No. SOC-105

# TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING

PART I - USER'S GUIDE

J.W. Bandler, J.H.K. Chen, P. Dalsgaard and P.C. Liu

September 1975

FACULTY OF ENGINEERING
McMASTER UNIVERSITY
HAMILTON, ONTARIO, CANADA

# TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING

J.W. Bandler, J.H.K. Chen, P. Dalsgaard and P.C. Liu

Abstract This report describes the development, organization and implementation of a user-oriented computer program package called TOLOPT (TOLerance OPTimization), which can solve continuous and/or discrete worst-case tolerance assignment problems. Worst-case vertices can be automatically selected and optimization will lead to the most favorable nominal design simultaneously with the largest possible tolerances on specified toleranced components. The program contains recent techniques and algorithms for nonlinear programming. The optimization is carried out by subprograms substantially the same as ones in the DISOPT package. The full Fortran IV listing is included in this report as well as three circuit examples illustrating the use of and typical printouts from TOLOPT.

---

This work was supported by the National Research Council of Canada under Grant A7239 and by the Danish Council for Scientific and Industrial Research through support to P. Dalsgaard.

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

J.H.K. Chen is with Bell-Northern Research, Ottawa, Canada.

P. Dalsgaard is with the Institute of Electronic Systems, Aalborg University, Aalborg, Denmark.

P.C. Liu is with Bell-Northern Research, Verdun, P.Q., Canada.

# I. INTRODUCTION

TOLOPT is a package of subroutines which can solve continuous and discrete worst-case tolerance assignment problems simultaneously with the selection of the most favorable nominal design [1-3]. The package is designed to handle the objective function, performance specifications, and parameter constraints in a unified manner such that any of the nominal values or tolerances (relative or absolute) can be fixed or varied automatically at the user's discretion. Time-saving techniques for choosing constraints (vertices selection) are incorporated. The routine involved also checks assumptions and performs worst-case analyses.

The continuous and (optional) discrete optimization methods are programmed in such a way that they may be used as a separate unit. This part, called DISOP2 and incorporating several optional features, is an updated version of DISOPT, which has been successfully applied in many different areas [3 - 6]. Dakin's tree search for discrete problems [7], efficient gradient minimization of functions of many variables by a recent quasi-Newton method [8] and recent developments in least pth approximation by Bandler and Charalambous [9 - 12] are employed. Extrapolation is also featured [13]. (Another practical problem which is analogous to the tolerance assignment problem is to determine the optimum component values to a certain number of significant figures, which can be done with DISOP2.)

The TOLOPT program is organized in such a way that future additions and deletions of performance specifications and constraints, replacement of cost functions and optimization methods are readily realized. Any of the different vertices elimination schemes can be bypassed or replaced by the user. It is felt that the program is particularly flexible in the way that the user may enter at any stage of the problem's solution. The user supplies the network analysis subroutines. With an arbitrary initial acceptable or unacceptable design as a starting point, the program will print out the set of nominal component parameters together with a set of optimal tolerances satisfying all the specifications in the worst-case sense. The user decides on a continuous solution and/or discrete solutions.

The package is written in Fortran IV. Several test examples are presented here to illustrate the approach. Typically, 64000 octal words of memory are required on a CDC 6400 computer.

## II. FEATURES OF TOLOPT

### The Overall Structure of TOLOPT

Fig. 1 displays a block diagram of the principal subprograms comprising the tolerance optimization program. A brief description of these subprograms is given in this section. A flow diagram is shown in Fig. 2.

TOLOPT (TOLerance OPTimization program) is the subroutine called by the user. It organizes input data and coordinates other subprograms. Subroutine DISOP2 is a general program for continuous and discrete nonlinear programming problems. Subroutine VERTST eliminates the inactive vertices of the tolerance region. Subroutine CONSTR sets up the constraint functions based on the response specifications, component bounds and other constraints supplied in the user subroutine USERCN. Subroutine COSTFN computes the cost function. The user has the option of supplying his own subroutine to define other cost functions. The user supplied subroutine NETWRK returns the network responses and the partial derivatives. In the user supplied subroutine USERCN the user has to define whatever extra constraints he needs and the corresponding partial derivatives. It should be noted that the constraints given in USERCN are not checked against the worst-case vertices. Table I is a summary of the features and options currently incorporated in TOLOPT. Some tolerances and nominal parameter values may be fixed and, hence, do not enter into the set of optimization parameters. The user supplies the initial values of the tolerances (relative or absolute) and the nominals with an appropriate vector to indicate whether they are fixed or variable, relative or absolute. The program will assign those variable components to the vector of optimization parameters. As initial values of the tolerances (relative or absolute) we recommend using small numbers, say 0.01 (relative) or whatever absolute values correspond to this.

## The Objective Function

The objective function we have investigated and implemented [1 - 3] is the weighted summation of the inverses of the relative or the absolute tolerances. The weighting factors may (as default values) be taken as one, but the user can, by using the default parameter ND2, specify his own set of weighting factors.

## Vertices Selection Schemes and Constraints

Various schemes have been developed to identify or to predict the most critical vertices that are likely to give rise to active constraints. Our proposed schemes will eliminate all but one vertex for each constraint function in the most favourable conditions. When monotonicity assumptions [2, 14] are not sufficient to describe the function behaviour, our scheme will increase the number of vertices until, at worst, all the 2**KD vertices are included.

Two major schemes of increasing complexity are programmed in the subroutine VERTST. Scheme 2 involves vertices $\phi^a = \phi^0 - \varepsilon_j u_j$ and $\phi^b = \phi^0 + \varepsilon_j u_j$. $u_j$ is the jth unit vector and $j \in I_\phi$, where $I_\phi \triangleq \{1,2,\ldots,k\}$ is the index set for the network components. Scheme 3 involves all vertices. Also, the special case (scheme 1) which occurs for $\phi^a = \phi^b$, has been programmed. In this case only one vertex is considered for each sample point.

The user decides on which vertices selection scheme (parameter ISCEME) he wants to use as well as the maximum number of allowable calls for the scheme selected for the updating procedure (parameter ND1). He may, if he wishes, bypass the whole subroutine (parameter IUPD) by supplying his own vertices or set up his own strategy of vertices selection. Furthermore, the user decides on the maximum number of vertices allowable at each sample point (parameter MAXVN). If more than the maximum allowable numbers are detected, the subroutine selects the ones corresponding to the lowest constraint value arranged in ascending order.

Printing out the detected vertices (parameter IWORST) and the value of the corresponding constraints, the user has the possibility of eliminating further vertices by considering the relative magnitude of the constraints.

As an option, if the parameter IUPD is given any value other than 0,1 and 2, the TOLOPT program can be used for vertices detection only. The program will print out the detected vertices and the value of the corresponding constraints such that the user has the possibility manually to eliminate vertices using his own judgement. The user has the possibility of supplying his own set of active vertices in two different ways. This will be illustrated in one of the examples.

The vertices selection schemes used are based on local information, therefore, the vertices should be updated at suitable intervals (see later).

The user supplies 3 sets of numbers, the elements of which correspond to the controlling parameter $\psi_i$, the specification $S_i$ and the weighting factor $w_i$. $\psi_i$ is an independent parameter, e.g., frequency, or any number to identify a particular function. $w_i$ is given by

$$w_i = \begin{cases} +1 \text{ if } S_i \text{ is an upper specification} \\ -1 \text{ if } S_i \text{ is a lower specification.} \end{cases}$$

If both upper and lower specifications are assigned to one point, the user can treat it as two points, one with an upper specification and the other with a lower specification. The theory presented in [3] will apply in this case under the monotonicity restrictions.

The vertices selection scheme will, for each i select a set of appropriate $\underset{\sim}{\mu}$. Corresponding to each $\underset{\sim}{\mu}$, the values $\psi_i$, $S_i$ and $w_i$ are stored. This information is used for forming the constraint functions.

The constraints associated with response specifications are of the form

$$g = w(S - F) \geq 0$$

with appropriate subscripts, where F is the circuit response function of $\underset{\sim}{\phi}$ and $\psi$, and w and S are as before.

The parameter constraints are

$$\phi_j^0 - \varepsilon_j - \phi_{\ell j} \geq 0$$

and

$$\phi_{uj} - \phi_j^0 - \varepsilon_j \geq 0$$

where $\phi_{uj}$ and $\phi_{\ell j}$, $j \in I_\phi$, are the user supplied upper and lower bounds.

## Updating Procedure

Before using the automated vertex selection an initial feasibility check is performed to check the feasibility of the nominal design. The outcome from this feasibility check is used as a starting point in the tolerance assignment problem. If a feasible nominal point is not attainable, the user has to relax some specifications or change his design.

The different optimization methods [9 - 13] are summarized in Table II. Once the constraints have been selected, optimization is started with a small value of p and $\alpha$ ($p = \alpha = 10$ as default values). See [3, 4] for definitions of these parameters. The routine for updating constraints is called whenever the $\alpha$ value is updated and/or each time new constraints have been added. For updating the values, we add new values of $\underset{\sim}{\mu}$ to the existing ones without any eliminations. This may not be the most efficient way but will be stable. When the maximum number of calls is exceeded or when there is no change of values for consecutive calls the program goes to the final optimization with the set of vertices chosen.

Using all the detected vertices could, depending on the problem under investigation easily involve so many constraints that the optimization would be very time consuming. This could, however, for some problems, be overcome by specifying a sufficiently large but reasonable number MAXVN (the maximum number of constraints involved would be MAXVN multiplied by the number of sample points NSP). In such cases the updating and optimization procedure will converge if the vertices, which are active at the solution, are not discarded during updating. The same convergence will occur if manual elimination by the user is performed without discarding vertices which are active at the solution.

It should be pointed out that vertices which are detected at an early stage of the updating procedure need not be active at the solution and vice-versa. The final solution is worst-case only at the chosen sample points.

The solution process may provide valuable information to the user, e.g., parameter or frequency symmetry, which could be useful in order to reduce the number of active vertices.

## Options and Default Values

Options and default values are used to enhance flexibility. Table I summarizes the features and options. Table II summarizes the optimization methods. Tables III and IV survey parameters used in TOLOPT.

Table III involves parameters which decide on vertices selection, continuous and discrete optimization and default values. Table IV surveys certain parameters, some of which have to be set on entry each time TOLOPT is used. Others have to be set only when the indicated parts of TOLOPT are used.

