A COMPARISON OF RECENTLY IMPLEMENTED OPTIMIZATION TECHNIQUES

J.W. Bandler, W. Kellermann and W.M. Zuberek

SOS-83-8-R

July 1983

A COMPARISON OF RECENTLY IMPLEMENTED OPTIMIZATION TECHNIQUES

J.W. Bandler, W. Kellermann and W.M. Zuberek*

Simulation Optimization Systems Research Laboratory
and Department of Electrical and Computer Engineering
McMaster University, Hamilton, Canada L8S 4L7

Abstract

This paper reviews the practical implementation of recent optimization techniques and their application in the area of electrical circuit design. The discussion is focussed on four nonlinear programming codes including unconstrained minimax optimization, linearly constrained minimax optimization and optimization with general constraints. A brief introduction to the optimization methods used in these codes is presented. Practical examples illustrate the formulation of design problems in terms of mathematical programming problems as well as the performance of the optimization codes presented. The discussion includes a comparison of features of the packages from the user-designer point of view.

1. Introduction

This paper reviews the practical implementation of recent optimization techniques and their application in the area of electrical circuit design. The discussion is focussed on four nonlinear programming codes.

The MMUM package [1] solves unconstrained minimax optimization problems and is based on the method described by Hald and Madsen [2]. It is an extension and modification of the MINI5W package due to Madsen [3]. The MMLC package [4] solves linearly constrained minimax optimization problems and is based on the method described by Hald and Madsen [2]. It is an extension and modification of the MMLA1Q package due to Hald [5]. In both packages first derivatives of all functions with respect to all variables are assumed to be known. The solution is found by an iteration that uses either linear programming applied in connection with first-order derivatives or a quasi-Newton method applied in connection with first-order and approximate second-order derivatives.

MFNC [6] is a package for minimization of a nonlinear objective function subject to nonlinear constraints. It is an extension and modification of a set of subroutines from the Harwell Subroutine Library [7]. The method implemented was presented by Han [8] and Powell [9]. First derivatives of all functions with respect to all variables are assumed to be available. The solution is found by an iteration that minimizes a quadratic approximation of the objective function subject to linearized constraints.

The MINOS/AUGMENTED system [10] is a general purpose programming system to solve large-scale optimization problems involving sparse linear and nonlinear constraints. Any nonlinear functions appearing in the objective or the constraints must be continuous and smooth. MINOS/AUGMENTED employs a projected augmented Lagrangian algorithm to solve problems with nonlinear constraints presented by Murtagh and Saunders [11]. This involves a sequence of sparse, linearly constrained subproblems, which are solved by a reduced-gradient algorithm.

A wide variety of test problems for the comparison of different nonlinear programming algorithms and their practical implementation exist in the literature. Generally, they fall into two categories. One category consists of nonlinear programming problems where the objective function and constraints are given explicitly in the form of a mathematical formulation, e.g., the Colville series of problems [12], the Wang family of problems [13], the Rosen-Suzuki problem [14] and the Rosenbrock problem [15]. Usually they are designed to test the performance of algorithms under difficult conditions such as narrow valleys, numerical singularities, etc. As a representative for this category the Colville test problem 2 has been chosen.

The second category of test problems includes practical engineering design problems where the formulation of the problem is not explicitly given in terms of the objective function and constraints, and different formulations may exist which adequately represent the engineering design problem. A three-section 100 percent relative-bandwidth 10:1 transmission-line transformer is an example. It is a special case of an N-section transmission-line transformer. Originally studied by Bandler and developed into a family of test problems by Bandler and Macdonald [16,17] this type of test problem is now widely considered [18-22]. Another example of the second category is the optimal design of a LC low-pass filter embodying centering, tolerancing and tuning developed by Bandler, Liu and Tromp [23-24].

2. Nonlinear Programming Problem and Minimax Problem

The nonlinear programming problem can be stated as follows

$$\text{minimize } U(\underset{\sim}{x}) \tag{1}$$

subject to

$$g_i(\underset{\sim}{x}) \geq 0 \quad , \quad i=1,2,..,m, \tag{2}$$

where U is the generally nonlinear objective function of n variables $\underset{\sim}{x}$, where

---

This work was supported by the Natural Sciences and Engineering Research Council of Canada under Grants A7239 and G0647.

* W.M. Zuberek is on leave from the Institute of Computer Science, Technical University of Warsaw, Warsaw, Poland. He is now with the Department of Electrical Engineering, Texas A&M University, College Station, TX 77843.

$$\underset{\sim}{x} \triangleq \begin{bmatrix} x_1 \\ x_2 \\ \cdot \\ \cdot \\ \cdot \\ x_n \end{bmatrix}, \tag{3}$$

and $g_1(\underset{\sim}{x})$, $g_2(\underset{\sim}{x})$, ..., $g_m(\underset{\sim}{x})$ are, in general, nonlinear functions of the variables. We will assume that all the functions are continuous with continuous partial derivatives, and that the inequality constraints $g_i(\underset{\sim}{x}) \geq 0$, $i=1,2,\ldots,m$, are such that a Kuhn-Tucker solution exists [25].

