NEW ALGORITHMS FOR CONSTRAINED MINIMAX OPTIMIZATION WITH APPLICATIONS TO NETWORK DESIGN

Shiraj Robi Kumar Dutta

A Thesis

in

The Faculty

of

Engineering

Presented in Partial Fulfillment of the Requirement for
the degree of Doctor of Engineering at
Concordia University
Montreal, Quebec, Canada

August, 1975

NEW ALGORITHMS FOR CONSTRAINED MINIMAX OPTIMIZATION WITH APPLICATIONS TO NETWORK DESIGN

SHIRAJ ROBI KUMAR DUTTA

ABSTRACT

The problem of minimax optimization is studied from a mathematical point of view. It is shown that the negative of the gradient of a least p-th performance function with an extremely large value of p does not always yield a descent direction for the corresponding minimax function. Based on the results of the study an algorithm is proposed for unconstrained minimax problems, where the direction of steepest descent to a first order approximation is obtained by a linear programming subproblem.

A mathematical justification is given for applying an algorithm to convert a constrained minimax problem to a sequence of unconstrained minimax problems. The idea is then used to solve a nonlinear programming problem with equality and inequality constraints by converting it into a minimax problem with equality constraints.

A method is given to convert a constrained minimax problem into sequential minimization of continuously differentiable least-squares type functions, thereby enabling one to use an efficient gradient technique to minimize such a function. Two algorithms are described for such an approach, of which the first one is simpler to program whereas the second algorithm is much faster in general.

It is shown that a penalty function method can be efficiently applied for the optimal design of nonlinear dc transistor circuits without solving network equations, for either a least-squares type or a minimax performance function.

The algorithms are applied to solve different kinds of problems, including filter design, optimal design of nonlinear dc circuits, and the modelling of a Schottky-clamped transistor.

ACKNOWLEDGEMENTS

The author expresses his sincere gratitude to Dr. M. Vidyasagar for his guidance and assistance during the course of this investigation, and for his advice during the preparation of the manuscript.

The author gratefully acknowledges the support of the National Research Council of Canada through grant A-7790, and through the award of an NRC Postgraduate Scholarship.

TO MY PARENTS

AND

TO MY WIFE SAKHI

LIST OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS

R^n	:	n-dimensional Euclidean space
$\underset{\sim}{\ }$	:	This symbol under a letter denotes a vector
v^T	:	Superscript T denotes transposition
∇f	:	Gradient of a function f
$\|\|.\|\|$	:	Norm of a vector
sup	:	Supremum
inf	:	Infimum
sgn	:	Signum
max	:	Maximum
Re(.)	:	Real part of a complex argument
$\delta f(x;h)$	:	Directional derivative of f at $\underset{\sim}{x}$ in the direction $\underset{\sim}{h}$
$x \leftarrow y$	:	Left hand side is assigned the value of the right hand side
$\triangleq$	:	Equal by definition
ρ	:	Reflection coefficient
$\|.\|$	:	Absolute value of a function
$\underset{\sim}{H}$	:	Hessian of a function

CHAPTER 1

INTRODUCTION

1.1 MOTIVATION FOR COMPUTER-AIDED DESIGN

In a remarkably short span of time, the digital computer has become an indispensable design tool. With the aid of a computer and the mathematical optimization techniques, the engineer can not only design large and complex systems in short periods of time, but perhaps even more importantly, he can also remove many of the restrictive, and sometimes invalid, assumptions that are made when classical design procedures are used [36, 38]. He can also incorporate several kinds of constraints, and can realize compromise solutions reconciling conflicting requirements carrying different weights. The main advantage of design techniques based upon iterative optimization is that they lead to the best possible design under the given circumstances rather than just a "feasible" design that meets some given specifications.

The field of electrical circuit design is prominent among the many areas in which iterative optimization techniques have been successfully applied. The classical network synthesis procedures restrict the network configuration and the degrees of freedom that may be demanded by the designer. However, with the help of a computer-aided design procedure, one can accommodate prescribed active elements, nonlinearities, parasitics, as well as restrictions on the types and values of the elements. One can also "grow" new elements in a given circuit [14] thereby modifying the structure of the network, or can improve the performance of circuits desgined by simplified or heuristic methods [36, 38]. However, before any such automated design procedure is

envisaged, the following steps must be completed:

(i) An appropriate design criterion or a performance function involving the design specifications, the constraints on the design parameters etc. must be explicitly defined.

(ii) A reliable and efficient algorithm must be chosen for the minimization of the performance function.

The performance function can be defined in various ways depending on the design requirements. Two of the most commonly used performance functions are (i) least-squares type, and (ii) the minimax or Chebyshev type. Minimax optimization is of particular significance in the fields of circuit design and system modelling. For example, in filter design, the specifications are normally given in terms of the maximum allowable ripple in the passband and minimum allowable attenuation in the stopband [36]. Similarly, in the worst-case design of dc amplifiers, one tries to minimize the maximum fluctuation in the operating point from the nominal value over varying operating conditions. Requirements like these can be most naturally represented by a minimax performance function. Likewise in a modelling problem, minimax optimization assures that the maximum deviation of the model response from that of the actual system is minimized. However, the reliability of minimax modelling very much depends on the reliability of the available measurements on the device. If some measurements are highly unreliable, the minimax optimum may be drastically affected by the unreliable data, and hence a least-squares performance function is a better choice in such a case. However, minimax modelling is more desirable when the error in measurement is uniform.

1.2 SCOPE OF THE THESIS

This thesis is mainly concerned with (i) finding efficient algorithms and proposing new approaches to the constrained optimization of minimax performance functions and (ii) the application of these algorithms to circuit design.

In Chapter 2, the minimax optimization problem is studied from a mathematical point of view, and based on the results, a minimax algorithm is proposed for filter design problems.

An expression for the directional derivative of the ℓ_∞-norm is derived, and it is shown that one can always find a descent direction for a minimax performance function by solving a linear programming problem, which in fact is the direction of steepest-descent to a first order approximation. The gradient minimization techniques of Fletcher [21] and Fletcher-Powell [20] are then adapted to generate the search directions for an efficient minimax algorithm.

A recent practice [5] is to solve the minimax problem by solving the corresponding least p-th problem with extremely large values of p. However, it is shown in this chapter that the limit as $p \to \infty$ of the negative of the gradient of the least p-th performance function does not in general yield a descent direction for the minimax performance function. This explains why the least p-th approach with extremely large values of p is sometimes found not to converge to the minimax optimum [4].

Chapter 3 deals with the applications of the minimax algorithm proposed in Chapter 2 to circuit design problems. First, it is applied to the design of three kinds of filters, namely, (i) n-section transmission line transformers, (ii) lumped LC filters, and (iii) lumped-distributed-active lowpass filters. Next, computer-aided minimax modelling is proposed for a Schottky-Barrier diode clamped transistor. It is also illustrated that the model may be improved by "growing" certain new elements in the original model.