## III. ARGUMENT LIST

The TOLOPT program is called by

SUBROUTINE TOLOPT (NR,KT,KR,KD,KP,NP,Z,I1,I2,AZ,AX,MU,NV,SAMPT,GRAD, PL,PU,W1,CW,IB1,SG,I3,I4,X,EPS,G,PS,XB,IX,X1,X2,W,H,XE,INDX,GF,IAA,IBB,A, T1,T1P,NSTEP,QSTEP,DISCR,XU,XL,ID,IB,ICHECK,IVAR,P1,P2,ESTD,AL,GPHI,PHI)

and two common statements (see Tables III and IV)

COMMON/TOL/IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,IOPT4, IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,NSP,MAXVN,NVSUM, NEC,ND1,ND6

and

COMMON/DEFAULT/EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN

of which the common statement /DEFAULT/ only has to be specified in the user supplied main program, if the default values are not to be used.

The arguments are as follows.

| | |
|---|---|
| NR | number of independent optimization variables ($NR \geq 2$) |
| KT | number of toleranced components |
| KR | number of toleranced components of relative value |
| KD | number of variables of discrete value |
| KP | integer constant of value 2*KT |
| NP | number of p-values used in the final optimization |
| Z | vector of KP elements in which the user has to supply initial relative tolerances and absolute tolerances followed by corresponding nominal values. Z will on exit contain the optimum solution. |
| I1 | integer vector of KP elements in which the user on entry has to identify the elements of Z<br>The following indicators should be used<br>1: for discrete value<br>2: for continuous value<br>3: for fixed value |
| I2 | working vector of KP elements |
| AZ,AX | working vectors of KT elements |
| MU | array of KT by NVC elements in which the current number of vertices at all frequency points are stored .<br>NVC is the number of vertices chosen at all sample points and has to be foreseen by the user. Maximum is (2**KD)*NSP. |
| NV | vector of 2*NSP elements in which the current number of vertices is stored for each sample point |

| | |
|---|---|
| SAMPT | array of 3 by NSP elements. The user has to supply the following on entry for each sample point: SAMPT(1,.) the controlling parameter $\psi_i$ SAMPT(2,.) the specification $S_i$ SAMPT(3,.) the weighting factor $w_i$ $\psi_i$ is an independent parameter, e.g., frequency, or any number to identify a particular function. $w_i$ is given by $w_i = \begin{cases} +1 \text{ for upper specifications } S_i \\ -1 \text{ for lower specifications } S_i \end{cases}$ |
| GRAD | working vector of KT elements |
| PL,PU | vectors of KT elements to be set on entry. PL denotes lower bounds on the toleranced components PU denotes upper bounds on the toleranced components |
| W1 | working vector of KP elements |
| CW,IB1 | working vectors of KT elements. As default values $CW_i$ is set to one. By using the default parameter ND2 the user can on entry supply any other value |
| SG,I3,I4 | vectors of a number of elements to be set to the anticipated number of vertices chosen at all sample points (say NVC) |
| X | vector of KP elements. The current values of the variables for the continuous optimization are stored in the first NR elements of this vector |
| EPS | vector of NR elements to be set to the test quantities used in the Fletcher program |
| G | vector of NR elements in which the gradient vector corresponding to the optimization variables is |

|  |  |
|---|---|
|  | currently stored |
| PS | vector of NP elements to be set to the values of p used in the final optimization |
| XB,IX,X1,X2 | working vectors of NR elements |
| W | working vector of 4*NR elements |
| H | working vector of NR*(NR+1)/2 elements |
| XE | a three suffix working array of NR by NP by NP elements |
| INDX,GF | working vectors of NR elements |
| IAA,IBB,A,T1,T1P | working vectors of NCONS+1 elements In the case of the continuous problem the total number of constraints NCONS is computed by the program as NCONS=NVC + 2*NPC + NEC, where NPC is the number of non-fixed elements of Z (computed in the program) and NVC is the total number of detected vertices. In the discrete problem the total number of constraints is computed from the program DISOP2 as NCONS = (NVC + 2*NPC + NEC) + M, where M is an updating number corresponding to the number of extra constraints added at each node. |
| NSTEP | vector of KD elements to be set to the number of discrete values available for each of the KD discrete variables if IOPT5 = 1 |
| QSTEP | vector of KD elements to be set to the quantization step sizes for the KD discrete variables of IOPT5 $\neq$ 1 |
| DISCR | array of KD by NSTEP elements to be set to the discrete values imposed upon each discrete variable if IOPT5 = 1. |
| XU,XL | working vectors of KD elements |
| ID | working vector of 2**KD elements |
| IB | working array of KD by 2**KD elements |

| | |
|---|---|
| ICHECK,IVAR,P1, P2,ESTD,AL | working vectors of NCONS + 1 elements |
| GPHI | working array of KP by NCONS+1 elements |
| PHI | working vector of NCONS + 1 elements |

For the discrete problem each value is considered as a discrete number if it falls within a tolerable error from the given values. The program takes this tolerable error as 1 % of the lowest discrete value given for each discrete component. The tolerable errors are stored in vectors ERR and ERRO for problems with discrete values and uniform quantization steps, respectively. The program is limited to handle 25 discrete tolerable errors. To increase this limit the common statement /TOL4/ in the subroutines DISTRF and DISOP2 should be modified.

## IV. USER SUBROUTINES

The user must supply subroutines NETWRK and USERCN as follows.

SUBROUTINE NETWRK (Y,OM,RSP,GRAD,IG)

DIMENSION Y(1),GRAD(1)

| | |
|---|---|
| RSP | set to the actual response function of the physical circuit components in the array Y and the controlling parameter OM (=SAMPT(1,.)) |
| IF(IG.EQ.0)* | IG is an indicator which in TOLOPT is set to 1 whenever the gradients are required. * denotes either a RETURN or a GO TO statement for jumping to the first executable statement following computation of the gradients |

```
.

.

.

GRAD(i)             partial derivative of the response function w.r.t.

                    the ith element of Y

.

.

.

RETURN

END

SUBROUTINE USERCN (Z,PHI,GPHI,NR,KP)

DIMENSION Z(1),PHI(1),GPHI(KP,1)

.

.

PHI(i)              the ith inequality constraint function of Z required

                    by the user

.

.

.

GPHI(i,j)           partial derivative of the jth constraint function w.r.t.

                    the ith element of Z

.

RETURN

END
```

The user should supply the heading, dimensions, return and end statements of USERCN even if he does not supply any extra constraints.

## V. SUBROUTINES CALLED BY TOLOPT

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

| | |
|---|---|
| UPDATE | stores new vertices following previously detected ones |
| XZTRAN | reorders the user supplied Z-vector in optimization order |
| BDDB | converts -1 and +1 digits to integer number and vice-versa. IMODE = 0 converts integer number to digits in -1 and +1 states IMODE = 1 converts -1 and +1 digits to integer number |
| SORT | rearranges value of detected constraints in ascending order. Rearranges also the corresponding vertices |
| DISTRF | transforms the user supplied discrete values to appropriate values for the discrete problem and selects a tolerable error for discrete values |
| COSTFN | defines the cost function and its derivatives |

The following is a brief description of the subroutines called by DISOP2, also called by TOLOPT.

| | |
|---|---|
| DSPTA | coordinates the input, the output and the minimization |
| DSPTB | minimizes a function using the Fletcher unconstrained minimization program |
| DSPTC | formulates the artificial unconstrained objective function and the necessary gradients |
| DSPTD | supplies additional variable constraints for discrete optimization |
| DSPTE | returns the gradients of the additional variable constraints |

| | |
|---|---|
| DSPTH | transforms a nonlinear programming problem into an equivalent unconstrained objective function |
| DSPTI | prints the input data |
| DSPTJ | prints out the result of the feasibility check and/or the optimum solution at each node |
| DSPTK | prints out the best current discrete solution after checking the vertices about the continuous solution and the optimum discrete solution |
| DSPTL | checks the gradient formulation by perturbation |
| DSPTM | performs extrapolation when using algorithm 3 |

## VI. EXAMPLES

### Example 1: Design of a voltage divider [4,15]

A diagram of the voltage divider considered is shown in Fig. 3. The transfer function is $\phi_2/(\phi_1+\phi_2)$ and the input resistance $\phi_1+\phi_2$. The design specifications are $0.46 \leq \phi_2/(\phi_1+\phi_2) \leq 0.53$ and $1.85 \leq \phi_1+\phi_2 \leq 2.15$. In the case of the discrete problem the set of obtainable discrete values for the tolerances of $\phi_1$ and $\phi_2$ are

DISCR = {1,3,5,10,15} per cent.

A typical main program to supply the values and proper dimensioning for the parameters in the argument list of subroutine TOLOPT and the common statements /TOL/ and /DEFAULT/ is displayed in Fig. 4. Fig. 5 shows the subroutine NETWRK and Fig. 6 illustrates USERCN for a constraint inactive at the solution. Typical printouts of data and the gradient check are shown in Figs. 7 and 8, respectively. Results of continuous and discrete optimizations are shown in Fig. 9.

In this example all four known vertices are supplied and by setting IUPD to zero, the TOLOPT program goes directly to the final optimization.

To further specify the given vertices the parameter NVSUM is set to four and the vector NV set to $[1\ 1\ 1\ 1]^T$ to identify specific vertices and sample points. The MU-Matrix given is

$$MU = \begin{bmatrix} -1 & -1 & 1 & 1 \\ -1 & 1 & -1 & 1 \end{bmatrix} .$$

An alternative and, for problems with several toleranced components and many detected vertices, more practical way of specifying a set of vertices is as follows.

1- supply in vector I3 a set of integer numbers corresponding to the chosen vertices. (See below for a unique relationship between a vertex and the integer number).

2- call subroutine BDDB using IMODE = 0 to convert each integer number into -1 and +1 states corresponding to the vertex.

3- transfer the output IB1 from BDDB into the MU-matrix.

The following example will demonstrate the conversion of a chosen vertex to the corresponding integer number. Each state -1 is substituted by 0(zero) and the binary representation is converted to an integer number:

chosen vertex $\underset{\sim}{\mu} = [1\ -1\ 1\ 1]^T$ => equiv. binary 1 0 1 1 =>

integer number I3 $= (1.2^0 + 0.2^1 + 1.2^2 + 1.2^3) + 1 = 14.$

In the case of the continuous problem it is seen that only two of the four vertices are active at the solution. The parameter bounds have been chosen such that these will not be active at the solution.

In the case of the discrete problem eight nodes have been searched to identify the discrete solution. Generally, the number of nodes searched by the program DISOP2 depends highly on the parameter ZERO, the value of which can be modified by the user using the default parameter ND5.

Example 2: Design of an LC-lowpass filter [2-4]

This is the same problem as described in [3]. A circuit diagram is shown in Fig. 10. Fig. 11 shows the results from the vertices selection procedure. With the parameter IWORST set to 2 the figure shows data from vertices selection scheme 2, where only those detected vertices associated with negatively valued constraints are printed out. The print-out of the vector NV denotes the number of vertices detected at each sample point. The columns containing -1 and +1 relate to specific detected vertices and are stored in the MU matrix.