The equivalent minimax problem can be formulated using the Bandler and Charalambous approach [26]. Consider the problem of minimizing the unconstrained function

$$F(\underset{\sim}{x},\underset{\sim}{\alpha}) = \max_{1\leq i\leq m+1} f_i(\underset{\sim}{x}), \tag{4}$$

where

$$f_i(\underset{\sim}{x}) = U(\underset{\sim}{x}) - \alpha_i\, g_i(\underset{\sim}{x}), \quad i=1,\ldots,m, \tag{5}$$

$$f_{m+1}(\underset{\sim}{x}) = U(\underset{\sim}{x}), \tag{6}$$

$$\underset{\sim}{\alpha} \triangleq [\alpha_1\ \alpha_2\ \ldots\ \alpha_m]^T, \tag{7}$$

$$\alpha_i > 0, \quad i=1,2,\ldots,m. \tag{8}$$

Bandler and Charalambous [26] proved that for sufficiently large α_i the optimum of the minimax function coincides with that of the nonlinear programming problem.

The general minimax optimization problem can be stated as

$$\underset{\underset{\sim}{x}}{\text{minimize}}\ \{F(\underset{\sim}{x}) = \max_{1\leq i\leq m} f_i(\underset{\sim}{x})\}, \tag{9}$$

subject to

$$g_j(\underset{\sim}{x}) \geq 0, \quad j=m+1,\ldots,\ell, \tag{10}$$

where $\underset{\sim}{x}$ is a vector of optimization variables defined as in (3), $f_1(\underset{\sim}{x})$, $f_2(\underset{\sim}{x})$, ..., $f_m(\underset{\sim}{x})$ are, in general, nonlinear functions with respect to variables x_1, x_2, ..., x_n and $g_j(\underset{\sim}{x})$ are, in general, nonlinear constraints.

The minimax problem is equivalent to the following nonlinear programming problem [27]

$$\underset{\underset{\sim}{x},x_{n+1}}{\text{minimize}}\ F(\underset{\sim}{x}, x_{n+1}) = x_{n+1} \tag{11}$$

subject to the constraints of the form

$$g_i(\underset{\sim}{x},x_{n+1}) \geq 0, \quad i=1,\ldots,m,m+1,\ldots,\ell, \tag{12}$$

where

$$g_i(\underset{\sim}{x},x_{n+1}) \triangleq x_{n+1} - f_i(\underset{\sim}{x}) \geq 0, \; i=1,2,\ldots,m. \tag{13}$$

Now the nonlinear programming problem can be solved by any nonlinear programming algorithm, thus obtaining the optimum minimax solution.

3. Unconstrained Minimax Optimization (the MMUM Package [1])

Given a set of nonlinear differentiable residual functions $f_i(\underset{\sim}{x})$, $i=1,2,\ldots,m$, of n variables $\underset{\sim}{x} = [x_1\ x_2\ \ldots\ x_n]^T$, it is the purpose of the package to find a local minimum of the minimax objective function

$$F(\underset{\sim}{x}) = \max_{1\leq i\leq m} |f_i(\underset{\sim}{x})|. \tag{14}$$

The objective is, in general, a non-differentiable function and normally the minimum is situated at a point where two or more residual functions are equal. If there is no smooth valley through the solution and the minimum is numerically well-defined then the minimum is characterized by only first derivatives of the residual functions which determine it. For such cases it is possible to construct algorithms based on first derivative information only with fast final convergence. It has been proved [2,28] that if the so-called Haar condition (which ensures that no smooth valley passes through the solution) is satisfied then quadratic final rate of convergence can be obtained. If there is, however, a smooth valley through the solution, the first-order derivations may be insufficient and some second-order information may be needed to obtain a fast final convergence. For such cases the quasi-Newton iteration has been proposed [2] in which the second-order derivatives are approximated by the Powell method.

The minimax algorithm is a two-stage one [2]. Normally, Stage 1 is used [29], and at each point the nonlinear residual functions are approximated by linear functions using the first derivative information. However, if a smooth valley through the solution is detected, a switch to Stage 2 is made and the quasi-Newton iteration is used. If it turns out that the Stage 2 iteration is unsuccessful (for instance, if the set of active functions has been wrongly chosen) then a switch is made back to Stage 1. The algorithm may switch several times between Stage 1 and Stage 2 but normally only a few switches will take place and the iteration will terminate either in Stage 1 with quadratic rate of convergence or in Stage 2 with superlinear rate of convergence [2].

4. Linearly Constrained Minimax Optimization (the MMLC Package [4])

Given a set of nonlinear differentiable residual functions $f_i(\underset{\sim}{x})$, $i=1,2,\ldots,m$, of n variables $\underset{\sim}{x} = [x_1\ x_2\ \ldots\ x_n]^T$, it is the purpose of the package to find a local minimum of the minimax objective function

$$F(\underset{\sim}{x}) = \max_{1\leq i\leq m} f_i(\underset{\sim}{x}) \tag{15}$$

subject to linear constraints

$$\underset{\sim}{c}_i^T\,\underset{\sim}{x} + b_i = 0, \quad i=1,\ldots,\ell_{eq}, \tag{16}$$

$$\underset{\sim}{c}_i^T\,\underset{\sim}{x} + b_i \geq 0, \quad i=\ell_{eq}+1,\ldots,\ell, \tag{17}$$

where $\underset{\sim}{c}_i$ and b_i, $i=1,\ldots,\ell$, are constants.

The algorithm employed is substantially similar to that described in Section 3. The algorithm is a feasible point algorithm which means that the residual functions are only evaluated at points satisfying the linear constraints. Initially a feasible point is determined by the package, and from that point feasibility is retained.