Next, in Chapter 4, the interrelationships between a nonlinear programming problem with equality constraints, and an unconstrained minimax optimization problem are studied. An algorithm is proposed for the minimax optimization with constraints, by giving a mathematical justification for applying an existing algorithm [27, 31] for nonlinear programming to minimax performance functions. It is shown that a nonlinear programming problem with both equality and inequality constraints can be converted into a minimax problem with only equality constraints, which can then be solved by the method proposed for solving a constrained minimax problem. An example is taken to show the effectiveness of the method.

In Chapter 5, the unconstrained minimax optimization problem is tackled from entirely a different viewpoint. Here, a minimax optimization problem with constraints is converted to a sequence of unconstrained least-squares type optimizations, of continuously differentiable functions, so that any efficient gradient method can be used to perform these minimizations. Hence, unlike the methods proposed in Chapters 2 and 4,

the computation of a descent direction does not involve solving a linear programming subproblem, and at the same time, the computation of the gradient of the performance function is much more economical than in the least p-th techniques.

It is shown that under relatively mild assumptions, the sequence of solutions for the least-squares type problems converges to the minimax solution. Two algorithms for minimax optimization are presented based on this method. The second one converges faster in general, whereas the first one is simpler to program. These algorithms are applied to solve several unconstrained and constrained problems, including filter design problems. The feasibility of the algorithms for constrained problems is verified by the results. They are also found to be robust and fast compared to the existing methods for unconstrained methods [4, 5, 6, 9].

Finally, in Chapter 6, a method is given for the optimal design of nonlinear dc transistor circuits without solving network equations. The novel approach in this chapter is to treat the nonlinear network equations as equality constraints on the design parameters. The constrained optimization problem is then converted to a sequence of unconstrained optimization problems by the method due to Morrison [31]. A straightforward method is then given for computing all the gradients needed for the optimization, given only the topology of the network and the branch relationships. The fact that the gradient vector of the performance function can be obtained readily makes the algorithm easily amenable to a package program.

Worst-case design of dc amplifiers is also studied in this Chapter. It is shown by taking an example that a minimax performance function as opposed to a least-squares type is better suited for this purpose. The method for constrained minimax optimization as described in Chapter 4 is used for this purpose. Various examples are taken to study the efficiency and feasibility of the approach, and it is found to be much faster than the existing approach of solving the network equations explicitly at each step [17].

The digital computer used to solve all the numerical examples is a CDC 6400. All the programs are written in FORTRAN IV.

1.3 MAIN CONTRIBUTIONS OF THE THESIS

(i) The minimax optimization problem is studied from a directional derivative point of view which enables one to detect that the direction of descent obtained by linear programming is in fact the direction of steepest descent to a first order approximation. Based on the study a minimax optimization algorithm is proposed, which is found to be at least competitive with the existing methods.

(ii) The limiting behaviour of the least p-th optimization with large values of p is studied, and it is shown that the negative of the gradient of a least p-th performance function with extremely large values of p does not in general give a descent direction for the corresponding true minimax performance function. This shows that, even though the least p-th performance function converges point wise to the minimax performance function (as $p \to \infty$), sequential minimization

of the least p-th performance function does not necessarily lead to a solution of the minimax problem. This explains why the least p-th approach with $p \to \infty$ is not always a good way to solve the minimax problem, since the final convergence may be far removed from the true minimax optimum.

(iii) Minimax optimization is applied to the modelling of a Schottky-clamped transistor, where it is shown that the approach lends flexibility to modelling by allowing one to "grow" new elements, or put varying emphasis on the responses at different excitations by using weighting parameters. We also discuss how one can have a combination of least-squares and minimax performance functions for better results under certain conditions.

(iv) A mathematical justification is given for adopting the nonlinear programming algorithm of Kowalik et al. [27] for solving a constrained minimax problem. It is quite significant since no reliable method exists for such a problem.

(v) A method is given to solve a nonlinear programming problem with equality and inequality constraints by converting it into a minimax problem with equality constraints, which is found to be at least comparable to the existing methods.

(vi) A new approach to constrained minimax problem is given, by converting it into a sequential minimization of continuously differentiable least-squares type performance functions, which can be carried out very efficiently. One of the advantages of the approach is the flexibility it can offer to the computer-aided designer through a unified treatment

of the constrained and unconstrained minimax optimization problems. In fact no other satisfactory method exists in the literature for constrained problems [8]. Even for unconstrained minimax problems, the method is found to be much faster than the existing techniques [4, 5, 9].

(vii) A method is given for the optimal design of nonlinear dc transistor circuits without solving the network equations explicitly. It is well known that the analysis of a nonlinear circuit is a time consuming process. In a design problem involving such a network, the network has to be analyzed each time one needs to compute the performance function and its gradient, which can slow down an algorithm considerably. Instead, we treat the nonlinear network equations as equality constraints on the optimization parameters, and a penalty function method [31] is used to solve the problem. It is found by solving various examples that this procedure cuts down the execution time quite significantly. It is also shown that the gradient of the performance function can be computed readily which makes it easily amenable to packaging.

(viii) Worst-case design of dc amplifiers is proposed by defining a minimax performance function instead of a least-squares one. By solving an example it is found that the maximum deviation in the operating point is less for a minimax performance function than for a least-squares performance function.

CHAPTER 2

UNCONSTRAINED MINIMAX OPTIMIZATION

2.1 INTRODUCTION

The problem of minimax optimization is of prime significance in many circuit design problems. For example, in filter design, the passband and stopband specifications are most naturally stated in minimax terms [36,38]. Up to now, there has been a reluctance to apply any of the efficient gradient minimization techniques, such as the Fletcher-Powell method [20] and the method due to Fletcher [21] to minimax problems. This is mainly due to the increased mathematical complexity involved in dealing with the ℓ_∞-norm, which is not differentiable everywhere. In contrast, the ℓp-norm with $p < \infty$ is differentiable everywhere except at the origin. Since, for each fixed $\underset{\sim}{x}$, the ℓp-norm of $\underset{\sim}{x}$ approaches the ℓ_∞-norm of $\underset{\sim}{x}$ as $p \to \infty$, a popular method has been to minimize an ℓp-norm cost functional with very large values of p [5].

Recently, Bandler and Charalambous have given a near minimax method [5] and a minimax approach [9] using the least p-th optimization. In [5] large values of p are used to obtain near minimax results, and a few strategies are adopted to get around the problem of ill-conditioning. Bandler et al. also give an approach in [4] where a descent direction for a minimax cost functional is found through linear programming, and a modified method of steepest descent is used in the minimization algorithm.

The objective of this chapter is two-fold: (i) to study the minimax optimization problem from a mathematical point of view, and (ii) to propose a practical computational scheme for minimax optimization. It

is shown that at every nonzero point, one can find a unique "steepest descent direction" for the ℓ_∞-norm, even if no Frechet derivative exists. However, such a direction is a direction of steepest descent to a first order approximation only, in the same way as the negative of the gradient for differentiable functions. On this basis, a computational scheme is developed for minimax optimization.

Another algorithm is also given for near minimax optimization, which is similar to the least p-th approach in [5]; however, it has two advantages over the least p-th technique: (i) it is inherently well-conditioned, and (ii) it requires many fewer gradient calculations and hence, in the case of circuit design, many fewer adjoint network analyses [13].

