1), PY(1), Y(1), Z(1), ID(1), IF(1)
      EXTERNAL USER
C
      IFN=0
      NAO=NA
      DO 1 I=1,NA
      IF(I)=I
      ID(I)=I
    1 CONTINUE
      NO=1
      L=1
      IF (IGC.EQ.1) CALL GDCHK (USER,Z,A,NA,GA,X,N,T1,T1P,B,GU,PY,Y,IP,S
     1I,IFN,IF)
      IF (IP.GE.0) WRITE (6,28)
      CALL USER (Z,A,NA,GA,X,N,IF,0)
      IFN=IFN+1
      AM=A(1)
      DO 2 I=1,NA
      AM=AMAX1(AM,A(I))
    2 CONTINUE
      SI=AMIN1(SI,AM)
      IF (IP.GE.0) WRITE (6,25) SI
      IF (ISP.GT.0) GO TO 4
      AM=AM-SIO
      DO 3 I=1,NAO
      B(I)=A(I)-SIO
    3 CONTINUE
      GO TO 13
    4 IF (IP.GE.0) WRITE (6,24) NA
      ESD=EST
      IF (IP.GE.0) WRITE (6,26) NO
      CALL VA09A (USER,N,X,ESD,GU,H,W,0.,EPS,L,MAX,IP,IEX,Z,A,GA,T1,T1P,
     1NO,IFN,SI,AM,P,NA,ID)
```

```
      L=3
      IF (IFX.NE.1) GO TO 22
      ISP=ISP-1
      CALL USER (Z,A,NAO,GA,X,N,IE,0)
      IF (ISP.GE.0) GO TO 6
      AMN=A(1)
      B(1)=A(1)-SI
      DO 5 I=1,NAO
      AMN=AMAX1(AMN,A(I))
      B(I)=A(I)-SI
 5    CONTINUE
 6    CALL LPOBJ (N,A,NA,GA,GU,U,T1,T1P,P,SI,AM,ID,0)
      IFN=IFN+1
      AMS=AM+SI
      IF (IP.GE.0) WRITE (6,27) AMS
      IF (ISP.GE.0) GO TO 7
      IF (ABS((AMN-AMS)/AMN).LE.0.001) GO TO 7
      KO=2
      GO TO 23
 7    IF (IFC.EQ.1.AND.NO.EQ.1.AND.AMS.GT.0.) GO TO 21
      NO=NO+1
      IF (NO.GT.NOM) GO TO 20
      K=0
      SV=0.
      DO 9 I=1,NA
      J=ID(I)
      IF (AM.LT.0.) GO TO 8
      IF (A(J).LE.0.) GO TO 9
 8    K=K+1
      ID(K)=J
      VI(K)=T1P(J)/T1(J)
      SV=SV+VI(K)
 9    CONTINUE
      SIO=SI
      SI=0.
      DO 10 I=1,K
      VI(I)=VI(I)/SV
      J=ID(I)
      SI=SI+VI(I)*(A(J)+SIO)
 10   CONTINUE
      DO 11 I=1,N
      EPS(I)=EPS(I)*0.1
 11   CONTINUE
      IF (IP.GE.0) WRITE (6,25) SI
      IF ((AM+SIO-SI).LE.ETA) GO TO 19
      IF (ISP.GT.0) GO TO 14
      IF (ISP.LT.0) GO TO 13
      DO 12 I=1,NAO
      B(I)=A(I)
 12   CONTINUE
 13   IF (AM.GT.0.) GO TO 16
      AT=-ABS(SIO*0.01)
      IF (AM.GE.AT) GO TO 17
 14   DO 15 I=1,NAO
      ID(I)=I
 15   CONTINUE
      NA=NAO
      GO TO 4
```