5. Han-Powell Algorithm (the MFNC Package [6])

The purpose of the package is to minimize the objective function $F(\underset{\sim}{x})$ of n variables, $\underset{\sim}{x} = [x_1\ x_2\ \dots\ x_n]^T$, subject to general equality and inequality constraints

$$f_j(\underset{\sim}{x}) = 0\ , \quad j=1,\dots,\ell_{eq}\ , \qquad (18)$$

$$f_j(\underset{\sim}{x}) \geq 0\ , \quad j=\ell_{eq}+1,\dots,\ell\ , \qquad (19)$$

where the objective and the constraint functions are differentiable and their first-order derivatives are available.

The algorithm used in the package is Powell's [9,30] variable metric method for constrained optimization, which is based on the results of Han [8]. In each kth iteration the search direction $\underset{\sim}{h}^k$ is determined as the solution of the linearly constrained quadratic minimization subproblem

$$\underset{\underset{\sim}{h}^k}{\text{minimize}}\ \tilde{F}(\underset{\sim}{x}^{k-1}, \underset{\sim}{h}^k) = F(\underset{\sim}{x}^{k-1}) + \underset{\sim}{h}^{k^T} \underset{\sim}{F}'(\underset{\sim}{x}^{k-1}) + 0.5\ \underset{\sim}{h}^{k^T} \underset{\sim}{B}^k \underset{\sim}{h}^k \qquad (20)$$

subject to the constraints

$$\underset{\sim}{h}^{k^T} \underset{\sim}{f}_j'(\underset{\sim}{x}^{k-1}) + \alpha^k f_j(\underset{\sim}{x}^{k-1}) = 0\ , \quad j=1,\dots,\ell_{eq}, \qquad (21)$$

$$\underset{\sim}{h}^{k^T} \underset{\sim}{f}_j'(\underset{\sim}{x}^{k-1}) + \alpha_j^k f_j(\underset{\sim}{x}^{k-1}) \geq 0\ , \quad j=\ell_{eq}+1,\dots,\ell, \qquad (22)$$

$$0 \leq \alpha^k \leq 1\ , \qquad (23)$$

where $\underset{\sim}{F}'(\underset{\sim}{x})$ and $\underset{\sim}{f}_j'(\underset{\sim}{x})$, $j=1,\dots,\ell$, are the gradient vectors of the objective and constraint functions, respectively, $\underset{\sim}{B}^k$ is a positive definite square matrix of dimension n containing second-order derivative information, which is updated in consecutive iterations according to the BFGS formula (initially the matrix is set to the unit matrix, $\underset{\sim}{B}^0 = \underset{\sim}{1}$), and α^k is an additional variable introduced in order to allow infeasibility in linearized constraints, while α_j^k, $j=\ell_{eq}+1,\dots,\ell$, are defined as

$$\alpha_j^k = \begin{cases} 1\ , & \text{if } f_j(\underset{\sim}{x}^{k-1}) > 0\ , \\ \alpha^k, & \text{if } f_j(\underset{\sim}{x}^{k-1}) \leq 0\ . \end{cases} \qquad (24)$$

Usually the solution of the quadratic subproblem results in $\alpha^k=1$. If the only feasible solution corresponds to $\alpha^k=0$ and $\underset{\sim}{h}^k = \underset{\sim}{0}$, the algorithm terminates and it is assumed that the constraints are inconsistent.

6. Augmented Lagrangian (the MINOS/AUGMENTED System [10])

The problem to be solved must be expressed in the following standard form [10]

$$\text{minimize } f_0(\underset{\sim}{x}) + \underset{\sim}{c}^T\underset{\sim}{x} + \underset{\sim}{d}^T\underset{\sim}{y} \qquad (25)$$

subject to

$$\underset{\sim}{f}(\underset{\sim}{x}) + \underset{\sim}{A}_1\underset{\sim}{y} = \underset{\sim}{b}_1\ , \qquad (26)$$

$$\underset{\sim}{A}_2\underset{\sim}{x} + \underset{\sim}{A}_3\underset{\sim}{y} = \underset{\sim}{b}_2\ , \qquad (27)$$

$$\underset{\sim}{\ell} \leq \begin{bmatrix} \underset{\sim}{x} \\ \underset{\sim}{y} \end{bmatrix} \leq \underset{\sim}{u}\ , \qquad (28)$$

where

$$\underset{\sim}{f}(\underset{\sim}{x}) = \begin{bmatrix} f_1(\underset{\sim}{x}) \\ \cdot \\ \cdot \\ \cdot \\ f_m(\underset{\sim}{x}) \end{bmatrix}$$

and the functions $f_i(\underset{\sim}{x})$ are smooth and have known gradients. The components of $\underset{\sim}{x}$ are called the nonlinear variables, and they must be the first set of unknowns. Similarly, constraints (26) are called the nonlinear constraints and they must appear before the linear constraints (27).

All types of inequality are allowed in the general constraints. Thus, the "=" sign in (26) and (27) may mean "≤" or "≥" or "free" for individual rows.

Upper and lower bounds (28) may be specified for all variables, and similar bounds (ranges) may be defined for the general constraints.