2.2 MATHEMATICAL BACKGROUND

2.2.1 The ℓ_∞-norm

We begin by summarizing a few well known results, which can be found in [1]. Let $f: R^n \to R$, and let $\underline{x}$, $\underline{h} \in R^n$. If the limit

$$\delta f(\underline{x};\underline{h}) = \lim_{\alpha \to 0+} \frac{f(\underline{x} + \alpha\underline{h}) - f(\underline{x})}{\alpha} \tag{2.1}$$

exists, it is called the directional derivative of f at $\underline{x}$ in the direction $\underline{h}$. Another way of defining $\delta f(\underline{x};\underline{h})$ is

$$\delta f(\underline{x};\underline{h}) = \frac{d^+}{d\alpha}\left[f(\underline{x} + \alpha\underline{h})\right]_{\alpha=0} \tag{2.2}$$

In addition, if there is an element $\underline{y} \in R^n$ such that

$$\delta f(\underline{x};\underline{h}) = \underline{y}^T\underline{h}, \ \forall\ \underline{h} \in R^n \tag{2.3}$$

then f is said to be Fréchet differentiable at $\underset{\sim}{x}$, and $\underset{\sim}{y}$ is called the Fréchet derivative of f at $\underset{\sim}{x}$ (denoted by $\nabla f(\underset{\sim}{x})$).

Proposition 2.1 below gives a general necessary condition for minimality, and can be found in [1, p. 95].

Proposition 2.1: Suppose $f: R^n \to R$ has a directional derivative at each $\underset{\sim}{x} \in R^n$ in each direction $\underset{\sim}{h} \in R^n$. Then a necessary condition for f to attain a minimum at $\underset{\sim}{x}^* \in R^n$ is

$$\delta f(\underset{\sim}{x}^*;\underset{\sim}{h}) \geq 0, \ \forall \ \underset{\sim}{h} \in R^n \tag{2.4}$$

If f is actually Fréchet differentiable at $\underset{\sim}{x}^*$, we must have

$$\nabla f(\underset{\sim}{x}^*) = 0 \tag{2.5}$$

Let us now turn to norm functions.

Proposition 2.2: Let $||\cdot|| : R^n \to [0,\infty]$ be any norm on R^n. Then $||\cdot||$ has a directional derivative at each point in each direction.

This is derived in [16, p. 30, Lemma II.8.4].

Proposition 2.3: Let $||\cdot||$ be any norm on R^n, and let $||\cdot||_d$: $R^n \to [0,\infty]$ be its dual norm, defined by

$$||\underset{\sim}{y}||_d = \sup_{||\underset{\sim}{x}||=1} \underset{\sim}{y}^T \underset{\sim}{x} \tag{2.6}$$

Then $||\cdot||$ is Fréchet differentiable at $\underset{\sim}{x} \in R^n$ if and only if there exists a unique $\underset{\sim}{y} \in R^n$ such that

$$||\underset{\sim}{y}||_d = 1, \text{ and } \underset{\sim}{y}^T\underset{\sim}{x} = ||\underset{\sim}{x}|| \tag{2.7}$$

If (2.7) holds, this unique $\underset{\sim}{y}$ is the Fréchet derivative of $||\cdot||$ at $\underset{\sim}{x}$.

This proposition is based on a classical theorem due to Smulian [15], and is proved in [39].

Proposition 2.4: Let $||\cdot||_\infty$ be the ℓ_∞-norm on R^n defined by

$$||\underset{\sim}{x}||_\infty = \max_i |x_i| \tag{2.8}$$

where x_i, $i = 1, \ldots, n$ are the components of $\underset{\sim}{x}$. Then

(i) $||\cdot||_\infty$ is Frechet differentiable at $\underset{\sim}{x}$ if and only if the maximum indicated in (2.8) is attained for a unique value of i, say i_0. In this case, the Frechet derivative of $||\cdot||_\infty$ at $\underset{\sim}{x}$ is

$$[0 \quad 0 \quad . \; . \quad \underset{\substack{\uparrow \\ i_0\text{-th component}}}{\operatorname{sgn} x_{i_0}} \; . \; . \quad 0]^T \tag{2.9}$$

(ii) If $\underset{\sim}{x}$ is such that the maximum indicated in (2.8) is attained for more than one value of i, define

$$I(\underset{\sim}{x}) = \{i : |x_i| = ||\underset{\sim}{x}||_\infty\} \tag{2.10}$$

If $\underset{\sim}{x} \neq 0$, the directional derivative of $||\cdot||_\infty$ at $\underset{\sim}{x}$ in the direction $\underset{\sim}{h}$ (denoted by $D(\underset{\sim}{x};\underset{\sim}{h})$) is given by

$$D(\underset{\sim}{x};\underset{\sim}{h}) = \max_{i \in I(\underset{\sim}{x})} (h_i \operatorname{sgn} x_i) \tag{2.11}$$

while if $\underset{\sim}{x} = \underset{\sim}{0}$,

$$D(\underset{\sim}{0};\underset{\sim}{h}) = ||\underset{\sim}{h}||_\infty \tag{2.12}$$

Proof: The dual norm of $||\cdot||_\infty$ is the ℓ_1-norm defined by

$$||\underset{\sim}{x}||_1 = \sum_{i=1}^{n} |x_i| \tag{2.13}$$

It is easy to see that corresponding to a given $\underset{\sim}{x} \in R^n$, there exists a unique $\underset{\sim}{y} \in R^n$ satisfying

$$||\underset{\sim}{y}||_1 = 1, \quad \text{and} \quad \underset{\sim}{y}^T\underset{\sim}{x} = ||\underset{\sim}{x}||_\infty \tag{2.14}$$

if and only if one component of $\underset{\sim}{x}$ is larger in magnitude than all the others (as in case (i) above), in which case $\underset{\sim}{y}$ is of the form (2.9). This proves (i).

To prove (ii), note that for sufficiently small α, we have

$$\max_{i \in I(\underset{\sim}{x})} |x_i + \alpha h_i| > \max_{i \notin I(\underset{\sim}{x})} |x_i + \alpha h_i| \tag{2.15}$$

Therefore,

$$||\underset{\sim}{x} + \alpha\underset{\sim}{h}||_\infty = \max_{i \in I(\underset{\sim}{x})} |x_i + \alpha h_i| \tag{2.16}$$

If $||\underset{\sim}{x}||_\infty > 0$, we can write (2.16) as

$$||\underset{\sim}{x} + \alpha\underset{\sim}{h}||_\infty = \max_{i \in I(\underset{\sim}{x})} |x_i||1 + \alpha(h_i/x_i)| \tag{2.17}$$

Again for sufficiently small α, $1 + (\alpha h_i/x_i) > 0$ for all $i \in I(\underset{\sim}{x})$, so that (2.17) can be rewritten as

$$||\underset{\sim}{x} + \alpha\underset{\sim}{h}||_\infty = \max_{i \in I(\underset{\sim}{x})} |x_i|(1 + \alpha(h_i/x_i)) \tag{2.18}$$

Now, for positive α, (2.18) can be rewritten as

$$\begin{aligned} ||\underset{\sim}{x} + \alpha\underset{\sim}{h}||_\infty &= ||\underset{\sim}{x}||_\infty + \alpha \max_{i \in I(\underset{\sim}{x})} |x_i|(h_i/x_i) \\ &= ||\underset{\sim}{x}||_\infty + \alpha \max_{i \in I(\underset{\sim}{x})} (h_i \operatorname{sgn} x_i) \end{aligned} \tag{2.19}$$

Applying definition (2.1), we immediately have (ii). On the other hand, if $\underset{\sim}{x} = \underset{\sim}{0}$, then for $\alpha > 0$, we have

$$||\underset{\sim}{x} + \alpha\underset{\sim}{h}||_\infty = ||\underset{\sim}{x}||_\infty + \alpha||\underset{\sim}{h}||_\infty \tag{2.20}$$

from which (2.12) follows readily. ⊠

Note that the first step in (2.19) is valid only for $\alpha > 0$. Also, $D(\underset{\sim}{x};\underset{\sim}{h})$ is continuous in $\underset{\sim}{h}$ at each $\underset{\sim}{x}$.