Generally the program will print out the MU-matrix in parts, each of which contains a maximum of 25 columns.

Fig. 12 shows the results from the continuous and discrete optimizations. In the case of the continuous problem it is seen that only 5 vertices are active at the solution. Re-running the problem with these vertices only will give the same solution. Furthermore, it may be noted that the continuous solution yields symmetrical results although symmetry is not assumed in the formulation of the problem.

Example 3: 10/1 quarter-wave transformer [3]

This is the same problem as described in [3]. Fig. 13 shows results from one of the updating procedures. Vertices selection scheme 1 has been used. Fig. 14 shows the continuous solutions when relative tolerances have been assumed. Fig. 15 shows results at certain nodes and which can be identified as discrete solutions, although the program does not recognize them as such. This is probably due to the tolerables errors chosen and the termination criteria for optimization. The user should exercise discretion in interpreting the results from a program as general as TOLOPT.

## VII. CONCLUSIONS

We have presented an efficient user-oriented program for worst-case tolerance optimization, particularly suited to circuit design. It is based on work carried out by Chen [4], Liu [2] and Bandler, Liu and Chen [3]. The user is well-advised to consult the appropriate references before attempting to use the TOLOPT package.

The package has been under continuous development to make it sufficiently user-oriented. This has been to some extent at the expense of the greater efficiency which can be realized by a more specialized program. In particular, the exploitation of symmetry [3] requires careful problem preparation and possibly some changes to the program. Furthermore, running times of the package can vary significantly according to the various termination and error criteria used as data. This is particularly true in the generation of the tree structure in a discrete optimization and the interpretation of the solutions as being feasible, discrete, etc.

## REFERENCES

[1] J.W. Bandler, "Optimization of design tolerances using nonlinear programming", J. Optimization Theory and Applications, vol. 14, 1974, pp. 99-114.

[2] P.C. Liu, "A theory of optimal worst-case design embodying centering, tolerancing and tuning, with circuit applications", McMaster University, Hamilton, Canada, Internal Report in Simulation, Optimization, and Control, No. SOC-87, May 1975.

[3] J.W. Bandler, P.C. Liu and J.H.K. Chen, "Worst case network tolerance optimization", IEEE Trans. Microwave Theory Tech., vol. MTT-23, Aug. 1975, pp. 630-641.

[4] 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.

[5] J.H.K. Chen, "DISOPT- a general program for continuous and discrete nonlinear programming problems", McMaster University, Hamilton, Canada, Internal Report in Simulation, Optimization and Control, No. SOC-29, March 1974 (Revised June 1975).

[6] J.W. Bandler, B.L. Bardakjian and J.H.K. Chen, "Design of recursive digital filters with optimized word length coefficients", Computer Aided Design, vol. 7, July 1975, pp. 151-156.

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

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

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

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

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

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

[13] W.Y. Chu, "Extrapolation in least pth approximation and nonlinear programming", McMaster University, Hamilton, Canada, Internal Report in Simulation, Optimization and Control, No. SOC-71, Dec. 1974.

[14] J.W. Bandler and P.C. Liu, "Some implications of biquadratic functions in the tolerance problem", IEEE Trans. Circuits and Systems, vol. CAS-22, May 1975, pp. 385-390.

[15] B.J. Karafin, "The optimum assignment of component tolerances for electrical networks", B.S.T.J., vol. 50, April 1971, pp. 1225-1242.

TABLE I

SUMMARY OF FEATURES, OPTIONS, PARAMETERS AND SUBROUTINES REQUIRED

| Features | Type | Options | Parameters[†]/subroutines |
|---|---|---|---|
| Design parameters | Tolerance and Nominal | Variable or fixed<br>Relative or absolute tolerances | Number of parameters<br>Starting values<br>Indication for fixed or variable nominals and relative or absolute tolerances |
| Objective function | Cost | Reciprocal of relative and/or absolute tolerances | Weighting factors |
| | | Other | Subroutine to define the objective function and its partial derivatives |
| Vertices selection* | Gradient direction strategy | | Maximum allowable number of calls of the vertices selection subroutine |
| Constraints | Specifications on functions of network parameters | Upper and/or lower | Sample points (e.g., frequency)<br>Specifications<br>Subroutine to calculate, for example, the network response and its partial derivatives (NETWRK) |
| | Network parameter bounds | | Upper and lower bounds |
| | Other constraints | As many as required | Subroutine to define the constraint functions and their partial derivatives (USERCN) |

to be continued

TABLE I [Continued].

| | | | |
|---|---|---|---|
| Nonlinear programming | Bandler-Charalambous minimax | Least pth optimization algorithms<br>See Table II | Controlling parameter $\alpha$<br>Value(s) of p<br>Test quantities for termination |
| | Exterior-point | | Optimistic estimate of objective function<br>Value of p |
| Solution feasibility check* | Least pth | Discrete problem<br>Continuous and discrete problem | Constraint violation tolerance<br>Value of p |
| Unconstrained minimization method | Quasi-Newton | Gradient checking at starting point by numerical perturbation | Number of function evaluations allowed<br>Estimate of lower bound on least pth objective<br>Test quantities for termination |
| Discrete optimization* | Dakin tree-search | Reduction of dimensionality<br>User supplied or program determined initial upper bound on objective function<br>Single or multiple optimum discrete solution<br>Uniform or nonuniform quantization step sizes | Upper bound on objective function<br>Maximum permissible number of nodes<br>Discrete values on step sizes<br>Number of discrete variables<br>Discrete value tolerance<br>Order of partitioning<br>Indication for discrete variables |

† Parameters associated with the options are not explicitly listed.

* These features are optional and may be bypassed.

TABLE II

THE OPTIONAL LEAST PTH ALGORITHMS †

| Algorithm | Definition of $e_i$ | Convergence feature | Value(s) of p | Number of optimizations |
|---|---|---|---|---|
| 1 | $e_i \leftarrow \begin{cases} f - \alpha g_i, i=1,2,\ldots,m \\ f, \; i = m+1 \end{cases}$ | | Large | 1 |
| 2 | where $\alpha > 0$ | Increment of p | Increasing | Implied by the sequence but superceded by the stopping quantity |
| 3 | | Extrapolation | Geometrically increasing | |
| 4 | $e_i \leftarrow \begin{cases} f - \alpha g_i - \xi^r, i=1,2,\ldots,m \\ f - \xi^r, i = m+1 \end{cases}$ where $\alpha > 0$, $\xi^r \leftarrow \begin{cases} \min[0, M^0 + \gamma], \; r=1 \\ \check{M}^{r-1} + \gamma, \; r > 1 \end{cases}$ r indicates the optimization number; $\gamma$ is a small positive quantity | Updating of $\xi^r$ | Finite | Depend on the stopping quantity |
| 5 | $e_i \leftarrow \begin{cases} -g_i, \; i=1,2,\ldots,m \\ f - t^r, \; i = m+1 \end{cases}$ where $t^r \leftarrow \begin{cases} \text{optimistic estimate of } \check{f}, \; r = 1 \\ t^{r-1} + \check{U}^{r-1}, \; r > 1 \end{cases}$ r is defined as in 4 | Updating of $t^r$ | | |

†For definitions of the parameters see [3].

TABLE III
PARAMETERS IN TOLOPT FOR
VERTICES SELECTION, OPTIMIZATION AND DEFAULT VALUES

| | | | Vertices Select-ion | | Continuous and Discrete Optimization | | | | | | | Default Values | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | IUPD | ISCHEME | IOPT1 | IOPT2 | IOPT3 | IOPT4 | IOPT5 | IOPT6 | IOPT7 | ND1 | ND2 | ND3 | ND4 | ND5 | ND6 |
| Vertices Selection | | Vertices selection only | <0,>2 | | | | | | | | | | | | | | |
| | | User supplied MU-matrix -final optimization | 2 | | | | | | | | | | | | | | |
| | | User supplied MU-matrix-updating-optimization | 1 | | | | | | | | | | | | | | |
| | | Automated vertices selection | 0 | | | | | | | | | | | | | | |
| | | One vertex selected at nominal point | | 0 | | | | | | | | | | | | | |
| | 1 | Number of vertices selected | | ±1 | | | | | | | | | | | | | |
| | 1 | All vertices selected | | ±2 | | | | | | | | | | | | | |
| Continuous and Discrete Optimization | | Dimensionality of discrete problem reduced by 1 | | | 1 | | | | | | | | | | | | |
| | | Gradients supplied in user subroutines checked | | | | 1 | | | | | | | | | | | |
| | | Vertices about cont.sol. checked in discr. problem | | | | | 1 | | | | | | | | | | |
| | 2 | Feasibility checked-1 from very beginning, 2 only in discrete problem | | | | | | 1,2 | | | | | | | | | |
| | | Finite set of discrete values used | | | | | | | 1 | | | | | | | | |
| | 3 | An integer set to I if algorithm I is to be used | | | | | | | | . | | | | | | | |
| | | Only one discrete solution is required | | | | | | | | | 1 | | | | | | |
| Default Values Used | 7 | Maximum number of vertices selections equals 5 | | | | | | | | | | 0 | | | | | |
| | 7 | Weighting factors in obj. function equals 1 | | | | | | | | | | | 0 | | | | |
| | 4,7 | EST=0., EST1=0. | | | | | | | | | | | | 0 | | | |
| | 5,7 | AO=10., AI=10., XMAL=1.E5 | | | | | | | | | | | | | 0 | | |
| | 6,7 | ZERO, ETA=1.E-4 INSOLN=0 | | | | | | | | | | | | | | 0 | |
| | 7 | pvalue used in updating procedure equals 10 | | | | | | | | | | | | | | | 0 |

TABLE III

(Cont'd)

| | |
|---|---|
| 1 | Minus sign in front of the ISCEME-indicator indicates a further reduction based on magnitude considerations |
| 2 | The feasibility is checked according to these settings in the final optimization (feasibility is always checked with respect to nominal starting point) |
| 3 | Table I surveys the optimization methods. The • means that a suitable number has to be set |
| 4 | EST A real number set to the estimated minimum value of the artificial unconstrained objective function<br>EST1 A real number set to the initial estimated minimum value of the actual objective function when using algorithm 5 |
| 5 | AO A real number set to the initial value of $\alpha$ when using algorithms 1 to 4<br>AI Multiplication factor of $\alpha$ when using algorithms 1 to 4<br>XMAL Maximum allowable value of $\alpha$ when using algorithms 1 to 4 |
| 6 | ZERO Set to 1% of the smallest/largest given specification if it is positive/negative<br>ETA Stopping test quantity when using algorithms 2, 4 or 5<br>INSOLN Set to 1 if an upper bound on the actual function value is available<br>BSOLN Upper bound on the actual function value if INSOLN=1 |
| 7 | To use other than default values supply alternatives<br>Note: ND1 will be used as maximum number |