The solution process [31], [32] consists of a sequence of "major iterations". At the start of each major iteration, the nonlinear constraints are linearized at the current point $\underset{\sim}{x}_k$. This just means that $\underset{\sim}{f}(\underset{\sim}{x})$ in equation (26) is replaced by the approximation

$$\tilde{\underset{\sim}{f}}(\underset{\sim}{x},\underset{\sim}{x}_k) = \underset{\sim}{f}(\underset{\sim}{x}_k) + \underset{\sim}{J}(\underset{\sim}{x}_k)\ (\underset{\sim}{x}-\underset{\sim}{x}_k)\ , \qquad (29)$$

which can be written as

$$\tilde{\underset{\sim}{f}} = \underset{\sim}{f}_k + \underset{\sim}{J}_k\ (\underset{\sim}{x}-\underset{\sim}{x}_k)\ . \qquad (30)$$

Here, $\underset{\sim}{J}(\underset{\sim}{x})$ is the Jacobian matrix whose ij-th element is $\partial f_i(\underset{\sim}{x})/\partial x_j$.

The objective function is also modified, giving the following subproblem:

$$\text{minimize } f_0(\underset{\sim}{x}) + \underset{\sim}{c}^T\underset{\sim}{x} + \underset{\sim}{d}^T\underset{\sim}{y} - \underset{\sim}{\lambda}_k^T(\underset{\sim}{f}-\tilde{\underset{\sim}{f}}) + 0.5\ \rho\ (\underset{\sim}{f}-\tilde{\underset{\sim}{f}})^T\ (\underset{\sim}{f}-\tilde{\underset{\sim}{f}}) \qquad (31)$$

subject to

$$\tilde{f} + A_1 y = b_1 , \quad (32)$$

$$A_2 x + A_3 y = b_2 , \quad (33)$$

$$\ell \le \begin{bmatrix} x \\ y \end{bmatrix} \le u . \quad (34)$$

The objective function (31) is called an augmented Lagrangian. The vector λ_k is an estimate of the Lagrange multipliers for the nonlinear constraints, are the term involving ρ is a modified quadratic penalty function.

Using (30), we can see that the linear constraints (32) and (33) take the form

$$\begin{bmatrix} J_k & A_1 \\ A_2 & A_3 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} b_1 + J_k x_k - f_k \\ b_2 \end{bmatrix} . \quad (35)$$

Since MINOS takes advantage of sparsity within the constraint matrix, it is clear that a sparse Jacobian matrix J_k can be handled efficiently.

7. Test Problems

7.1 Test Problem 1

This is the design of a 3-section 100-percent relative bandwidth 10:1 transmission-line transformer [16]. The problem is to minimize the maximum reflection coefficient of this matching network. A detailed discussion on the formulation of direct minimax response objectives is presented in [33]. Formally, the problem is to reach

$$\min_{x} F(x) = \min_{x} \left\{ \max_{[0.5,\ 1.5]} |\rho(x,\psi)| \right\} , \quad (36)$$

where

$$x = [\ell_1/\ell_q \ Z_1 \ \ell_2/\ell_q \ Z_2 \ \ell_3/\ell_q \ Z_3]^T .$$

The error functions represent the modulus of the reflection coefficient sampled at the 11 normalized frequencies ψ (w.r.t. 1 GHz) {0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.3, 1.4, 1.5}. The known quarter-wave solution is given by

$$\ell_1 = \ell_2 = \ell_3 = \ell_q ,$$

$$Z_1 = 1.63471 ,$$

$$Z_2 = 3.16228 ,$$

$$Z_3 = 6.11729 ,$$

where ℓ_q is the quarter wavelength at the center frequency, namely,

$$\ell_q = 7.49481 \text{ cm for 1 GHz} .$$

The corresponding maximum reflection coefficient is 0.19729. The starting point is

$$x^0 = [0.8 \ \ 1.5 \ \ 1.2 \ \ 3.0 \ \ 0.8 \ \ 6.0]^T .$$

Gradient vectors with respect to section lengths and characteristic impedances are obtained using the adjoint network method.

7.2. Test Problem 2

This is the Colville test problem 2 [12] in the form used in [34]. It is to minimize the objective function

$$F(x) = -\sum_{1\le i\le 10} b_i x_{5+i} + \sum_{1\le i\le 5} \sum_{1\le j\le 5} c_{ij} x_i x_j + 2 \sum_{1\le j\le 5} d_j x_j^3 \quad (37)$$

subject to the constraints

$$x_i \ge 0 , \quad i=1, \ldots, 15 , \quad (38)$$

$$e_j - \sum_{1\le i\le 10} a_{ij} x_{5+i} + 2 \sum_{1\le i\le 5} c_{ij} x_i + 3 d_j x_j^2 \ge 0, \ j=1,\ldots,5 , \quad (39)$$

where a_{ij}, b_i, c_{ij}, d_j, e_j are given in Table I. The solution is $F(x^*) = 32.34868$. The feasible starting point used is $x_i^0 = 0.0001$, $i \ne 12$, and $x_{12} = 60.0$.