Proposition 2.5: Let $\underset{\sim}{e}: R^n \to R^m$ be Fréchet differentiable (i.e. each component of $\underset{\sim}{e}$ is Fréchet differentiable), and define $f: R^n \to R$ by

$$f(\underset{\sim}{x}) = ||\underset{\sim}{e}(\underset{\sim}{x})||_\infty \tag{2.21}$$

Then f has a directional derivative at each $\underset{\sim}{x} \in R^n$ in each direction $\underset{\sim}{h} \in R^n$, and

$$\delta f(\underset{\sim}{x};\underset{\sim}{h}) = D\{\underset{\sim}{e}(\underset{\sim}{x}); [\nabla\underset{\sim}{e}(\underset{\sim}{x})]^T\underset{\sim}{h}\} \tag{2.22}$$

where $\nabla\underset{\sim}{e}(\underset{\sim}{x})$ is the Fréchet derivative of $\underset{\sim}{e}$ at $\underset{\sim}{x}$, and $D(.;.)$ is as defined in (2.11) and (2.12).

Proof: Let $o(\alpha)$ represent a function with the property that $o(\alpha)/\alpha \to 0$ as $\alpha \to 0$. Then

$$\begin{aligned} f(\underset{\sim}{x} + \alpha\underset{\sim}{h}) &= ||\underset{\sim}{e}(\underset{\sim}{x} + \alpha\underset{\sim}{h})||_\infty \\ &= ||\underset{\sim}{e}(\underset{\sim}{x}) + \alpha[\nabla\underset{\sim}{e}(\underset{\sim}{x})]^T\underset{\sim}{h} + \underset{\sim}{o}_1(\alpha)||_\infty \\ &= ||\underset{\sim}{e}(\underset{\sim}{x})||_\infty + \alpha D(\underset{\sim}{e}(\underset{\sim}{x}); [\nabla\underset{\sim}{e}(\underset{\sim}{x})]^T\underset{\sim}{h} + \underset{\sim}{o}_1(\alpha)/\alpha) + \underset{\sim}{o}_2(\alpha) \\ &= ||\underset{\sim}{e}(\underset{\sim}{x})||_\infty + \alpha D(\underset{\sim}{e}(\underset{\sim}{x}); [\nabla\underset{\sim}{e}(\underset{\sim}{x})]^T\underset{\sim}{h}) + \underset{\sim}{o}_3(\alpha) \end{aligned}$$

This proves (2.22). ⊠

Thus the directional derivative of the ℓ_∞ norm can be readily calculated, once the vector $\nabla e(\underset{\sim}{x})$ is known.

2.2.2 Directional Derivative of $f(\underset{\sim}{x}) = \max_i e_i(\underset{\sim}{x})$

In a filter design problem it is very often desired to minimize $\max_i e_i(\underset{\sim}{x})$ instead of the ℓ_∞-norm given by $\max_i |e_i(\underset{\sim}{x})|$ [11, 38]. In this section, we derive the expressions for the directional derivative of this function. It is also worth noting that the problem of ℓ_∞-norm minimization can be converted to this kind of a problem since the minimization of $\max_i |e_i(\underset{\sim}{x})|$ is equivalent to the minimization of $\max_i [e_i(\underset{\sim}{x})]^2$. However, minimization of $\max_i e_i(\underset{\sim}{x})$ cannot be converted to an ℓ_∞-norm minimization problem.

Proposition 2.6: Defining $f_M: R^m \to R$ by

$$f_M(\underset{\sim}{x}) = \max_i x_i \tag{2.23}$$

the directional derivative of $f_M(\cdot)$ at $\underset{\sim}{x}$ in the direction $\underset{\sim}{h}$ is given by

$$D_M(\underset{\sim}{x};\underset{\sim}{h}) = \max_{i \in I_M(\underset{\sim}{x})} h_i \tag{2.24}$$

where $I_M(\underset{\sim}{x}) = \{i: x_i = \max_j x_j\}$

Proof: In (2.23), $f_M(\underset{\sim}{x})$ is continuous. Thus,

$$f_M(\underset{\sim}{x} + \alpha\underset{\sim}{h}) = \max_{1 \le i \le n} (x_i + \alpha h_i)$$

$$= \max_{i \in I_M(\underset{\sim}{x})} (x_i + \alpha h_i)$$

for sufficiently small α. The directional derivative can now be expressed as

$$D_M(\underset{\sim}{x};\underset{\sim}{h}) = (f_M(\underset{\sim}{x} + \alpha\underset{\sim}{h}) - f_M(\underset{\sim}{x}))/\alpha$$

$$= \max_{i \in I_M(\underset{\sim}{x})} h_i \quad \square$$

The directional derivative of $f(\underset{\sim}{x}) = \max_i e_i(\underset{\sim}{x})$ is now given by the following proposition.

Proposition 2.7: Let $\underset{\sim}{e}: R^n \to R^m$ be Frechet differentiable everywhere, and let $f: R^n \to R$ be defined by $f(\underset{\sim}{x}) = f_M(\underset{\sim}{e}(\underset{\sim}{x}))$, where f_M is as defined in (2.23). Then the directional derivative of $f(\cdot)$ at $\underset{\sim}{x}$ in the direction $\underset{\sim}{h}$ is given by

$$\delta f_M(\underset{\sim}{x};\underset{\sim}{h}) = \max\{[\nabla \underset{\sim}{e}(\underset{\sim}{x})]^T \underset{\sim}{h}\}_i, \ i \in I_M(\underset{\sim}{e}(\underset{\sim}{x})) \qquad (2.25)$$

where

$$I_M(\underset{\sim}{e}(\underset{\sim}{x})) = \{i : [\underset{\sim}{e}(\underset{\sim}{x})]_i = \max_j [\underset{\sim}{e}(\underset{\sim}{x})_j]\}$$

The proof is similar to that of Proposition 2.5, and is therefore omitted.

2.3 CALCULATING A DESCENT DIRECTION

An algorithm due to Bandler and Macdonald [3] exploits the gradient information of the extrema of the performance function to get a downhill direction by solving a set of simultaneous equations. More recently, Bandler et al. [4] have proposed a method to find a descent direction by using linear programming. In this section, we show that the calculation of a descent direction can be converted to a linear programming problem, the solution of which in fact yields the direction of steepest descent to a first order approximation.