TABLE IV
OTHER PARAMETERS IN TOLOPT

| | Parameters which have to be given | | | | | | | Parameters which must be given conditionally | | | | * Indicates that proper value has to be set |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | NSP | NSEC | NDIM | IDATA | IPRINT | MAX | MAXNOD | IWORST | NVSUM | MAXVN | ICON | To be set |
| Number of sample points | * | | | | | | | | | | | On entry |
| Number of extra contraints given in USERCN | | * | | | | | | | | | | On entry |
| Number of anticipated columns in matrix GPHI | | | * | | | | | | | | | On entry |
| Printing of input data | | | | 1 | | | | | | | | On entry |
| Output printing of optimization data. Printing for every IPRINT iterations | | | | | * | | | | | | | On entry |
| Printing at each node | | | | | 0 | | | | | | | On entry |
| Printing of optimum continous and discrete solutions only | | | | | -1 | | | | | | | On entry |
| Printing suppressed | | | | | -2 | | | | | | | |
| Maximum permissible number of function evaluations per node | | | | | | * | | | | | | On entry |
| Maximum number of nodes to be searched. MAXNOD = 0 if only continuous sol. is required | | | | | | | * | | | | | On entry |
| 1-Print all vertices and corresp. constraint value 2-Print only vertices associated with neg. constraints | | | | | | | | 1,2 | | | | If third vertices selection scheme is used |
| Number of elements in user supplied MU-matrix | | | | | | | | | * | | | If IUPD = 0 or IUPD = 1 |
| Maximum allowable number of vertices at each sample point | | | | | | | | | | * | | When second or third vertices selection scheme is used |
| Partitioning is imposed on first discrete parameter first, any other value will impose partitioning on last discrete parameter first | | | | | | | | | | | 1 | If discrete optimization is performed |

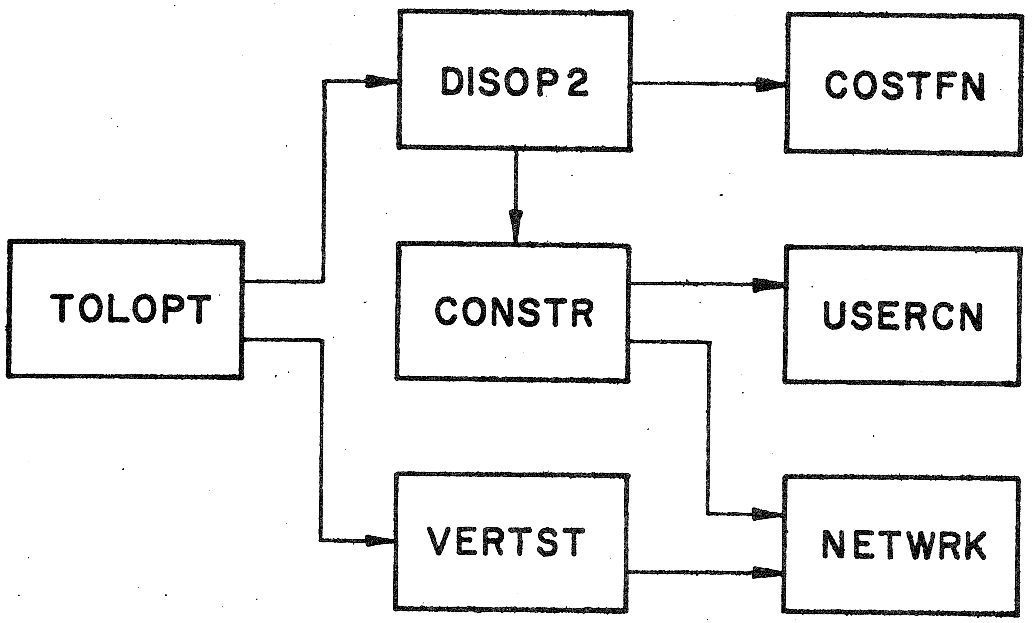
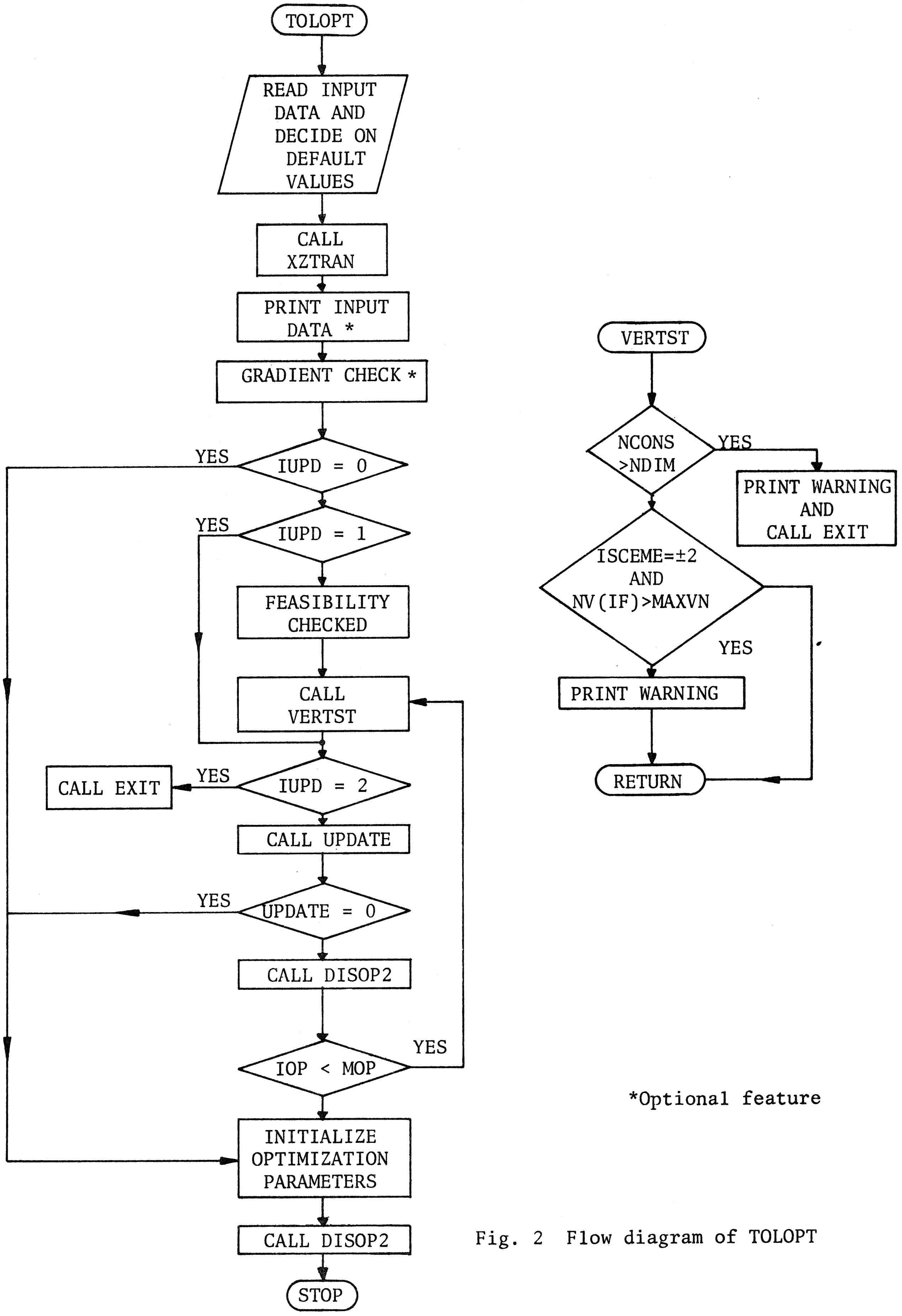


Fig. 1 The overall structure of TOLOPT. The user is responsible for NETWRK and USERCN.