7.3 Test Problem 3

This is the design of a LC low-pass filter [23-24]. The problem is the optimal worst case design embodying centering, tolerancing and tuning at the design stage. A detailed discussion on the formulation is presented in [23]. If the designer has no prior knowledge of the choice of the tuning components we consider an objective function of the form

$$C = \sum_{i=1}^{3} \left[\frac{\phi_i^0}{\varepsilon_i} + c_i \frac{t_i}{\phi_i^0} \right] , \quad (40)$$

where ϕ_i^0, ε_i and t_i represent nominal values, tolerances and tuning parameters of components, respectively. The performance constraints may be written in the form

$$g = w(S-F) , \quad (41)$$

where w is +1 if S is an upper specification or -1 if S is a lower specification. F is the circuit response function evaluated at sample frequency ψ. Table II summarizes the specifications. The critical vertices used can be obtained from published vertex selection schemes [24]. Table III summarizes the data for the filter. There are 21 variables including nominal values, tolerances and tuning parameters as well as slack variables ρ which represent the settings of tuning components and 43 constraints including

TABLE I
DATA FOR TEST PROBLEM 2
(COLVILLE'S)

a_{ij} (i \ j)	1	2	3	4	5	b_i
1	-16	2	0	1	0	-40
2	0	-2	0	0.4	2	-2
3	-3.5	0	2	0	0	-.25
4	0	-2	0	-4	-1	-4
5	0	-9	-2	1	-2.8	-4
6	2	0	-4	0	0	-1
7	-1	-1	-1	-1	-1	-40
8	-1	-2	-3	-2	-1	-60
9	1	2	3	4	5	5
10	1	1	1	1	1	1

c_{ij} (i \ j)	1	2	3	4	5
1	30	-20	-10	32	-10
2	-20	39	-6	-31	32
3	-10	-6	10	-6	-10
4	32	-31	-6	39	-20
5	-10	32	-10	-20	30
d_j	4	8	10	6	2
e_j	-15	-27	-36	-18	-12

TABLE II
SPECIFICATIONS FOR LC LOW-PASS FILTER

Frequency Range (rad/s)	Sample Points (rad/s)	Insertion Loss Specification (dB)	Type	Weight w
0-1	0.45, 0.50, 0.55, 1.0	1.5	upper	+1
2.5	2.5	25	lower	-1

performance constraints and additional constraints on variables. The starting point is the solution for c_i=10 given in [23].

TABLE III
DATA FOR LOW-PASS FILTER

$\underset{\sim}{a}$ \ i	1	2	3	4	5	6	7	8	9	10
r	6	6	6	8	1	3	3	3	3	3
	+1	+1	+1	+1	-1	-1	-1	-1	-1	-1
$\underset{\sim}{\mu}$	-1	-1	-1	+1	-1	+1	+1	+1	+1	+1
	+1	+1	+1	+1	-1	-1	-1	-1	-1	-1
ψ	0.45	0.50	0.55	1.0	2.5	0.45	0.50	0.55	1.0	2.5
S	1.5	1.5	1.5	1.5	25	1.5	1.5	1.5	1.5	25
w	1	1	1	1	-1	1	1	1	1	-1

8. Discussion of Results

To evaluate different nonlinear programming techniques one should examine first the question of what criteria to use in the evaluation. Specifically, the following criteria can be used [35]:

1) time required in a series of tests (execution time and/or number of functional evaluations);

2) size (dimensionality, number of inequality constraints, number of equality constraints) of the problem;

3) accuracy of the solution with respect to the optimal vector $\underset{\sim}{x}^*$ and/or with respect to the objective function or constraints;

4) simplicity of use (time required to introduce data and functions into the computer program);

5) simplicity of computer program to execute the algorithm.

These criteria are global rather than local in the sense that they relate to the overall performance of the optimization from start to end rather than to the performance at a single stage.

The most common criteria used to evaluate the relative effectiveness of programming codes have been

(1) the number of function evaluations required to obtain the optimal solution of a given test problem to a given degree of precision and/or

(2) the computation time required to reach the solution of the given test problem.

The number of function evaluations is a less meaningful criterion for large constrained problems of several variables because the time required by the algorithm to determine the point at which to evaluate the functions can often be several times greater than that required for the evaluation of functions. Thus, computation time is the most commonly used criterion for comparing the effectiveness of different programming algorithms.

The first and most important consideration in comparing the effectiveness of the various algorithms is the success or failure of a given code to solve a given test problem. This criterion is chosen because the ability of an algorithm to solve a wide variety of problems is the most valuable feature to the user of a programming code. All four packages succeeded in solving the test problems. Results of an optimization of a three-section microwave transformer are summarized in Table IV. Both minimax packages seem to be more effective since this is originally a minimax problem, however the MFNC package requires the least number of function evaluations. Table V shows $\log_{10}\ [\max_{1\le i\le m} (|\rho_i| - F^*)]$ versus the number of function evaluations. This kind of comparison is useful when the user is interested in the solution with the accuracy acceptable from the practical point of view and the question is after how many function evaluations this accuracy can be obtained. It should be noted, however, that the comparison shown in Table V does not take time into account. Another important aspect in comparing optimization codes is by how much the constraints are violated. Usually, the packages find the solution satisfying the constraints with a certain accuracy, which in most cases is acceptable for practical purposes.