2.3.1 A Necessary Condition for a Descent Direction

Proposition 2.8: Let $\underset{\sim}{e}: R^n \to R$, and let $f = ||\underset{\sim}{e}(\underset{\sim}{x})||_\infty$, and $\underset{\sim}{h} \to R^n$, where $\underset{\sim}{e}(\underset{\sim}{x})$ is the vector of the components $e_i(\underset{\sim}{x})$. Then a necessary condition for $\underset{\sim}{h}$ to be a descent direction of f is

$$[\nabla e_i^T(\underset{\sim}{x}) \text{ sgn } e_i(\underset{\sim}{x})]\underset{\sim}{h} \leq \underset{\sim}{0}, \quad \forall\ i \in I(e_i(\underset{\sim}{x})) \tag{2.26}$$

and a necessary condition for a local minimum is

$$\max_i\{[\nabla e_i^T(\underset{\sim}{x}) \text{ sgn } e_i(\underset{\sim}{x})]\}\underset{\sim}{h} \geq 0, \quad \forall\ \underset{\sim}{h} \in R^n \tag{2.27a}$$

which can also be written as

$$\min_{||\underset{\sim}{h}||=1} \{[\nabla e_i^T(\underset{\sim}{x}) \text{ sgn } e_i(\underset{\sim}{x})]\}\underset{\sim}{h} \geq 0 \tag{2.27b}$$

The proof follows readily from (2.22). ∎

It may be noted that (2.27) with strict inequality is a sufficient condition for $\underset{\sim}{x}$ to be a local minimum of f. Clearly, if $\underset{\sim}{x} \in R^n$ violates (2.27), then there exists an $h \in R^n$, such that

$$\underset{\sim}{M}\underset{\sim}{h}_1 < 0 \tag{2.28}$$

where the i-th row of the q x n matrix $\underset{\sim}{M}$ is given by

$$M_i \triangleq [\nabla e_i^T(\underset{\sim}{x}) \text{ sgn } e_i(\underset{\sim}{x})], \quad i \in I(e_i(\underset{\sim}{x})) \tag{2.29}$$

In other words, if (2.27) is violated at $\underset{\sim}{x}$, then one can always find a descent direction.

2.3.2 Linear Programming Formulation

Any vector $\underset{\sim}{h}$ satisfying (2.28) is a descent direction of f. However, to determine the direction of steepest descent we must solve the problem of finding $\underset{\sim}{h}$ which minimizes $\max \underset{\sim}{M}\underset{\sim}{h}_1$. The problem is expressed as

Problem 2.1: Minimize $F = \phi$

such that $\phi - M_i \underset{\sim}{h} \geq 0$

$\underset{\sim}{h}^T \underset{\sim}{h} = 1$

Note that we impose a condition $\underset{\sim}{h}^T \underset{\sim}{h} = 1$, since we are interested only in the direction of descent.

The solution of Problem 2.1 is given by the following proposition.

Proposition 2.9: The solution $\underset{\sim}{h}^*$ of Problem 2.1 is of the form

$$\underset{\sim}{h}^* = -\lambda_h \sum \lambda_i M_i \ , \ \{i \in J = I(\underset{\sim}{e}(\underset{\sim}{x}))\}$$

where $\lambda_i \geq 0$, and the set I is as defined in (2.10).

Proof: The proof can be easily obtained by invoking the Kuhn-Tucker conditions [1, pp. 181-189]. ⊠

In view of Proposition 2.9, Problem 2.1 can be redefined as

Problem 2.2: Minimize $F = -\phi$

such that $-\phi + \sum_j \lambda_j M_i M_j^T \geq 0$, $i \in I(\underset{\sim}{e}(\underset{\sim}{x}))$

and $\lambda_j \geq 0$.

In order to convert Problem 2.2 to a standard linear programming problem we prove the following lemma.

Lemma 2.1: Suppose $\lambda_j \geq 0$, and $\sum_j \lambda_j = 1$. Then

$\sum_j \lambda_j M_i M_j^T \geq 0$ for some i.

Proof: Suppose the contrary is true, i.e.

$\Sigma \lambda_j M_i M_j^T < 0$, $\forall$ i

then $\lambda_i \neq 0$ for some i. Moreover,

$$\sum_i \lambda_i \sum_j \lambda_j M_i M_j^T < 0 \tag{2.30}$$

On the other hand

$$\sum_i \lambda_i \sum_j \lambda_j M_i M_j^T = \sum_i \lambda_i M_i^T \cdot \sum_j M_j \lambda_j$$
$$= \underset{\sim}{g}^T \underset{\sim}{g} \geq 0 \tag{2.31}$$

where

$$\underset{\sim}{g} = \sum_j M_j^T \lambda_j$$

Hence, (2.30) and (2.31) contradict each other, thereby proving the lemma. ∎

The above lemma shows that at a solution of Problem 2.2, ϕ is non-negative. Hence, Problem 2.2 is equivalent to the linear programming problem shown in Problem 2.3 below.

Problem 2.3: Minimize $[-1 \quad \underset{\sim}{0}^T] \begin{bmatrix} \phi \\ \underset{\sim}{\lambda} \end{bmatrix}$,

subject to the constraints

$$\begin{bmatrix} -\underset{\sim}{1} & \underset{\sim}{R} \\ 1 & \underset{\sim}{0}^T \end{bmatrix} \begin{bmatrix} \phi \\ \lambda \end{bmatrix} \geq \underset{\sim}{0} \qquad \underset{\sim}{\lambda} \geq \underset{\sim}{0}$$

where

$$\underset{\sim}{R} \triangleq \underset{\sim}{M}\underset{\sim}{M}^T$$
$$\underset{\sim}{\lambda} \triangleq [\lambda_1, \lambda_2, \ldots, \lambda_q]^T$$
$$\underset{\sim}{1} \triangleq [1, 1, \ldots, 1]^T$$

and q is the number of components in $I(\underset{\sim}{e}(\underset{\sim}{x}))$.

Finally, the problem may be represented as a standard linear programming problem by introducing q slack variables to convert the inequality constraints to equality constraints, as follows:

Problem 2.4: Minimize $F = [\underset{\sim}{0}^T \; -1 \; \underset{\sim}{0}^T] \; \underset{\sim}{x}$ (2.32)

such that $\begin{bmatrix} -\underset{\sim}{R} & \underset{\sim}{1} & \underset{\sim}{I}_q \\ \underset{\sim}{1}^T & 0 & \underset{\sim}{0} \end{bmatrix} \underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{b} \\ 1 \end{bmatrix}$ (2.33)

where

$$\underset{\sim}{x} = [\underset{\sim}{\lambda}^T \; \phi \; \underset{\sim}{x}_q^T]^T \geq \underset{\sim}{0} \; , \tag{2.34}$$

where $\underset{\sim}{I}_q$ is the q x q identity matrix; $\underset{\sim}{b}$ is a vector of zeroes, and $\underset{\sim}{x}_q$ is the vector of slack variables.