```
16    AT=0.
17    J=0
      DO 18 I=1,NAO
      IF (B(I).LE.AT) GO TO 18
      J=J+1
      ID(J)=I
18    CONTINUE
      NA=J
      GO TO 4
19    KO=1
      GO TO 23
20    KO=3
      GO TO 23
21    KO=4
      GO TO 23
22    KO=0
23    IF (IP.GE.0) CALL OUTPUT (N,X,U,GU,KO,IFN)
      RETURN
C
C
C
24    FORMAT (1H0,47HNUMBER OF FUNCTIONS FOR THE NEXT OPTIMIZATION =,I5)
25    FORMAT (1H0,45HARTIFICIAL MARGIN FOR THE NEXT OPTIMIZATION =,F16.8
     1)
26    FORMAT (1H0,12HOPTIMIZATION,I3,/,16H ----------------,/,5H ITER,3X,
     15HFUNCT,8X,9HOBJECTIVE,6X,8HVARIABLE,7X,8HGRADIENT,/)
27    FORMAT (1H0,32HCURRENT MAXIMUM FUNCTION VALUE =,F16.8)
28    FORMAT (1H1)
      END
```

```
      SUBROUTINE LPOBJ (N,A,NA,GA,GU,U,T1,T1P,P,SI,AM,ID,IG)
C
C     THIS SUBROUTINE FORMULATES THE LEAST PTH OBJECTIVE.
C
      DIMENSION A(1), GA(N,1), GU(1), T1(1), T1P(1), ID(1)
C
      J=ID(1)
      A(J)=A(J)-SI
      AM=A(J)
      DO 1 I=2,NA
      J=ID(I)
      A(J)=A(J)-SI
      AM=AMAX1(AM,A(J))
    1 CONTINUE
      IF (AM.NE.0.) GO TO 3
      DO 2 I=1,NA
      J=ID(I)
      A(J)=A(J)-1.E-10
    2 CONTINUE
      AM=AM-1.E-10
    3 PP=SIGN(P,AM)
      S1=0.
      DO 5 I=1,NA
      J=ID(I)
      IF (AM.LT.0.) GO TO 4
      IF (A(J).LE.0.) GO TO 5
    4 T1(J)=A(J)/AM
      T1P(J)=T1(J)**PP
      S1=S1+T1P(J)
    5 CONTINUE
      S3=S1**(1./PP)
      U=AM*S3
      IF (IG.EQ.0) RETURN
      DO 8 I=1,N
      S2=0.
      DO 7 J=1,NA
      K=ID(J)
      IF (AM.LT.0.) GO TO 6
      IF (A(K).LE.0.) GO TO 7
    6 S2=S2+T1P(K)/T1(K)*GA(I,K)
    7 CONTINUE
      GU(I)=S3/S1*S2
    8 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE GRCHK (USER,SP,A,NA,GA,X,N,T1,T1P,P,G,PY,Y,IP,SI,IEN,ID
     1)
C
C     THIS SUBROUTINE CHECKS THE DERIVATIVES BY NUMERICAL PERTURBATION.
C
      DIMENSION SP(1), A(1), GA(N,1), X(1), T1(1), T1P(1), G(1), PY(1),
     1Y(1), ID(1)
C
      CALL USER (SP,A,NA,GA,X,N,ID,1)
      CALL LPOBJ (N,A,NA,GA,G,U,T1,T1P,P,SI,AM,ID,1)
      DO 1 I=1,N
      Z=X(I)
      DELX=1.E-6*X(I)
      IF (ABS(X(I)).LT.1.E-10) DELX=1.E-10
      X(I)=Z+DELX
      CALL USER (SP,A,NA,GA,X,N,ID,0)
      CALL LPOBJ (N,A,NA,GA,PY,F2,T1,T1P,P,SI,AM,ID,0)
      X(I)=Z-DELX
      CALL USER (SP,A,NA,GA,X,N,ID,0)
      CALL LPOBJ (N,A,NA,GA,PY,F1,T1,T1P,P,SI,AM,ID,0)
      Y(I)=0.5*(F2-F1)/DELX
      X(I)=Z
    1 CONTINUE
      IEN=IEN+1+N+N
      DO 2 I=1,N
      IF (ABS(Y(I)).LT.1.E-20) Y(I)=1.E-20
      IF (ABS(G(I)).LT.1.E-20) G(I)=1.E-20
      PY(I)=ABS((Y(I)-G(I))/Y(I))*100.