Table VI shows the results for the Colville test problem 2. In all cases the problem has been programmed to take advantage of all the features of the package. For this problem in two cases, namely, MMLC and MINOS, the linear constraints can be treated explicitly, either by means of the coefficients matrix (MMLC) or MPS file and the BOUNDS section (MINOS). In MINOS, moreover, the linear part of the objective function can be accommodated by means of the MPS file. In the case of the MMUM and MMLC packages the equivalent minimax formulation of the problem has been used with $\alpha = 10^3$ and $\alpha = 10$, respectively. Moreover, for MMUM, another technique has been used to avoid the undesired effects of transformed constraints on the minimax optimization, due to the absolute value operator in the objective function. The residual functions are forced to be non-negative by adding a constant c to the original objective function of the problem ($c = 10^5$ was used). Table VII shows $\log_{10}\ [\max_{1\le i\le m} (f_i - F^*)]$ versus the number of function evaluations. A minimax kind of measure was used for the originally non-minimax problem to take into account violated constraints in evaluation of the performance of the packages. Since the problem contains a substantial linear part, packages which can distinguish linear constraints are more efficient than those which assume only nonlinear functions.

Tables VIII and IX summarize the results for the LC low-pass filter problem. For this problem the choice of cost coefficient c_i in (40) for tuning is very important. The most appropriate choice is the one for which both terms in the objective function (40) have the same order of magnitude. The advantage gained in the formulation used is that the optimization will automatically choose the most appropriate

TABLE IV
OPTIMIZATION OF A 3-SECTION 10:1 TRANSFORMER OVER A 100 PERCENT BANDWIDTH

Variable	MMUM	MMLC	MFNC ‡	MINOS ‡
1	1.00000	1.00000	1.00000	1.00000
2	1.63471	1.63471	1.63471	1.63471
3	1.00000	1.00000	1.00000	1.00000
4	3.16228	3.16228	3.16228	3.16228
5	1.00000	1.00000	1.00000	1.00000
6	6.11730	6.11730	6.11731	6.11731
7			0.19729	0.19729
Minimax Function Value	a	a	b	a
Number of Function Evaluations ‡‡	18 (14)	22 (18)	13 (12)	179 (176)
Time(s) ‡‡‡	0.8	0.8	3.6	6.8

‡ In both cases the equivalent formulation of the minimax problem was used, so the number of variables is increased by one.
‡‡ In brackets is shown the number of function evaluations to reach 0.19729.
‡‡‡ Execution time (seconds) on CYBER 170/730.

a 0.1972906269 b 0.1972906258

TABLE V
COMPARISON OF OPTIMIZATION CODES FOR TEST PROBLEM 1

Function Evaluation Number	$\log_{10}\ [\max_i (|\rho_i| - F^*)]$			
	MMUM	MMLC	MFNC	MINOS ‡
1	-0.7	-0.7	-0.7	-1.3
2	-1.2	-0.9	-1.5	-1.3
3	-1.1	-1.5	-2.0	-1.3
4	-2.3	-1.2	-2.9	-1.3
5	-1.1	-2.7	-3.1	-1.3
6	-1.4	-1.5	-3.4	-1.3
7	-1.9	-2.6	-3.5	-1.7
8	-2.4	-3.0	-3.6	-1.5
9	-2.3	-3.1	-4.2	-1.6
10	-3.1	-3.1	-4.6	-1.6
11	-3.2	-3.0	-5.6	-1.7
12	-3.5	-3.3	-6.5	-1.9
13	-3.8	-3.4	-8.4	-1.9
14	-5.9	-3.7		-3.0
15	-6.9	-3.5		-3.0
16	-9.5	-3.7		-3.8
17	-10.6	-4.7		-5.2
18	-10.6	-5.7		-8.5
19		-6.9		
20		-8.9		
21		-10.5		
22		-10.6		

‡ For MINOS the number of function evaluations should be multiplied by 10, hence 1 corresponds to the 10th evaluation.

TABLE VI
RESULTS FOR COLVILLE'S TEST PROBLEM 2

Variable	MMUM‡	MMLC	MFNC	MINOS
1	0.30738	0.30000	0.29999	0.30000
2	0.33090	0.33347	0.33346	0.33347
3	0.41243	0.40000	0.39999	0.40000
4	0.42087	0.42831	0.42831	0.42831
5	0.22328	0.22396	0.22397	0.22396
6	-4.1×10^{-13}	0.0	9.1×10^{-16}	0.00000
7	-2.1×10^{-13}	0.0	1.1×10^{-14}	0.00000
8	5.05312	5.17404	5.17413	5.17404
9	-1.6×10^{-13}	0.0	0.0	0.00000
10	3.07959	3.06111	3.06111	3.06111
11	11.59737	11.83955	11.83972	11.83955
12	-2.5×10^{-13}	0.0	0.0	0.00000
13	-1.8×10^{-14}	0.0	0.0	0.00000
14	0.05312	0.10390	0.10393	0.10390
15	-2.6×10^{-13}	0.0	9.7×10^{-15}	0.00000
Objective Function Value	32.35284	a	b	a
Number of Function Evaluations‡‡	150	30 (28)	16 (16)	207 (obj.) 216 (con.)
Time(s)‡‡‡	41.5	3.1	13.2	3.2

‡ The optimization process has been stopped by imposing the limit on function evaluations.

‡‡ In brackets is shown the number of function evaluations to reach 32.34868.