It should be noted that in order to obtain an initial basic solution for Problem 4, only one extra slack variable need to be introduced in addition to $\underset{\sim}{x}_q$. The sub-problem to obtain the initial basic solution is expressed as

Problem 2.5: Minimize y

subject to $\begin{bmatrix} -R & \underset{\sim}{1} & \underset{\sim}{I}_q & 0 \\ \underset{\sim}{1}^T & 0 & 0 & 1 \end{bmatrix} \underset{\sim}{x} = \begin{bmatrix} \underset{\sim}{b} \\ 1 \end{bmatrix}$ (2.35)

where

$$\underset{\sim}{x} = [\underset{\sim}{\lambda}^T \; \phi \; \underset{\sim}{x}_q^T \; y]^T \geq 0 \; ,$$

and $y = 0$ at the solution.

To guarantee convergence to the solution in Problem 5, the vector $\underset{\sim}{b}$ is made ≥ 0. It should be noted that the solution to Problem 4 is then given by

$$\phi_1 = -\phi + b_1$$

where ϕ is the solution with $\underset{\sim}{b} = \underset{\sim}{0}$, and b_1 is an element of $\underset{\sim}{b}$.

2.4 ALGORITHM AND PROGRAM STRATEGY

2.4.1 Introduction

It can easily be shown [18] that if the minimax function is minimized by doing a linear search in the direction of "steepest descent" $\underset{\sim}{h}$ as obtained by the linear programming, it is equivalent to a "method of feasible directions" due to Zoutendijk [23] for solving the following nonlinear programming problem:

Minimize ϕ subject to

$$f_i(\underset{\sim}{x}) - \phi \leq 0$$

An adaptation of the steepest descent technique to the optimization of a directionally differentiable function is studied in [40], and a proof of convergence is given therein. However, in practice this procedure is found to be very slow. Thus we adapt the methods of Fletcher [21], and Fletcher-Powell to generate a search direction by treating the direction of steepest descent as if it were the negative of the gradient. Also, certain other strategies have to be made to make the approach effective.

Based on the foregoing ideas, a computer program has been written and tested. Figure 2.1 shows the program organization, how the different subroutines are used, and their hierarchy. The algorithm is described in the next section.

2.4.2 Description of the Algorithm

The performance function is first minimized by using Fletcher's

method without any modifications. However, it is found that the rate of convergence near the minimum slows down considerably. Therefore, when this happens, the algorithm switches to Fletcher-Powell method, with the final parameter values from Fletcher's method taken as the initial values. A few modifications are made in the latter method (Fletcher-Powell) to assure convergence. This is illustrated by the flow-chart in Figure 2.2. It is found that the matrix $\underset{\sim}{H}$ used in determining the direction of search, (which is an approximation to the inverse of the Hessian), sometimes has a tendency to become singular. The slope of the performance function with respect to the step size α at $\alpha = 0$ then approaches zero, or may even become positive due to round-off errors. When such a situation arises, the H matrix is reset to be diagonal with the entry H_{ii} being the ratio of the i-th element of $(\Delta\underset{\sim}{x})$ to the i-th element of $\nabla f(\underset{\sim}{x})$ (see [23, p. 119]).

2.4.3 Criterion to Determine if a Component $e_i(\underset{\sim}{x})$ is close to the Maximum

While computing the direction of descent, it must be borne in mind that the direction should not be a descent direction with respect to the components of the performance function $\underset{\sim}{e}(\underset{\sim}{x})$ which are equal to the maximum but also with respect to the components close to the maximum. This strategy must be adopted to prevent zigzagging or jamming [1, p. 191]. Moreover, it is highly unlikely that two real numbers would be equal in magnitude. Thus, a component $e_i(\underset{\sim}{x})$ is accepted as "near-maximum" when the difference between e_i and the maximum $e_m(\underset{\sim}{x})$ is less than a predetermined small number δ. The strategy adopted in the program is to assume δ to be 10^{-3} initially and repeating the algorithm with the new value of δ being one hundredth of the previous value till it becomes

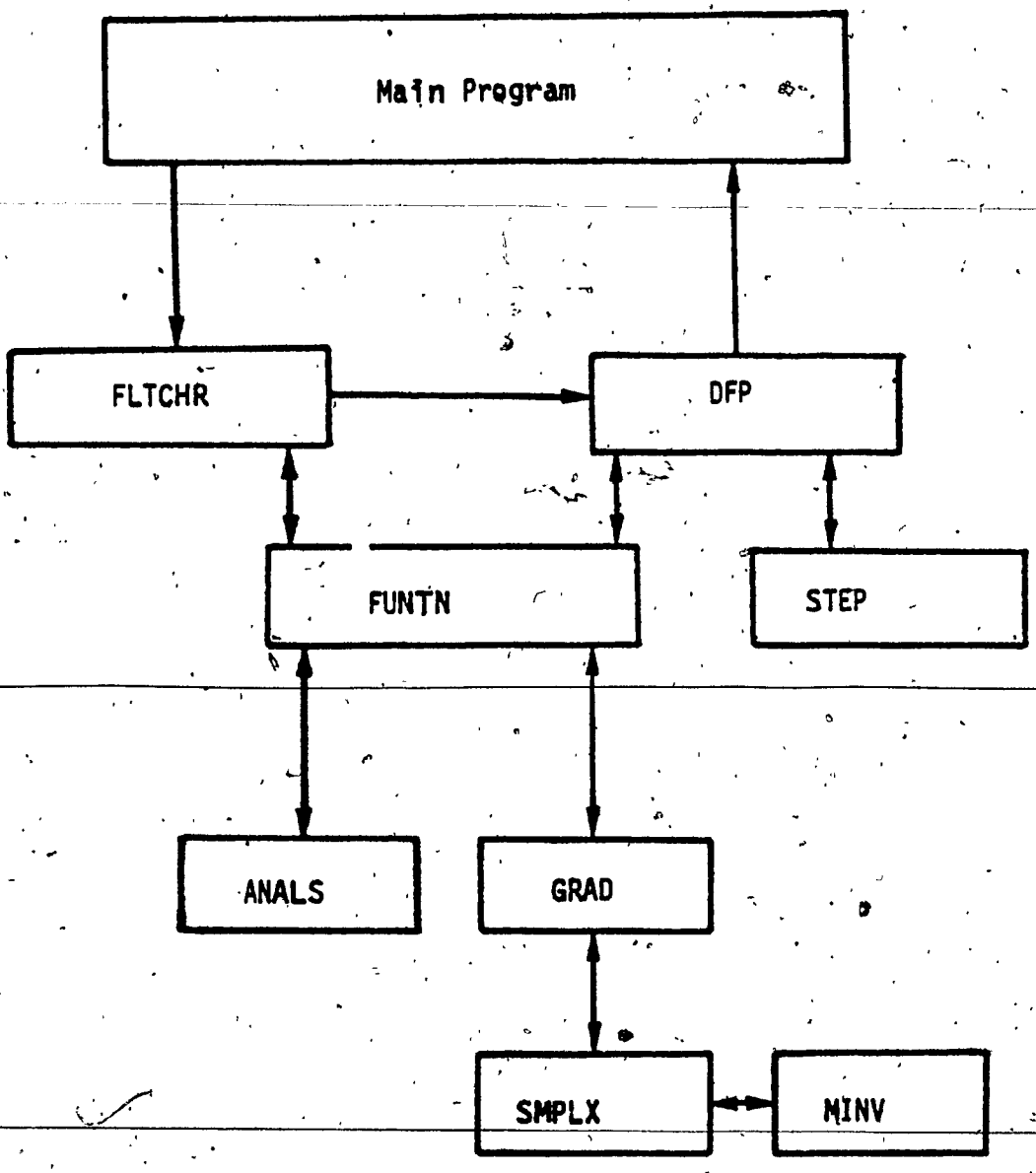


Fig. 2.1 Block Diagram Summarizing the Computer Program Structure.