    2 CONTINUE
      IF (IP.LT.0) GO TO 3
      WRITE (6,6)
      WRITE (6,7)
      WRITE (6,8) (I,X(I),I=1,N)
      WRITE (6,9)
      WRITE (6,10) (G(I),Y(I),PY(I),I=1,N)
    3 DO 4 I=1,N
      IF (PY(I).GT.10.) GO TO 5
    4 CONTINUE
      IF (IP.GE.0) WRITE (6,11)
      RETURN
    5 WRITE (6,12)
      CALL EXIT
C
C
C
    6 FORMAT (1H1)
    7 FORMAT (6X,18HGRADIENTS CHECKING,/,6X,18H------------------,//,6X,
     150HGRADIENTS HAVE BEEN CHECKED AT THE FOLLOWING POINT,/)
    8 FORMAT (10X,2HX(,I2,2H)=,F16.8)
    9 FORMAT (///,6X,20HANALYTICAL GRADIENTS,5X,19HNUMERICAL GRADIENTS,7
     1X,16HPERCENTAGE ERROR,/)
   10 FORMAT (6X,F16.8,9X,F16.8,9X,F16.8)
   11 FORMAT (//,6X,19HGRADIENTS ARE O. K.)
   12 FORMAT (//,6X,64HYOUR PROGRAM HAS BEEN TERMINATED BECAUSE GRADIENT
     1S ARE INCORRECT,/,6X,21HPLEASE CHECK IT AGAIN)
      END
```

```
      SUBROUTINE OUTPUT (N,X,U,GU,KO,IEN)
C
C     THIS SUBROUTINES OUTPUTS THE OPTIMUM SOLUTION OR THE CURRENT
C     ESTIMATE.
C
      DIMENSION X(1), GU(1)
C
      IF (KO.EQ.0.OR.KO.EQ.2.OR.KO.EQ.3.OR.KO.EQ.4) WRITE (6,2)
      IF (KO.EQ.1) WRITE (6,3)
      WRITE (6,4) U
      DO 1 I=1,N
      WRITE (6,5) I,X(I),I,GU(I)
    1 CONTINUE
      WRITE (6,6) IEN
      IF (KO.EQ.2) WRITE (6,7)
      IF (KO.EQ.3) WRITE (6,8)
      IF (KO.EQ.4) WRITE (6,9)
      RETURN
C
C
C
    2 FORMAT (1H1,15X,25HRESULTS AT LAST ITERATION,/,16X,25H------------
     1-------------)
    3 FORMAT (1H1,11X,33HFOLLOWING IS THE OPTIMUM SOLUTION,/,12X,33H----
     1-----------------------------)
    4 FORMAT (1H0,15X,22HOBJECTIVE FUNCTION U =,E16.8,/)
    5 FORMAT (8X,2HX(,I2,2H)=,E16.8,1X,3HGU(,I2,2H)=,E16.8)
    6 FORMAT (1H0,5X,32HNUMBER OF FUNCTION EVALUATIONS =,I5)
    7 FORMAT (1H0,5X,49HSCHEME FOR REDUCTION OF NUMBER OF FUNCTIONS FAIL
     1S)
    8 FORMAT (1H0,3X,52HMAXIMUM PERMISSIBLE NUMBER OF OPTIMIZATIONS EXCE
     1EDED)
    9 FORMAT (1H0,5X,49HNO SOLUTION POSSIBLE FOR THE GIVEN SPECIFICATION
     1S)
      END
```

```
      SUBROUTINE VA09A (USER,N,X,F,G,H,W,DFN,EPS,MODE,MAXFN,IPRINT,IEXIT
     1,SP,A,GA,T1,T1P,NO,IFN,SI,AM,P,NA,ID)
C
C     THIS SUBROUTINE MINIMIZES A FUNCTION BY QUASI-NEWTON METHOD.