‡‡‡ Execution time (seconds) on CYBER 170/730.

a 32.3486789657 b 32.3486790660

TABLE VII
COMPARISON OF OPTIMIZATION CODES FOR COLVILLE'S TEST PROBLEM 2

Function Evaluation Number	$\log_{10}[\max_i(f_i - F^*)]$ MMUM‡	MMLC	MFNC
1	3.4	3.4	3.4
2	3.3	3.4	3.4
3	3.3	3.4	2.9
4	3.3	3.4	2.6
5	3.3	3.4	1.8
6	3.1	3.4	1.3
7	3.0	3.3	1.0
8	2.9	3.3	0.3
9	2.7	3.1	0.2
10	2.4	2.9	-0.3
11	0.6	2.0	-0.4
12	-1.4	2.7	-1.5
13	-1.8	1.8	-2.0
14	-2.1	1.6	-3.8
15	-2.3	0.6	-4.5
16	-2.5	2.6	-5.1
17	-2.7	0.7	
18	-2.8	0.4	
19	-3.0	0.3	
20	-3.0	0.0	
21		-0.2	
22		-0.2	
23		-1.0	
24		-1.9	
25		-2.4	
26		-3.5	
27		-5.1	
28		-7.3	
29		-9.1	
30		-10.7	

‡ For MMUM the number of function evaluations should be multiplied by 10, hence 1 corresponds to the 10th evaluation.

component for tuning, which is the capacitor here. The observed discrepancies in values of slack variables ρ_1 and ρ_3 are insignificant since they correspond to settings of tuning parameters for which tuning is zero.

9. Conclusions

The packages presented can be used for solving a wide range of practical engineering design problems, however, a proper choice of the package for the particular problem is important and can result in major savings of time required to solve the problem. They can handle efficiently big problems (especially MINOS with the option of taking sparsity of the problem into account). The factor of human error in supplying analytical derivatives is eliminated by the gradient check option, which all of them have. The computer programs to execute the algorithms are simple and easy to write. The data preparation for the codes by the user is easy, with the exception of MINOS for which creating data files may be time consuming when a commercial matrix generator is not available. We feel that the optimization techniques presented and their implementation are powerful tools for solving difficult electrical circuit design problems.

10. References

[1] J.W. Bandler and W.M. Zuberek, "MMUM - a Fortran package for unconstrained minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-291, 1982.

TABLE VIII
OPTIMAL TUNING DESIGN OF THE LC LOW-PASS FILTER FOR c_i = 50

Solution	MMLC	MFNC	MINOS
$L_1^0 = L_2^0$	2.06696	2.06696	2.06696
C^0	0.90758	0.90758	0.90758
$100\ \varepsilon_1/L_1^0=100\ \varepsilon_3/L_2^0$	18.01%	18.01%	18.01%
$100\ \varepsilon_2/C^0$	14.14%	14.14%	14.14%
$100\ t_1/L_1^0=100\ t_3/L_2^0$	0.00%	0.00%	0.00%
$100\ t_2/C^0$	16.43%	16.43%	16.43%
$\rho_1(6)$	-1.00000	-0.99823	-1.00000
$\rho_2(6)$	1.00000	1.00000	1.00000
$\rho_3(6)$	-1.00000	-0.99823	1.00000
$\rho_1(8)$	-0.99935	-0.99255	-1.00000
$\rho_2(8)$	-1.00000	-1.00000	-1.00000
$\rho_3(8)$	-0.99935	-0.99255	-1.00000
$\rho_1(1)$	1.00000	0.99308	-1.00000
$\rho_2(1)$	1.00000	1.00000	1.00000
$\rho_3(1)$	1.00000	0.99308	-1.00000
$\rho_1(3)$	0.99885	0.98892	-1.00000
$\rho_2(3)$	-0.06969	-0.20796	-0.26655
$\rho_3(3)$	0.99885	0.98892	-1.00000
Cost Function	26.39285	26.39285	26.39284
Number of Function Evaluations	43	25	147 (obj.) 152 (con.)
Time(s)‡	10.6	36.1	13.9

‡ Execution time (seconds) on CYBER 170/730.

TABLE IX
OPTIMAL TUNING DESIGN OF THE LC LOW-PASS FILTER FOR c_i = 70

Solution	MMLC	MFNC	MINOS
$L_1^0 = L_2^0$	2.04102	2.04100	2.04102
C^0	0.90632	0.90632	0.90632
$100\ \varepsilon_1/L_1^0=100\ \varepsilon_3/L_2^0$	15.19%	15.19%	15.19%
$100\ \varepsilon_2/C^0$	11.95%	11.95%	11.95%
$100\ t_1/L_1^0=100\ t_3/L_2^0$	0.00%	0.00%	0.00%
$100\ t_2/C^0$	10.76%	10.75%	10.76%
$\rho_1(6)$	-1.00000	-0.99922	-1.00000
$\rho_2(6)$	1.00000	1.00000	1.00000
$\rho_3(6)$	-1.00000	-0.99922	-1.00000
$\rho_1(8)$	-0.98911	-0.89791	-1.00000
$\rho_2(8)$	-1.00000	-1.00000	-1.00000
$\rho_3(8)$	-0.98911	-0.89791	-1.00000
$\rho_1(1)$	1.00000	0.83052	-1.00000
$\rho_2(1)$	1.00000	1.00000	1.00000
$\rho_3(1)$	1.00000	0.83052	-1.00000
$\rho_1(3)$	0.98531	1.00000	-1.00000
$\rho_2(3)$	0.02449	-0.16450	-0.32004
$\rho_3(3)$	0.98531	1.00000	1.00000
Cost Function	29.06360	29.06360	29.06359
Number of Function Evaluations	54	26	153 (obj.) 158 (con.)
Time(s)‡	14.6	38.4	14.2

‡ Execution time (seconds) on CYBER 170/730.