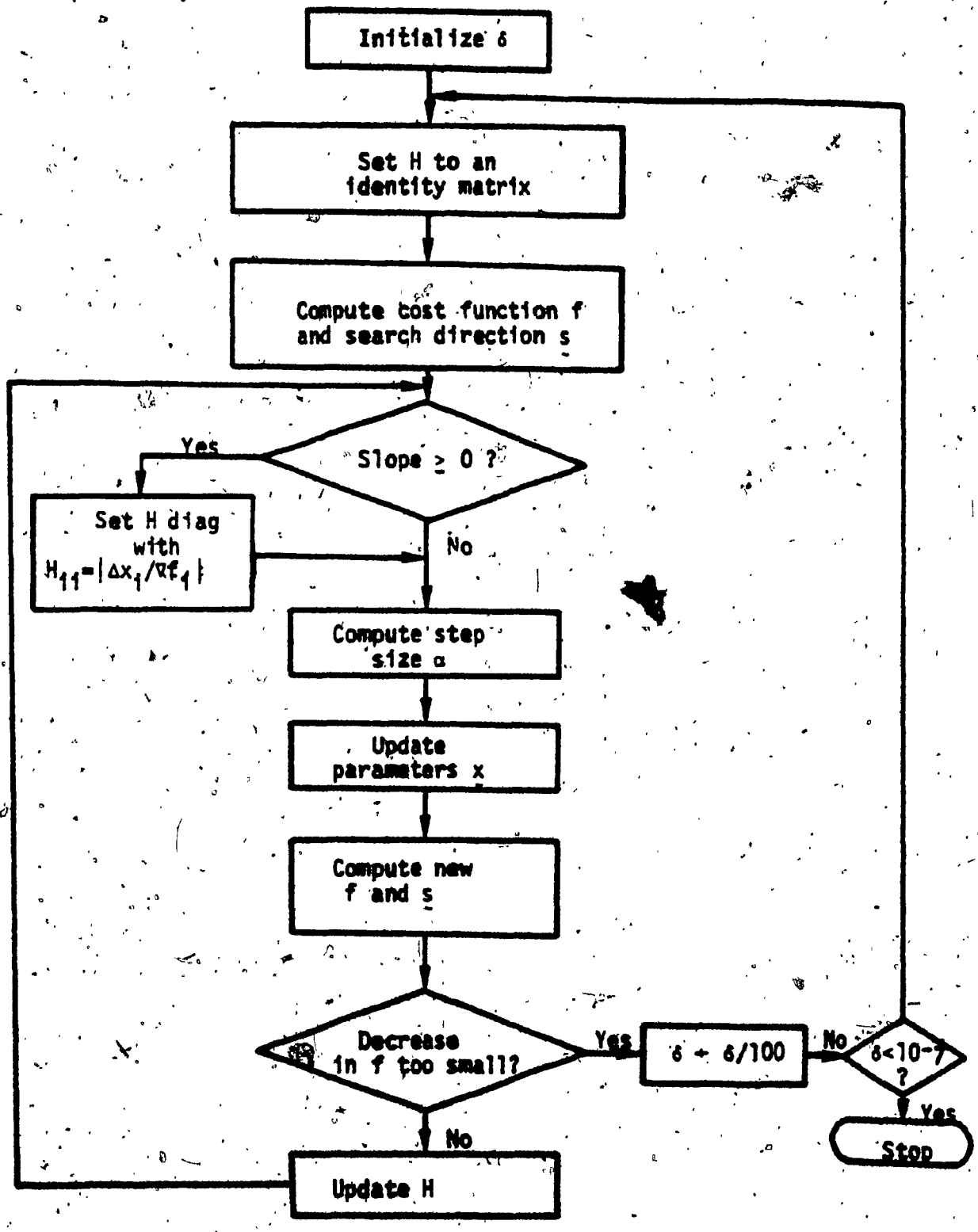


Fig. 2.2 Flowchart of the Minimax Algorithm.

10^{-7}, which is regarded as small enough to be a virtual zero.

In view of this, the following strategy is adopted in finding the descent direction: If the maximum (to within a tolerance of δ) occurs only at a single point, the steepest descent direction is obviously the negative of the corresponding gradient. However, if there is more than one $e_i(\underset{\sim}{x})$ equal in magnitude to the maximum within the accuracy of δ, each gradient vector $\nabla e_i(\underset{\sim}{x})$ is multiplied by $|f_m(\underset{\sim}{x})/e_i(\underset{\sim}{x})|$ where $f_m(\underset{\sim}{x})$ is the maximum, and the simplex method is then used to find the direction of steepest descent.

2.4.4 Various Subroutines Used in the Program

The various subroutines used in the program are listed below and are illustrated in Figure 2.1.

FLTCHR: Minimization method due to Fletcher.

DFP : Minimization method due to Fletcher-Powell.

FUNTN : Computation of the performance function.

ANALS : Analysis of the network to be designed at various sample points.

GRAD : Computation of the descent direction.

SMPLX : Linear programming subroutine.

STEP : Linear search by the method of Golden Sections.

MINV : Matrix inversion subroutine.

2.5 LEAST-p-TH OPTIMIZATION WITH LARGE VALUES OF p

2.5.1 Introduction

An objective function in the discrete least-p-th sense may be written as

$$f_p(\underset{\sim}{x}) = \left[\sum_{i \in I} |e_i(\underset{\sim}{x})|^p \right]^{1/p} \tag{2.36}$$

where I is a set of the sample points, and $e_i(\underset{\sim}{x})$ in general represents a weighted error between a specified function (desired response) and an approximating function (actual response) at a sample point i of a finite set I; and $\underset{\sim}{x}$ is the vector of the parameters. It is easy to see that as $p \to \infty$, for each fixed $\underset{\sim}{x}$,

$$\begin{aligned} f_p(\underset{\sim}{x}) &\to \max_{i \in I} |e_i(\underset{\sim}{x})| \\ &= |e_m| \text{ (say)} \end{aligned} \tag{2.37}$$