*Optional feature

Fig. 2 Flow diagram of TOLOPT

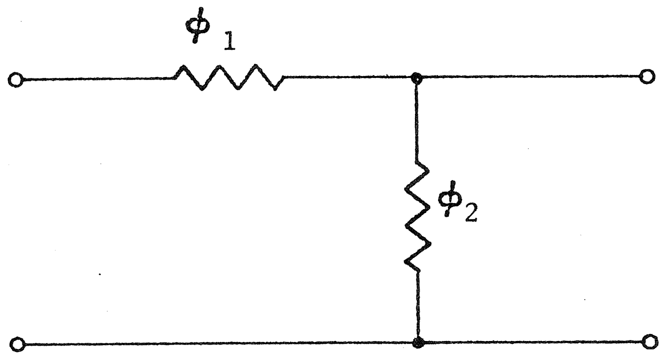


Fig. 3 The voltage divider example (Example 1)

```
      PROGRAM TESTVOL(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
      DIMENSION X(4),EPS(4),G(4),PS(1),XB(4),IX(4),X1(4),X2(4),W(16),
     *   H(10),XE(4,1,1),INDX(4),GF(4)
      DIMENSION IAA(50),IBB(50),A(50),T1(50),T1P(50)
      DIMENSION NSTEP(2),QSTEP(2),DISCR(2,5),XU(2),XL(2),ID(25),IB(2,25)
     *  ,ICHECK(25),IVAR(25),P1(25),P2(25),ESTD(25),AL(25)
      DIMENSION Z(4),I1(4),I2(4),AZ(2),AX(2),MU(2,10),NV(10),SAMPT(3,10)
     *  ,GRAD(2),PL(2),PU(2),W1(4),CW(2),IB1(2),SG(10)
      DIMENSION GPHI(4,24),PHI(24),I3(10),I4(10)
      COMMON /TOL/ IUPD,ISCEME,IWORST,IPRINT,IDATA,IOPT1,IOPT2,IOPT3,
     *   IOPT4,IOPT5,IOPT6,IOPT7,ND2,ND3,ND4,ND5,MAX,MAXNOD,ICON,NDIM,
     *   NSP,MAXVN,NVSUM,NEC,ND1,ND6
      COMMON /DEFAULT/ EST,EST1,AO,AI,XMAL,ZERO,ETA,INSOLN,BSOLN
      DATA KT,NR,KR,KD /2,4,2,2/
      DATA IPRINT,IDATA,MAX,NP,MAXNOD,ICON /10,1,1000,1,10,0/
      DATA IOPT1,IOPT2,IOPT3,IOPT4,IOPT5,IOPT6,IOPT7 /1,1,1,1,1,4,0/
      DATA NSP,MAXVN,NDIM,NEC /4,4,25,1/
      DATA NVSUM,IUPD /4,0/
      DATA ND1,ND2,ND3,ND4,ND5,ND6 /0,0,0,1,1,0/
      DATA AO,AI,XMAL / 100.,10.,1.E+7/
      DATA ZERO,ETA,INSOLN /-1.E-6,1.E-3,0/
      DATA Z /0.01,0.01,1.0,1.0/
      DATA I1 /1,1,2,2/
      DATA EPS /4*1.E-6/
      DATA PL /0.80,0.80/
      DATA PU /1.20,1.20/
      DATA PS /6./
      DATA NSTEP /2*5/
      DATA QSTEP /2*1./
      DATA ((DISCR(I,J),I=1,2),J=1,5) /2*1.,2*3.,2*5.,2*10.,2*15./
      KP=2*KT
      READ 3,((SAMPT(I,J),I=1,3),J=1,NVSUM)
    3 FORMAT(3F5.2)
      READ 4,((MU(I,J),J=1,NVSUM),I=1,KT)
      READ 4,(NV(I),I=1,NVSUM)
    4 FORMAT(4I3)
      CALL TOLOPT (NR,KT,KR,KD,KP,NP,Z,I1,I2,AZ,AX,MU,NV,SAMPT,
     *   GRAD,PL,PU,W1,CW,IB1,SG,I3,I4,X,EPS,G,PS,XB,IX,X1,X2,
     *   W,H,XE,INDX,GF,IAA,IBB,A,T1,T1P,NSTEP,QSTEP,DISCR,XU,XL,ID,
     *   IB,ICHECK,IVAR,P1,P2,ESTD,AL,GPHI,PHI)
      STOP
      END
```

Fig. 4  Main program for Example 1

```
      SUBROUTINE NETWRK (AX,OM,RSP,GR,IG)
      DIMENSION AX(1),GR(1)
C     VOLTAGE DIVIDER EXAMPLE
      A=AX(1)+AX(2)
      A2=A**2
      T=AX(2)/A
C     FOUR KNOWN VERTICES GIVES FOLLOWING CONSTRAINT FUNCTIONS
C       AND CORRESPONDING DERIVATIVES ARRANGED IN SUCH A WAY
C         THAT THE CONSTRAINTS SUBROUTINE IN TOLOPT CAN BE USED
C         DIRECTLY.
      KV=IFIX(OM)
      GO TO(1,2,2,1),KV
    1 RSP=A
      IF(IG.EQ.0) RETURN
      GR(1)=1.
      GR(2)=1.
      RETURN
    2 RSP=T
      IF(IG.EQ.0) RETURN
      GR(1)=-AX(2)/A2
      GR(2)=AX(1)/A2
      RETURN
      END
```

Fig. 5  Subroutine NETWRK for Example 1

```
      SUBROUTINE USERCN (Z,G,GG,NR,KP)
      DIMENSION Z(1),G(1),GG(KP,1)
C     THIS EXTRA CONSTRAINT HAS BEEN CHOSEN ONLY TO DEMON-
C        STRATE THE USE OF USERCN
      G(1)=Z(3)+Z(4)
      GG(1,1)=0.
      GG(2,1)=0.
      GG(3,1)=1.
      GG(4,1)=1.
      RETURN
      END
```

Fig. 6  Subroutine USERCN for Example 1

```
INPUT DATA
----------

DISCRETE VALUES FOR THE VARIABLES

Z( 1)    .10000000E+01   .30000000E+01   .50000000E+01   .10000000E+02   .15000000E+02

Z( 2)    .10000000E+01   .30000000E+01   .50000000E+01   .10000000E+02   .15000000E+02

USER SUPPLIED COMPONENTS........TOLERANCE,DISCRETE.............Z( 1)=   .10000000E-01
                                TOLERANCE,DISCRETE.............Z( 2)=   .10000000E-01
                                NOMINAL,CONTINUOUS.............Z( 3)=   .10000000E+01
                                NOMINAL,CONTINUOUS.............Z( 4)=   .10000000E+01

ERROR TOLERANCE IN CONSTRAINTS.................................ZERO=  -.10000000E-05

TEST QUANTITIES TO BE USED IN FLETCHER METHOD...............EPS( 1)=   .10000000E-05
                                                            EPS( 2)=   .10000000E-05
                                                            EPS( 3)=   .10000000E-05
                                                            EPS( 4)=   .10000000E-05

ESTIMATE OF LOWER BOUND ON ARTIFICIAL OBJECTIVE FUNCTION.........EST=  0.

INITIAL VALUE OF THE PARAMETER ALPHA..............................AO=   .10000000E+03

MAXIMUM ALLOWABLE VALUE OF THE PARAMETER ALPHA..................XMAL=   .10000000E+08

MULTIPLYING FACTOR IN ALPHA VALUE.................................AI=   .10000000E+02

TEST QUANTITY TO BE USED IN NLP ALGORITHM 2/4/5..................ETA=   .10000000E-02

NUMBER OF P VALUES................................................NP=    1

VALUE(S) OF P USED IN NLP ALGORITHM...........................PS( 1)=   .60000000E+01

FOLLOWING OPTIONS USED
----------------------

NLP ALGORITHM 4 EMPLOYED-SEQUENCE OF LEAST PTH OPTIMIZATION WITH FINITE VALUES OF P

(N-1) VARIABLE OPTIMIZATION PERFORMED IN DISCRETE PROBLEM

VERTICES CHECKED ABOUT CONTINUOUS SOLUTION TO OBTAIN AN INITIAL UPPER BOUND IN
DISCRETE PROBLEM

FEASIBILITY CHECKED IN FINAL OPTIMIZATION

PARTITIONING STARTS ON LAST DISCRETE VARIABLE
```

Fig. 7  Printout of data for Example 1

```
DATA GIVEN FOR SPECIFIC PROBLEM
-------------------------------

        IF        SAMPT(1,IF)        SAMPT(2,IF)        SAMPT(3,IF)

        1         .101000E+01        .185000E+01       -.100000E+01
        2         .201000E+01        .460000E+00       -.100000E+01
        3         .301000E+01        .530000E+00        .100000E+01
        4         .401000E+01        .215000E+01        .100000E+01
```

Fig. 7  [Continued]

```
GRADIENT CHECK AT NOMINAL STARTING POINT
----------------------------------------

THE GRADIENTS FROM THE USER SUPPLIED NETWRK HAVE BEEN CHECKED AT THE
FIRST SAMPLE POINT

  ANALYTICAL GRADIENTS  NUMERICAL GRADIENTS  PERCENTAGE ERRORS
        .100000E+01          .100000E+01         .378577E-08
        .100000E+01          .100000E+01         .378577E-08

THE GRADIENTS FROM NETWRK HAVE BEEN CHECKED AT ALL SAMPLE POINTS
THE LARGEST OVERALL DETECTED ERRORS ARE AS FOLLOWS

  ANALYTICAL GRADIENTS  NUMERICAL GRADIENTS  PERCENTAGE ERRORS  SAMPLE POINT
       -.250000E+00         -.250000E+00         .244904E-06            2
        .250000E+00          .250000E+00         .244904E-06            2


THE GRADIENTS FROM THE USER SUPPLIED USERCN HAVE BEEN CHECKED
FOR EACH GIVEN EXTRA CONSTRAINT THE ERRORS ARE AS FOLLOWS

  ANALYTICAL GRADIENTS  NUMERICAL GRADIENTS  PERCENTAGE ERRORS  CONSTRAINT
        .100000E-13          .100000E-13       0.                       1
        .100000E-13          .100000E-13       0.                       1
        .100000E+01          .100000E+01         .378577E-08            1
        .100000E+01          .100000E+01         .378577E-08            1

GRADIENTS ARE O.K.
```

Fig. 8  Check of user supplied gradients for Example 1

```
  RESULTS OF THE FEASIBILITY CHECK
  --------------------------------

                  NODE NUMBER =    0

INEQUALITY CONSTRAINTS      OCCURRING AT

G( 1)=    .13000000E+00     SAMPLE POINT 1
G( 2)=    .35000000E-01     SAMPLE POINT 2
G( 3)=    .25000000E-01     SAMPLE POINT 3
G( 4)=    .13000000E+00     SAMPLE POINT 4
G( 5)=    .19000000E+00     LOWER BOUND  1
G( 6)=    .19000000E+00     UPPER BOUND  1
G( 7)=    .19000000E+00     LOWER BOUND  2
G( 8)=    .19000000E+00     UPPER BOUND  2
G( 9)=    .20000000E+01     EXTRA CONST  1

  NUMBER OF CONSTRAINTS USED =     9

NUMBER OF VIOLATED CONSTRAINTS =   0

NUMBER OF FUNCTION EVALUATIONS =   1


  FOLLOWING IS RESULT OF OPTIMIZATION
  -----------------------------------
```

Fig. 9  Results of the continuous and discrete optimizations for Example 1

```
                            NODE NUMBER =     0
ARTIFICIAL UNCONSTRAINED FUNCTION U =  -.11592424E-01
      ACTUAL OBJECTIVE FUNCTION F =   .28569099E+02

      X( 1)=    .26458592E+00 GU( 1)=   .15487643E-03
      X( 2)=    .26458592E+00 GU( 2)=   .16120715E-03
      X( 3)=    .10139413E+01 GU( 3)=   .10367310E-03
      X( 4)=    .99376532E+00 GU( 4)=  -.10577793E-03

      INEQUALITY CONSTRAINTS     OCCURRING AT

      G( 1)=    .17155658E-01     SAMPLE POINT 1
      G( 2)=    .66321421E-06     SAMPLE POINT 2
      G( 3)=    .66395748E-06     SAMPLE POINT 3
      G( 4)=    .17425017E-02     SAMPLE POINT 4
      G( 5)=    .14295958E+00     LOWER BOUND  1
      G( 6)=    .11507707E+00     UPPER BOUND  1
      G( 7)=    .12419608E+00     LOWER BOUND  2
      G( 8)=    .13666543E+00     UPPER BOUND  2
      G( 9)=    .20077066E+01     EXTRA CONST  1

        NUMBER OF CONSTRAINTS USED =     9

     NUMBER OF VIOLATED CONSTRAINTS =     0

     NUMBER OF FUNCTION EVALUATIONS =   192

 FINAL VALUE OF THE PARAMETER ALPHA =   .10000000E+05


         FOLLOWING IS THE OPTIMUM SOLUTION
         ---------------------------------
      Z( 1)=    .70005708E-01
      Z( 2)=    .70005708E-01
      Z( 3)=    .10139413E+01
      Z( 4)=    .99376532E+00

        EXECUTION TIME IN SECONDS =    4.20300
```