C
      REAL X(1),G(1),H(1),W(1),EPS(1),SP(1),A(1),GA(N,1),T1(1),T1P(1),ID
     1(1)
C
      IF (NO.NE.1) GO TO 1
      ITN=0
    1 CONTINUE
      NP=N+1
      N1=N-1
      NN=N*NP/2
      IS=N
      IU=N
      IV=N+N
      IB=IV+N
      IEXIT=0
      IF (MODE.EQ.3) GO TO 7
      IF (MODE.EQ.2) GO TO 4
      IJ=NN+1
      DO 3 I=1,N
      DO 2 J=1,I
      IJ=IJ-1
      H(IJ)=0.
    2 CONTINUE
      H(IJ)=1.
    3 CONTINUE
      GO TO 7
    4 CONTINUE
      IJ=1
      DO 6 I=2,N
      Z=H(IJ)
      IF (Z.LE.0.) RETURN
      IJ=IJ+1
      I1=IJ
      DO 6 J=I,N
      ZZ=H(IJ)
      H(IJ)=H(IJ)/Z
      JK=IJ
      IK=I1
      DO 5 K=I,J
      JK=JK+NP-K
      H(JK)=H(JK)-H(IK)*ZZ
      IK=IK+1
    5 CONTINUE
    6 IJ=IJ+1
      IF (H(IJ).LE.0.) RETURN
    7 CONTINUE
      IJ=NP
      DMIN=H(1)
      DO 8 I=2,N
      IF (H(IJ).GE.DMIN) GO TO 8
      DMIN=H(IJ)
    8 IJ=IJ+NP-I
      IF (DMIN.LE.0.) RETURN
      Z=F
      CALL USER (SP,A,NA,GA,X,N,ID,1)
```

```
      CALL LPROBJ (N,A,NA,GA,G,F,T1,T1P,P,SI,AM,ID,1)
      IFN=IFN+1
      DF=DFN
      IF (DFN.EQ.0.) DF=F-Z
      IF (DFN.LT.0.) DF=ABS(DF*F)
      IF (DF.LE.0.) DF=1.
 9    CONTINUE
      IF (IPRINT.LE.0) GO TO 10
      IF (MOD(ITN,IPRINT).NE.0) GO TO 10
      PRINT 38, ITN,IFN,F,((X(I),G(I)),I=1,N)
 10   CONTINUE
      ITN=ITN+1
      W(1)=-G(1)
      DO 12 I=2,N
      IJ=I
      I1=I-1
      Z=-G(I)
      DO 11 J=1,I1
      Z=Z-H(IJ)*W(J)
      IJ=IJ+N-J
 11   CONTINUE
      W(I)=Z
 12   CONTINUE
      W(IS+N)=W(N)/H(NN)
      IJ=NN
      DO 14 I=1,N1
      IJ=IJ-1
      Z=0.
      DO 13 J=1,I
      Z=Z+H(IJ)*W(IS+NP-J)
      IJ=IJ-1
 13   CONTINUE
      W(IS+N-I)=W(N-I)/H(IJ)-Z
 14   CONTINUE
      GS=0.
      DO 15 I=1,N
      GS=GS+W(IS+I)*G(I)
 15   CONTINUE
      IEXIT=2
      IF (GS.GE.0.) GO TO 37
      GS0=GS
      ALPHA=-2.*DF/GS
      IF (ALPHA.GT.1.) ALPHA=1.
      DF=F
      TOT=0.
 16   CONTINUE
      IEXIT=3
      IF (IFN.EQ.MAXFN) GO TO 37
      ICON=0
      IEXIT=1
      DO 17 I=1,N
      Z=ALPHA*W(IS+I)
      IF (ABS(Z).GE.EPS(I)) ICON=1
      X(I)=X(I)+Z
 17   CONTINUE
      CALL USER (SP,A,NA,GA,X,N,ID,1)
      CALL LPROBJ (N,A,NA,GA,W,FY,T1,T1P,P,SI,AM,ID,1)
      IFN=IFN+1
```

```
      GYS=0.
      DO 18 I=1,N
      GYS=GYS+W(I)*W(IS+I)
   18 CONTINUE
      IF (FY.GE.F) GO TO 19
      IF (ABS(GYS/GS0).LE..9) GO TO 21
      IF (GYS.GT.0.) GO TO 19
      TOT=TOT+ALPHA
      Z=10.