[2] J. Hald and K. Madsen, "Combined LP and quasi-Newton methods for minimax optimization", Mathematical Programming, vol. 20, 1981, pp. 49-62.

[3] K. Madsen (Adapted and Edited by J.W. Bandler and W.M. Zuberek), "MINI5W - a Fortran package for minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-280, 1981.

[4] J.W. Bandler and W.M. Zuberek, "MMLC - a Fortran package for linearly constrained minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-292, 1982.

[5] J. Hald (Adapted and Edited by J.W. Bandler and W.M. Zuberek), "MMLA1Q - a Fortran package for linearly constrained minimax optimization", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-281, 1981.

[6] J.W. Bandler and W.M. Zuberek, "MFNC - a Fortran package for minimization with general constraints", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-294, 1982.

[7] VF02AD subroutine specification, Harwell Subroutine Library, AERE, Harwell, Oxfordshire, England, February 1978.

[8] S.P. Han, "Superlinearly convergent variable metric algorithms for general nonlinear programming problem", Mathematical Programming, vol. 11, 1976, pp. 263-282.

[9] M.J.D. Powell, "A fast algorithm for nonlinearly constrained optimization calculations", in Numerical Analysis, Proc. Biennial Conf. (Dundee, Scotland, 1977), Lecture Notes in Mathematics 630, G.A. Watson, Ed. Berlin: Springer-Verlag, 1978, pp. 144-157.

[10] B.A. Murtagh and M.A. Saunders, "MINOS/-AUGMENTED user's manual", Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California 94305, Technical Report SOL80-14, June 1980.

[11] B.A. Murtagh and M.A. Saunders, "A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints", Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, California 94305, Technical Report SOL80-1R, revised February 1981.

[12] A.R. Colville, "A comparative study of nonlinear programming codes", IBM Scientific Center, Yorktown Heights, New York, Report 320-2949, 1968.

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

[14] J.B. Rosen and S. Suzuki, "Construction of nonlinear programming test problems", Comm. ACM, vol. 8, 1965, p. 113.

[15] H.H. Rosenbrock, "An automatic method for finding the greatest or least value of a function", Comp. J., vol. 3, 1960, pp. 175-184.

[16] J.W. Bandler and P.A. Macdonald, "Optimization of microwave networks by razor search", IEEE Trans. Microwave Theory Tech., vol. MTT-17, 1969, pp. 552-562.

[17] J.W. Bandler and P.A. Macdonald", "Cascaded noncommensurate transmission-line networks as optimization problems", IEEE Trans. Circuit Theory, vol. CT-16, 1969, pp. 391-394.

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

[19] J.W. Bandler and C. Charalambous, "On conditions for optimality in least pth approximation with $p \to \infty$", J. of Optimization Theory and Applications, vol. 11, 1973, pp. 556-566.

[20] J.W. Bandler, T.V. Srinivasan and C. Charalambous, "Minimax optimization of networks by grazor search", IEEE Trans. Microwave Theory Tech., vol. MTT-20, 1972, pp. 596-604.

[21] S.R.K. Dutta and M. Vidyasagar, "New algorithms for constrained minimax optimization", Mathematical Programming, vol. 13, 1977, pp. 140-155.

[22] K. Madsen and H. Schjaer-Jacobsen, "Singularities in minimax optimization of networks", IEEE Trans. Circuits and Systems, vol. CAS-23, No. 7, 1976, pp. 456-460.

[23] J.W. Bandler, P.C. Liu and H. Tromp, "A nonlinear programming approach to optimal design centering, tolerancing and tuning", IEEE Trans. Circuits and Systems, vol. CAS-23, 1976, p. 155-165.

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

[25] W.I. Zangwill, Nonlinear Programming: A Unified Approach. Prentice-Hall, Englewood Cliffs, New Jersey, 1969.

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

[27] A.D. Warren, L.S. Lasdon and D.F. Suchman, "Optimization in engineering design", Proc. IEEE, vol. 55, 1967, pp. 1885-1897.

[28] K. Madsen and H. Schjaer-Jacobsen, "Linearly constrained minimax optimization", Mathematical Programming, vol. 14, 1978, pp. 208-223.

[29] K. Madsen, "An algorithm for minimax solution of overdetermined systems of non-linear equations", J. Inst. of Mathematics and its Applications, vol. 16, 1975, pp. 321-328.

[30] J.W. Bandler, "Nonlinear programming using Lagrangian functions", Faculty of Engineering, McMaster University, Hamilton, Canada, Report SOC-284, 1981.

[31] B.A. Murtagh and M.A. Saunders, "The implementation of a Lagrangian-based algorithm for sparse nonlinear constraints", Systems Optimization Laboratory, Department of Operations Research, Stanford University, Stanford, CA, Report SOL80-1, 1980.

[32] S.M. Robinson, "A quadratically convergent algorithm for general nonlinear programming problems", Mathematical Programming, vol. 3, 1972, pp. 145-156.

[33] J.W. Bandler, "Optimization methods for computer-aided design", IEEE Trans. Microwave Theory Tech., vol. MTT-17, 1969, pp. 533-552.

[34] C. Charalambous and A.R. Conn, "An efficient method to solve the minimax problem directly", SIAM J. Numerical Analysis, vol. 15, 1978, pp. 162-187.

[35] D.M. Himmelblau, Applied Nonlinear Programming. New York: McGraw-Hill, 1972.