Fig. 9  [Continued]

```
           BEST DISCRETE SOLUTION FOUND SO FAR
                                  F =   .40000000E+02

       X( 1)=    .50000000E-01
       X( 2)=    .50000000E-01
       X( 3)=    .10139413E+01
       X( 4)=    .99376532E+00

       INEQUALITY CONSTRAINTS      OCCURRING AT

       G( 1)=    .57321249E-01     SAMPLE POINT 1
       G( 2)=    .99904561E-02     SAMPLE POINT 2
       G( 3)=    .10014581E-01     SAMPLE POINT 3
       G( 4)=    .41908093E-01     SAMPLE POINT 4
       G( 5)=    .16324419E+00     LOWER BOUND  1
       G( 6)=    .13536168E+00     UPPER BOUND  1
       G( 7)=    .14407706E+00     LOWER BOUND  2
       G( 8)=    .15654641E+00     UPPER BOUND  2
       G( 9)=    .20077066E+01     EXTRA CONST  1
  NUMBER OF FUNCTION EVALUATIONS =  198

           RESULTS OF THE FEASIBILITY CHECK
           --------------------------------

                            NODE NUMBER =    8

       INEQUALITY CONSTRAINTS      OCCURRING AT

       G( 1)=    .94253713E-02     SAMPLE POINT 1
       G( 2)=   -.34594404E-02     SAMPLE POINT 2
       G( 3)=    .68230156E-02     SAMPLE POINT 3
       G( 4)=    .26401155E-01     SAMPLE POINT 4
       G( 5)=    .16418646E+00     LOWER BOUND  1
       G( 6)=    .13432023E+00     UPPER BOUND  1
       G( 7)=    .95238913E-01     LOWER BOUND  2
       G( 8)=    .14208092E+00     UPPER BOUND  2
       G( 9)=    .19915121E+01     EXTRA CONST  1
         NUMBER OF CONSTRAINTS USED =   10
      NUMBER OF VIOLATED CONSTRAINTS =    2
      NUMBER OF FUNCTION EVALUATIONS =   10
           EXECUTION TIME IN SECONDS =    .27500
```

Fig. 9 [Continued]

```
                 OPTIMUM DISCRETE SOLUTION FOUND
                       MINIMUM   F =    .40000000E+02

    X( 1)=    .50000000E-01
    X( 2)=    .50000000E-01
    X( 3)=    .10139413E+01
    X( 4)=    .99376532E+00

    INEQUALITY CONSTRAINTS      OCCURRING AT

    G( 1)=    .57321249E-01     SAMPLE POINT 1
    G( 2)=    .99904561E-02     SAMPLE POINT 2
    G( 3)=    .10014581E-01     SAMPLE POINT 3
    G( 4)=    .41908093E-01     SAMPLE POINT 4
    G( 5)=    .16324419E+00     LOWER BOUND  1
    G( 6)=    .13536168E+00     UPPER BOUND  1
    G( 7)=    .14407706E+00     LOWER BOUND  2
    G( 8)=    .15654641E+00     UPPER BOUND  2
    G( 9)=    .20077066E+01     EXTRA CONST  1
NUMBER OF FUNCTION EVALUATIONS =  620



                 FOLLOWING IS THE OPTIMUM SOLUTION
                 ---------------------------------

     Z( 1)=    .50000000E-01
     Z( 2)=    .50000000E-01
     Z( 3)=    .10139413E+01
     Z( 4)=    .99376532E+00
```

Fig. 9  [Continued]

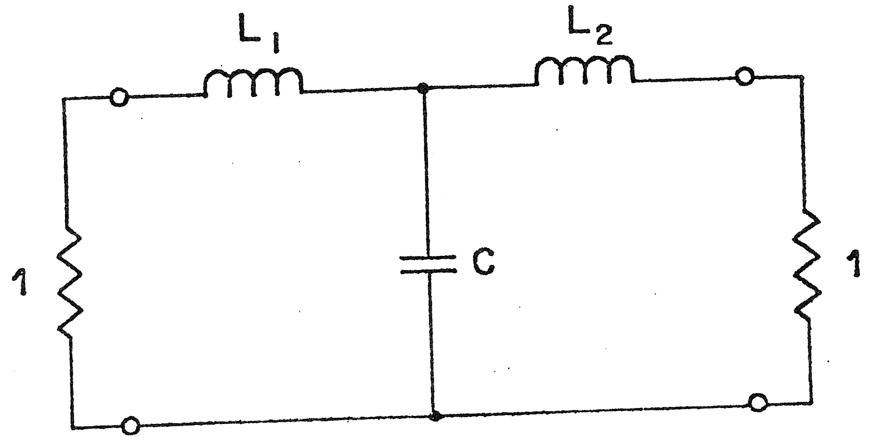


Fig. 10 The LC filter example (Example 2)

```
                DATA FROM VERTICES SELECTION NO  1
                ------------------------------------

              SAMPT(1, 4) =    1.00  VERTEX NO   8  G= -.360415E+00
              SAMPT(1, 5) =    2.50  VERTEX NO   1  G= -.204513E+01
              SAMPT(1, 5) =    2.50  VERTEX NO   2  G= -.227264E+00
              SAMPT(1, 5) =    2.50  VERTEX NO   5  G= -.227264E+00

MU-MATRIX TO BE USED IN FOLLOWING OPTIMIZATION

NV( 1)= 1 NV( 2)= 1 NV( 3)= 1 NV( 4)= 3 NV( 5)= 1


 1  1  1  1  1 -1 -1

-1 -1 -1  1  1  1 -1

 1  1  1  1 -1  1 -1



                DATA FROM VERTICES SELECTION NO  2
                ------------------------------------

              SAMPT(1, 1) =     .50  VERTEX NO   2  G= -.144878E+00
              SAMPT(1, 1) =     .50  VERTEX NO   5  G= -.144903E+00
              SAMPT(1, 1) =     .50  VERTEX NO   6  G= -.826916E+00
              SAMPT(1, 2) =     .55  VERTEX NO   2  G= -.290594E+00
              SAMPT(1, 2) =     .55  VERTEX NO   5  G= -.290622E+00
              SAMPT(1, 2) =     .55  VERTEX NO   6  G= -.911466E+00
              SAMPT(1, 3) =     .60  VERTEX NO   2  G= -.401682E+00
              SAMPT(1, 3) =     .60  VERTEX NO   5  G= -.401715E+00
              SAMPT(1, 3) =     .60  VERTEX NO   6  G= -.917050E+00
              SAMPT(1, 4) =    1.00  VERTEX NO   4  G= -.135796E+01
              SAMPT(1, 4) =    1.00  VERTEX NO   7  G= -.135805E+01
              SAMPT(1, 4) =    1.00  VERTEX NO   8  G= -.502019E+01
              SAMPT(1, 5) =    2.50  VERTEX NO   1  G= -.766354E+01
              SAMPT(1, 5) =    2.50  VERTEX NO   2  G= -.164023E+01
              SAMPT(1, 5) =    2.50  VERTEX NO   3  G= -.138788E+01
              SAMPT(1, 5) =    2.50  VERTEX NO   5  G= -.164023E+01

MU-MATRIX TO BE USED IN FOLLOWING OPTIMIZATION
  9 MORE VERTICES HAVE BEEN DETECTED IN UPDATING PROCEDURE
THE CURRENT NUMBER OF CONSTRAINTS IN PROBLEM ARE  22
NV( 1)= 1 NV( 2)= 3 NV( 3)= 4 NV( 4)= 7 NV( 5)= 1


 1  1  1 -1  1  1 -1 -1  1  1 -1  1  1 -1 -1 -1

-1 -1 -1 -1 -1 -1 -1 -1  1  1  1 -1 -1 -1 -1 -1

 1  1 -1  1  1 -1  1 -1  1 -1  1  1 -1  1 -1 -1
```

Fig. 11 Results of the vertex selection procedure for Example 2

```
      DATA FROM VERTICES SELECTION NO  3
      -----------------------------------

   SAMPT(1, 1) =     .50  VERTEX NO   2  G= -.144878E+00
   SAMPT(1, 1) =     .50  VERTEX NO   5  G= -.144903E+00
   SAMPT(1, 1) =     .50  VERTEX NO   6  G= -.826916E+00
   SAMPT(1, 2) =     .55  VERTEX NO   2  G= -.290594E+00
   SAMPT(1, 2) =     .55  VERTEX NO   5  G= -.290622E+00
   SAMPT(1, 2) =     .55  VERTEX NO   6  G= -.911466E+00
   SAMPT(1, 3) =     .60  VERTEX NO   2  G= -.401682E+00
   SAMPT(1, 3) =     .60  VERTEX NO   5  G= -.401715E+00
   SAMPT(1, 3) =     .60  VERTEX NO   6  G= -.917050E+00
   SAMPT(1, 4) =    1.00  VERTEX NO   4  G= -.135796E+01
   SAMPT(1, 4) =    1.00  VERTEX NO   7  G= -.135805E+01
   SAMPT(1, 4) =    1.00  VERTEX NO   8  G= -.502019E+01
   SAMPT(1, 5) =    2.50  VERTEX NO   1  G= -.766354E+01
   SAMPT(1, 5) =    2.50  VERTEX NO   2  G= -.164023E+01
   SAMPT(1, 5) =    2.50  VERTEX NO   3  G= -.138788E+01
   SAMPT(1, 5) =    2.50  VERTEX NO   5  G= -.164023E+01

MU-MATRIX TO BE USED IN FOLLOWING OPTIMIZATION
  0 MORE VERTICES HAVE BEEN DETECTED IN UPDATING PROCEDURE
THE CURRENT NUMBER OF CONSTRAINTS IN PROBLEM ARE  22
NV( 1)= 1 NV( 2)= 3 NV( 3)= 4 NV( 4)= 7 NV( 5)= 1


 1  1  1 -1  1  1 -1 -1  1  1 -1  1  1 -1 -1 -1
-1 -1 -1 -1 -1 -1 -1 -1  1  1  1 -1 -1 -1 -1 -1
 1  1 -1  1  1 -1  1 -1  1 -1  1  1 -1  1 -1 -1
MU-MATRIX HAS REMAINED UNCHANGED IN TWO CONSECUTIVE CALLS FOR VERTST
THE PROGRAM GOES ON WITH THE FINAL OPTIMIZATION
```