      IF (GS.LT.GYS) Z=GYS/(GS-GYS)
      IF (Z.GT.10.) Z=10.
      ALPHA=ALPHA*Z
      F=FY
      GS=GYS
      GO TO 16
   19 CONTINUE
      DO 20 I=1,N
      X(I)=X(I)-ALPHA*W(IS+I)
   20 CONTINUE
      IF (ICON.EQ.0) GO TO 37
      Z=3.*(F-FY)/ALPHA+GYS+GS
      ZZ=SQRT(Z**2-GS*GYS)
      Z=1.-(GYS+ZZ-Z)/(2.*ZZ+GYS-GS)
      ALPHA=ALPHA*Z
      GO TO 16
   21 CONTINUE
      ALPHA=TOT+ALPHA
      F=FY
      IF (ICON.EQ.0) GO TO 35
      DF=DF-F
      DGS=GYS-GS0
      LINK=1
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 23
      DO 22 I=1,N
      W(IU+I)=W(I)-G(I)
   22 CONTINUE
      SIG=1./(ALPHA*DGS)
      GO TO 30
   23 CONTINUE
      ZZ=ALPHA/(DGS-ALPHA*GS0)
      Z=DGS*ZZ-1.
      DO 24 I=1,N
      W(IU+I)=Z*G(I)+W(I)
   24 CONTINUE
      SIG=1./(ZZ*DGS**2)
      GO TO 30
   25 CONTINUE
      LINK=2
      DO 26 I=1,N
      W(IU+I)=G(I)
   26 CONTINUE
      IF (DGS+ALPHA*GS0.GT.0.) GO TO 27
      SIG=1./GS0
      GO TO 30
   27 CONTINUE
      SIG=-ZZ
      GO TO 30
   28 CONTINUE
```

```
      DO 29 I=1,N
      G(I)=W(I)
   29 CONTINUE
      GO TO 9
   30 CONTINUE
      W(IV+1)=W(IU+1)
      DO 32 I=2,N
      IJ=I
      I1=I-1
      Z=W(IU+I)
      DO 31 J=1,I1
      Z=Z-H(IJ)*W(IV+J)
      IJ=IJ+N-J
   31 CONTINUE
      W(IV+I)=Z
   32 CONTINUE
      IJ=1
      DO 33 I=1,N
      Z=H(IJ)+SIG*W(IV+I)**2
      IF (Z.LE.0.) Z=DMIN
      IF (Z.LT.DMIN) DMIN=Z
      H(IJ)=Z
      W(IB+I)=W(IV+I)*SIG/Z
      SIG=SIG-W(IB+I)**2*Z
      IJ=IJ+NP-I
   33 CONTINUE
      IJ=1
      DO 34 I=1,N1
      IJ=IJ+1
      I1=I+1
      DO 34 J=I1,N
      W(IU+J)=W(IU+J)-H(IJ)*W(IV+I)
      H(IJ)=H(IJ)+W(IB+I)*W(IU+J)
   34 IJ=IJ+1
      GO TO (35,28), LINK
   35 CONTINUE
      DO 36 I=1,N
      G(I)=W(I)
   36 CONTINUE
   37 CONTINUE
      IF (IPRINT.LT.0) RETURN
      PRINT 38, ITN,IFN,F,((X(I),G(I)),I=1,N)
      PRINT 39, IEXIT
      IF (IEXIT.EQ.1) PRINT 40
      IF (IEXIT.EQ.2) PRINT 41
      IF (IEXIT.EQ.3) PRINT 42
      RETURN
C
C
C
   38 FORMAT (1H ,I4,3X,I4,6X,F14.6,1X,80(F14.6,1X,F14.6,/,33X))
   39 FORMAT (1H0,7HIEXIT =,I5)
   40 FORMAT (1H0,11HNORMAL EXIT)
   41 FORMAT (1H0,29HEPS IS PROBABLY SET TOO SMALL)
   42 FORMAT (1H0,51HPERMISSIBLE NUMBER OF FUNCTION EVALUATIONS EXCEEDED
     1)
      END
```