Fig. 11 [Continued]

```
          FOLLOWING IS RESULT OF OPTIMIZATION
          -----------------------------------

                    NODE NUMBER =      0

     ACTUAL OBJECTIVE FUNCTION F =    .33354026E+02

    X( 1)=    .31460446E+00
    X( 2)=    .27579413E+00
    X( 3)=    .31460446E+00
    X( 4)=    .19992278E+01
    X( 5)=    .90563584E+00
    X( 6)=    .19992277E+01

    INEQUALITY CONSTRAINTS        OCCURRING AT

    G( 1)=    .21203482E-01       SAMPLE POINT 1
    G( 2)=    .11633554E-04       SAMPLE POINT 2
    G( 3)=    .17957767E+00       SAMPLE POINT 2
    G( 4)=    .17957788E+00       SAMPLE POINT 2
    G( 5)=    .44423348E-01       SAMPLE POINT 3
    G( 6)=    .18811532E+00       SAMPLE POINT 3
    G( 7)=    .18811533E+00       SAMPLE POINT 3
    G( 8)=    .40122591E+00       SAMPLE POINT 3
    G( 9)=    .35139453E-06       SAMPLE POINT 4
    G(10)=    .66981658E+00       SAMPLE POINT 4
    G(11)=    .66981663E+00       SAMPLE POINT 4
    G(12)=    .12550468E+01       SAMPLE POINT 4
    G(13)=    .13588466E+01       SAMPLE POINT 4
    G(14)=    .13588467E+01       SAMPLE POINT 4
    G(15)=    .14971984E+01       SAMPLE POINT 4
    G(16)=   -.45118520E-06       SAMPLE POINT 5
    G(17)=    .80135231E+00       LOWER BOUND  1
    G(18)=    .30289669E+00       UPPER BOUND  1
    G(19)=    .23675100E+00       LOWER BOUND  2
    G(20)=    .15254793E+01       UPPER BOUND  2
    G(21)=    .80135225E+00       LOWER BOUND  3
    G(22)=    .30289676E+00       UPPER BOUND  3

      NUMBER OF CONSTRAINTS USED =    22

   NUMBER OF VIOLATED CONSTRAINTS =     0

   NUMBER OF FUNCTION EVALUATIONS =    71

FINAL VALUE OF THE PARAMETER ALPHA =    .10000000E+03


          FOLLOWING IS THE OPTIMUM SOLUTION
          ---------------------------------
    Z( 1)=    .98975963E-01
    Z( 2)=    .76062400E-01
    Z( 3)=    .98975966E-01
    Z( 4)=    .19992278E+01
    Z( 5)=    .90563584E+00
    Z( 6)=    .19992277E+01

        EXECUTION TIME IN SECONDS =    4.96000
```

Fig. 12 Results of the continuous and discrete optimizations for Example 2

```
              OPTIMUM DISCRETE SOLUTION FOUND
                       MINIMUM   F =    .40000000E+02

        X(  1)=     .10000000E+00
        X(  2)=     .50000000E-01
        X(  3)=     .10000000E+00
        X(  4)=     .19992278E+01
        X(  5)=     .90563584E+00
        X(  6)=     .19992277E+01

        INEQUALITY CONSTRAINTS        OCCURRING AT

        G(  1)=     .70122022E-01     SAMPLE POINT 1
        G(  2)=     .60865763E-01     SAMPLE POINT 2
        G(  3)=     .23268614E+00     SAMPLE POINT 2
        G(  4)=     .23266616E+00     SAMPLE POINT 2
        G(  5)=     .11836619E+00     SAMPLE POINT 3
        G(  6)=     .25159133E+00     SAMPLE POINT 3
        G(  7)=     .25159135E+00     SAMPLE POINT 3
        G(  8)=     .45826866E+00     SAMPLE POINT 3
        G(  9)=     .25661869E+00     SAMPLE POINT 4
        G( 10)=     .84041227E+00     SAMPLE POINT 4
        G( 11)=     .84041234E+00     SAMPLE POINT 4
        G( 12)=     .10978658E+01     SAMPLE POINT 4
        G( 13)=     .12992456E+01     SAMPLE POINT 4
        G( 14)=     .12992457E+01     SAMPLE POINT 4
        G( 15)=     .14975881E+01     SAMPLE POINT 4
        G( 16)=     .27997158E+00     SAMPLE POINT 5
        G( 17)=     .79930503E+00     LOWER BOUND  1
        G( 18)=     .30084941E+00     UPPER BOUND  1
        G( 19)=     .26035405E+00     LOWER BOUND  2
        G( 20)=     .15490824E+01     UPPER BOUND  2
        G( 21)=     .79930497E+00     LOWER BOUND  3
        G( 22)=     .30084948E+00     UPPER BOUND  3
   NUMBER OF FUNCTION EVALUATIONS =    89


              FOLLOWING IS THE OPTIMUM SOLUTION
              ---------------------------------

        Z(  1)=     .10000000E+00
        Z(  2)=     .50000000E-01
        Z(  3)=     .10000000E+00
        Z(  4)=     .19992278E+01
        Z(  5)=     .90563584E+00
        Z(  6)=     .19992277E+01
```

Fig. 12 [Continued]

```
           DATA FROM VERTICES SELECTION NO  2
           ----------------------------------

AT FOLLOWING SAMPLE POINTS CONCAVITY VIOLATIONS HAVE BEEN INDICATED
          AT SAMPT(1, 2) =    .60
          AT SAMPT(1, 2) =    .60
          AT SAMPT(1, 3) =    .70
          AT SAMPT(1, 3) =    .70
          AT SAMPT(1, 4) =    .80
          AT SAMPT(1, 4) =    .80
          AT SAMPT(1, 5) =    .90
          AT SAMPT(1, 6) =   1.00
          AT SAMPT(1, 7) =   1.10
          AT SAMPT(1, 8) =   1.20
          AT SAMPT(1, 8) =   1.20
          AT SAMPT(1, 9) =   1.30
          AT SAMPT(1, 9) =   1.30
          AT SAMPT(1,10) =   1.40
          AT SAMPT(1,10) =   1.40

MU-MATRIX TO BE USED IN FOLLOWING OPTIMIZATION
 21 MORE VERTICES HAVE BEEN DETECTED IN UPDATING PROCEDURE
THE CURRENT NUMBER OF CONSTRAINTS IN PROBLEM ARE  36
NV( 1)= 1 NV( 2)= 4 NV( 3)= 4 NV( 4)= 4 NV( 5)= 2 NV( 6)= 2 NV( 7)= 2
NV( 8)= 4 NV( 9)= 4 NV(10)= 4 NV(11)= 1


-1 -1 -1  1  1  1 -1 -1  1  1 -1 -1  1  1 -1  1 -1  1 -1  1 -1 -1  1  1 -1
 1  1 -1 -1  1 -1 -1  1  1 -1 -1  1  1 -1 -1 -1 -1 -1 -1 -1 -1  1  1 -1 -1


-1  1 -1 -1  1  1 -1
 1  1  1 -1 -1  1  1
```

Fig. 13  Results for one of the updating procedures for Example 3

```
              FOLLOWING IS RESULT OF OPTIMIZATION
              -----------------------------------

                          NODE NUMBER =      0

          ACTUAL OBJECTIVE FUNCTION F =    .15690265E+02

         X( 1)=    .35702601E+00
         X( 2)=    .35702601E+00
         X( 3)=    .21486998E+01
         X( 4)=    .47308440E+01

         INEQUALITY CONSTRAINTS        OCCURRING AT

         G( 1)=    .37557513E-09       SAMPLE POINT 1
         G( 2)=    .16095318E+00       SAMPLE POINT 2
         G( 3)=    .27843044E+00       SAMPLE POINT 2
         G( 4)=    .48949641E+00       SAMPLE POINT 2
         G( 5)=    .27843044E+00       SAMPLE POINT 2
         G( 6)=    .32629272E+00       SAMPLE POINT 3
         G( 7)=    .43852556E+00       SAMPLE POINT 3
         G( 8)=    .34522595E+00       SAMPLE POINT 3
         G( 9)=    .43852557E+00       SAMPLE POINT 3
         G(10)=    .13225953E+00       SAMPLE POINT 4
         G(11)=    .33994274E+00       SAMPLE POINT 4
         G(12)=    .50995036E+00       SAMPLE POINT 4
         G(13)=    .33994274E+00       SAMPLE POINT 4
         G(14)=    .30728362E-01       SAMPLE POINT 5
         G(15)=    .23753313E+00       SAMPLE POINT 5
         G(16)=    .31685232E-09       SAMPLE POINT 6
         G(17)=    .20297994E+00       SAMPLE POINT 6
         G(18)=    .30728362E-01       SAMPLE POINT 7
         G(19)=    .23753313E+00       SAMPLE POINT 7
         G(20)=    .13225953E+00       SAMPLE POINT 8
         G(21)=    .33994274E+00       SAMPLE POINT 8
         G(22)=    .50995036E+00       SAMPLE POINT 8
         G(23)=    .33994274E+00       SAMPLE POINT 8
         G(24)=    .32629272E+00       SAMPLE POINT 9
         G(25)=    .43852556E+00       SAMPLE POINT 9
         G(26)=    .34522595E+00       SAMPLE POINT 9
         G(27)=    .43852557E+00       SAMPLE POINT 9
         G(28)=    .16095318E+00       SAMPLE POINT10
         G(29)=    .27843044E+00       SAMPLE POINT10
         G(30)=    .48949641E+00       SAMPLE POINT10
         G(31)=    .27843044E+00       SAMPLE POINT10
         G(32)=    .37558934E-09       SAMPLE POINT11
         G(33)=    .87481022E+00       LOWER BOUND  1
         G(34)=    .15774107E+01       UPPER BOUND  1
         G(35)=    .11278148E+01       LOWER BOUND  2
         G(36)=    .16661269E+01       UPPER BOUND  2

            NUMBER OF CONSTRAINTS USED =     36

        NUMBER OF VIOLATED CONSTRAINTS =      0

        NUMBER OF FUNCTION EVALUATIONS =     58

    FINAL VALUE OF THE PARAMETER ALPHA =    .10000000E+04


               FOLLOWING IS THE OPTIMUM SOLUTION
               ---------------------------------

          Z( 1)=    .12746757E+00
          Z( 2)=    .12746757E+00
          Z( 3)=    .21486998E+01
          Z( 4)=    .47308440E+01

            EXECUTION TIME IN SECONDS =    7.62100
```

Fig. 14 Continuous solution for Example 3

```
                    FOLLOWING IS THE OPTIMUM SOLUTION
                    ---------------------------------

                                NODE NUMBER =    1
          ACTUAL OBJECTIVE FUNCTION F =    .16462411E+02

         X( 1)=    .15474100E+00
         X( 2)=    .10000000E+00
         X( 3)=    .22180305E+01
         X( 4)=    .48489756E+01

         INEQUALITY CONSTRAINTS      OCCURRING AT

         G( 1)=    .74966167E-09     SAMPLE POINT 1
         G( 2)=    .16095318E+00     SAMPLE POINT 2
         G( 3)=    .25718188E+00     SAMPLE POINT 2
         G( 4)=    .46852210E+00     SAMPLE POINT 2
         G( 5)=    .29506929E+00     SAMPLE POINT 2
         G( 6)=    .32195248E+00     SAMPLE POINT 3
         G( 7)=    .45049599E+00     SAMPLE POINT 3
         G( 8)=    .34522595E+00     SAMPLE POINT 3
         G( 9)=    .40784752E+00     SAMPLE POINT 3
         G(10)=    .13132408E+00     SAMPLE POINT 4
         G(11)=    .39275967E+00     SAMPLE POINT 4
         G(12)=    .50995037E+00     SAMPLE POINT 4
         G(13)=    .28802719E+00     SAMPLE POINT 4
         G(14)=    .30565270E-01     SAMPLE POINT 5
         G(15)=    .28852866E+00     SAMPLE POINT 5
         G(16)=    .10196359E-08     SAMPLE POINT 6
         G(17)=    .25284784E+00     SAMPLE POINT 6
         G(18)=    .30565270E-01     SAMPLE POINT 7
         G(19)=    .28852866E+00     SAMPLE POINT 7
         G(20)=    .13132408E+00     SAMPLE POINT 8
         G(21)=    .39275967E+00     SAMPLE POINT 8
         G(22)=    .50995037E+00     SAMPLE POINT 8
         G(23)=    .28802719E+00     SAMPLE POINT 8
         G(24)=    .32195248E+00     SAMPLE POINT 9
         G(25)=    .45049599E+00     SAMPLE POINT 9
         G(26)=    .34522595E+00     SAMPLE POINT 9
         G(27)=    .40784752E+00     SAMPLE POINT 9
         G(28)=    .16095318E+00     SAMPLE POINT10
         G(29)=    .25718188E+00     SAMPLE POINT10
         G(30)=    .46852210E+00     SAMPLE POINT10
         G(31)=    .29506929E+00     SAMPLE POINT10
         G(32)=    .74967232E-09     SAMPLE POINT11
         G(33)=    .87481024E+00     LOWER BOUND  1
         G(34)=    .14387492E+01     UPPER BOUND  1
         G(35)=    .13640781E+01     LOWER BOUND  2
         G(36)=    .16661268E+01     UPPER BOUND  2

            NUMBER OF CONSTRAINTS USED =    36
        NUMBER OF VIOLATED CONSTRAINTS =     0
        NUMBER OF FUNCTION EVALUATIONS =    66
    FINAL VALUE OF THE PARAMETER ALPHA =   .10000000E+04
              EXECUTION TIME IN SECONDS =   8.53700
```

Fig. 15  Results at certain nodes in the discrete solution for Example 3

```
                 RESULTS AT LAST ITERATION
                 -------------------------

                        NODE NUMBER =     2
         ACTUAL OBJECTIVE FUNCTION F =    .16571981E+02

         X( 1)=    .15000000E+00
         X( 2)=    .10095591E+00
         X( 3)=    .22212193E+01
         X( 4)=    .48408048E+01

         INEQUALITY CONSTRAINTS      OCCURRING AT

         G( 1)=    .23001989E-02     SAMPLE POINT 1
         G( 2)=    .16499349E+00     SAMPLE POINT 2
         G( 3)=    .26344125E+00     SAMPLE POINT 2
         G( 4)=    .47031870E+00     SAMPLE POINT 2
         G( 5)=    .29505838E+00     SAMPLE POINT 2
         G( 6)=    .32229971E+00     SAMPLE POINT 3
         G( 7)=    .45580892E+00     SAMPLE POINT 3
         G( 8)=    .35109569E+00     SAMPLE POINT 3
         G( 9)=    .40972280E+00     SAMPLE POINT 3
         G(10)=    .13135500E+00     SAMPLE POINT 4
         G(11)=    .38511209E+00     SAMPLE POINT 4
         G(12)=    .51701184E+00     SAMPLE POINT 4
         G(13)=    .28991307E+00     SAMPLE POINT 4
         G(14)=    .30528585E-01     SAMPLE POINT 5
         G(15)=    .27971718E+00     SAMPLE POINT 5
         G(16)=   -.49983840E-04     SAMPLE POINT 6
         G(17)=    .24395837E+00     SAMPLE POINT 6
         G(18)=    .30528585E-01     SAMPLE POINT 7
         G(19)=    .27971718E+00     SAMPLE POINT 7
         G(20)=    .13135500E+00     SAMPLE POINT 8
         G(21)=    .38511209E+00     SAMPLE POINT 8
         G(22)=    .51701184E+00     SAMPLE POINT 8
         G(23)=    .28991307E+00     SAMPLE POINT 8
         G(24)=    .32229971E+00     SAMPLE POINT 9
         G(25)=    .45580892E+00     SAMPLE POINT 9
         G(26)=    .35109569E+00     SAMPLE POINT 9
         G(27)=    .40972280E+00     SAMPLE POINT 9
         G(28)=    .16499349E+00     SAMPLE POINT10
         G(29)=    .26344125E+00     SAMPLE POINT10
         G(30)=    .47031870E+00     SAMPLE POINT10
         G(31)=    .29505838E+00     SAMPLE POINT10
         G(32)=    .23001989E-02     SAMPLE POINT11
         G(33)=    .88803640E+00     LOWER BOUND  1
         G(34)=    .14455978E+01     UPPER BOUND  1
         G(35)=    .13520970E+01     LOWER BOUND  2
         G(36)=    .16704873E+01     UPPER BOUND  2

           NUMBER OF CONSTRAINTS USED =    37
        NUMBER OF VIOLATED CONSTRAINTS =     1
        NUMBER OF FUNCTION EVALUATIONS =    57
    FINAL VALUE OF THE PARAMETER ALPHA =    .10000000E+09
            EXECUTION TIME IN SECONDS =    8.57500
```

Fig. 15 [Continued]

```
                FOLLOWING IS THE OPTIMUM SOLUTION
                ---------------------------------

                          NODE NUMBER =     4

         ACTUAL OBJECTIVE FUNCTION F =    .16462274E+02

        X( 1)=     .10000000E+00
        X( 2)=     .15474428E+00
        X( 3)=     .20832190E+01
        X( 4)=     .46193291E+01

        INEQUALITY CONSTRAINTS       OCCURRING AT

        G( 1)=    -.13028112E-05     SAMPLE POINT 1
        G( 2)=     .16095089E+00     SAMPLE POINT 2
        G( 3)=     .29509583E+00     SAMPLE POINT 2
        G( 4)=     .46855302E+00     SAMPLE POINT 2
        G( 5)=     .25716429E+00     SAMPLE POINT 2
        G( 6)=     .32195750E+00     SAMPLE POINT 3
        G( 7)=     .40788304E+00     SAMPLE POINT 3
        G( 8)=     .34522262E+00     SAMPLE POINT 3
        G( 9)=     .45045664E+00     SAMPLE POINT 3
        G(10)=     .13132376E+00     SAMPLE POINT 4
        G(11)=     .28803467E+00     SAMPLE POINT 4
        G(12)=     .50994632E+00     SAMPLE POINT 4
        G(13)=     .39275117E+00     SAMPLE POINT 4
        G(14)=     .30563877E-01     SAMPLE POINT 5
        G(15)=     .18821132E+00     SAMPLE POINT 5
        G(16)=    -.15860206E-05     SAMPLE POINT 6
        G(17)=     .15499808E+00     SAMPLE POINT 6
        G(18)=     .30563877E-01     SAMPLE POINT 7
        G(19)=     .18821132E+00     SAMPLE POINT 7
        G(20)=     .13132376E+00     SAMPLE POINT 8
        G(21)=     .28803467E+00     SAMPLE POINT 8
        G(22)=     .50994632E+00     SAMPLE POINT 8
        G(23)=     .39275117E+00     SAMPLE POINT 8
        G(24)=     .32195750E+00     SAMPLE POINT 9
        G(25)=     .40788304E+00     SAMPLE POINT 9
        G(26)=     .34522262E+00     SAMPLE POINT 9
        G(27)=     .45045664E+00     SAMPLE POINT 9
        G(28)=     .16095089E+00     SAMPLE POINT10
        G(29)=     .29509583E+00     SAMPLE POINT10
        G(30)=     .46855302E+00     SAMPLE POINT10
        G(31)=     .25716429E+00     SAMPLE POINT10
        G(32)=    -.13028112E-05     SAMPLE POINT11
        G(33)=     .87489706E+00     LOWER BOUND  1
        G(34)=     .17084591E+01     UPPER BOUND  1
        G(35)=     .90451436E+00     LOWER BOUND  2
        G(36)=     .16658561E+01     UPPER BOUND  2

            NUMBER OF CONSTRAINTS USED =    37

        NUMBER OF VIOLATED CONSTRAINTS =     0

        NUMBER OF FUNCTION EVALUATIONS =    67

  FINAL VALUE OF THE PARAMETER ALPHA =    .10000000E+04

              EXECUTION TIME IN SECONDS =    8.76600
```

Fig. 15 [Continued]

SOC-105

# TOLOPT - A PROGRAM FOR OPTIMAL, CONTINUOUS OR DISCRETE, DESIGN CENTERING AND TOLERANCING PART 1 - USER'S GUIDE, PART II - FORTRAN LISTING

J.W. Bandler, J.H.K. Chen, P. Dalsgaard and P.C. Liu

September 1975, No. of Pages: Part I 47
Part II 27

Revised:

Key Words: Tolerancing, centering, continuous and discrete optimization, worst-case design

Abstract: This report describes the development, organization and implementation of a user-oriented computer program package called TOLOPT (TOLerance OPTimization), which can solve continuous and/or discrete worst-case tolerance assignment problems. Worst-case vertices can be automatically selected and optimization will lead to the most favorable nominal design simultaneously with the largest possible tolerances on specified toleranced components. The program contains recent techniques and algorithms for nonlinear programming. The optimization is carried out by subprograms substantially the same as ones in the DISOPT package. The full Fortran IV listing is included in this report as well as three circuit examples illustrating the use of and typical printouts from TOLOPT.

| | | |
|---|---|---|
| Description: | Part I | contains user's manual. |
| | Part II | contains Fortran listing. |
| | Source deck available for $300.00. | |

Related Work: Represents further development of work reported in IEEE Trans. Microwave Theory and Techniques, vol. MTT-23, Aug. 1975, pp. 630-641. As for SOC-1.

| | | |
|---|---|---|
| Price: | Part I | $15.00. |
| | Part II | $85.00. |