Thus the ℓ_p norm for a very large value of p approaches the ℓ_∞ norm. However, for large values of p, each component $e_i(\underset{\sim}{x})$ must be divided by the largest magnitude to avoid ill-conditioning. The corresponding gradient of $f_p(\underset{\sim}{x})$ is given by

$$\nabla f_p(\underset{\sim}{x}) = \left[\sum_{i \in I} |e_i(\underset{\sim}{x})|^p \right]^{(1/p-1)} \Sigma\{|e_i(\underset{\sim}{x})|^{p-1} \nabla |e_i(\underset{\sim}{x})|\} \tag{2.38}$$

where

$$\nabla |e_i(\underset{\sim}{x})| = \mathrm{Re}[e_i^*(\underset{\sim}{x})\, \nabla e_i(\underset{\sim}{x})]/|e_i(\underset{\sim}{x})| \tag{2.39}$$

and the asterisk denotes the complex conjugate. Taking the limit of $\nabla f(\underset{\sim}{x})$ as $p \to \infty$, we obtain

$$\lim_{p \to \infty} \nabla f_p(\underset{\sim}{x}) = (1/q) \sum_{j \in J(\underset{\sim}{x})} |e_j(\underset{\sim}{x})|^{-1} \mathrm{Re}\{e_j^*(\underset{\sim}{x}) \nabla e_j(\underset{\sim}{x})\} \tag{2.40}$$

where the set $J(\underset{\sim}{x}) = \{j : |e_j(\underset{\sim}{x})| = |e_m|\}$, and q is the number of elements in $J(\underset{\sim}{x})$. Thus

$$\begin{aligned} \lim_{p \to \infty} \nabla f_p(\underset{\sim}{x}) &= (1/q) \left[\sum_{i \in J} \nabla |e_i(x)| \right] \\ &= (1/q) \text{ (sum of the gradients of the components equal in magnitude to the maximum value).} \end{aligned} \tag{2.41}$$

2.5.2 The Modified Approach

Instead of following the approach as given in [4] with large values of p, one may minimize $|e_m|$ using the gradient as given in (2.41). The problem of ill-conditioning is thus avoided as one does not have to raise a number to a large power. One of the most important advantages of the modified approach is that the gradients need be computed only for the components equal to the maximum. However, before describing the implementation of the approach, it is shown that the negative of the sum of the gradients as in (2.41), though it represents the limit of $\nabla f_p(\underset{\sim}{x})$ as $p \to \infty$, does not always give a descent direction for the true minimax function, let alone giving the direction of steepest descent. As a simple example, consider

$$e_1(x_1,x_2) = 3x_2 + 1, \quad e_2(x_1,x_2) = x_1 - x_2$$

at the point

$$(x_1,x_2) = (1,0)$$

Then we have

$$I(\underset{\sim}{x}) = \{1,2\},$$

$$\nabla e_1 = [0 \quad 3], \quad \nabla e_2 = [1 \quad -1]$$

and

$$\lim_{p\to\infty} \nabla f_p(\underset{\sim}{x}) = \frac{1}{2}(\nabla e_1 + \nabla e_2)$$

$$= [1/2 \quad 1]$$

However, the directional derivative in the direction $-[1/2 \quad 1]$ from (2.25) is

$$\delta f_M(\underset{\sim}{x}; -[\nabla f_p(\underset{\sim}{x})]) = \max\left\{[0 \quad 3]\begin{bmatrix}-1\\-2\end{bmatrix}, [1 \quad -1]\begin{bmatrix}-1\\-2\end{bmatrix}\right\}$$

$$= 2 > 0$$

Hence, $\lim_{p\to\infty} - \nabla f_p(\underset{\sim}{x})$ is actually an ascent direction for $||f(\cdot)||_\infty$.

Actually, it is possible to give a simple geometric interpretation of the situation. A two-dimensional space will be considered to illustrate the point as it is easier to visualize in such a space. Let

$$\underset{\sim}{h}_1 = [\nabla e_1(\underset{\sim}{x})]^T \operatorname{sgn} e_1(\underset{\sim}{x})$$

and

$$\underset{\sim}{h}_2 = [\nabla e_2(\underset{\sim}{x})]^T \operatorname{sgn} e_2(\underset{\sim}{x})$$

be the two gradient vectors for $|e_1(\underset{\sim}{x})|$ and $|e_2(\underset{\sim}{x})|$ respectively, as illustrated in Figure 2.3. Let $\underset{\sim}{h}_3$ be the negative of the sum of the gradient vector $\underset{\sim}{h}_1$ and $\underset{\sim}{h}_2$. For $\underset{\sim}{h}_3$ to be a descent direction, the inner products of $\underset{\sim}{h}_3$ with $\underset{\sim}{h}_1$ and $\underset{\sim}{h}_2$ must both be negative, as given in (2.26). Thus the projections of $\underset{\sim}{h}_3$ on $\underset{\sim}{h}_1$ and $\underset{\sim}{h}_2$ must both be negative. However, as may be seen, the projection on $\underset{\sim}{h}_2$ is positive, hence $\underset{\sim}{h}_3$ is not a descent direction. In fact, $\underset{\sim}{h}_3$ is an ascent direction. This geometrical interpretation makes it clear that the above example is not just a pathological case.

2.5.3 Practical Implementation

The search direction is simply the negative of the sum of the gradients of the components which are equal in magnitude to the maximum. Like the true minimax case, a component of the performance function vector is considered near-maximum when its relative difference from the maximum is less than a predetermined small positive number δ. To obtain the direction of search, the method of Fletcher and Powell is

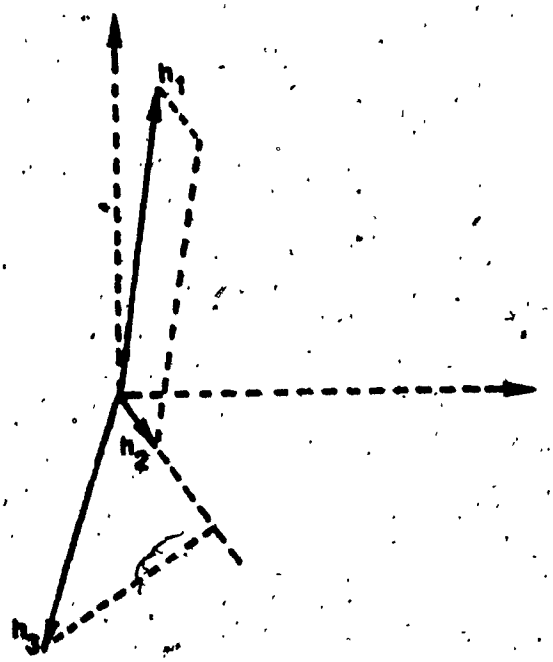


Fig. 2.3 Diagram Illustrating the Resultant of the Sum of Gradients.

adapted as described for the true minimax case. Thus the flow-chart for this near-minimax algorithm is the same as in Figure 2.3, where the search direction $\underset{\sim}{s}$ is given by modifying (2.41) as

$$\underset{\sim}{s} = -(1/q)\left[\sum_{i\in J(\underset{\sim}{x})} \nabla|e_i(\underset{\sim}{x})|(|e_i(\underset{\sim}{x})|/|e_m|)\right] \qquad (2.42)$$

where

$$J(\underset{\sim}{x}) = \{i: (1 - |e_i(\underset{\sim}{x})/e_m|) < \delta\}$$

For the problem discussed in section 2.3.2, where it is necessary to minimize $\max_i e_i(\underset{\sim}{x})$, the expression for the search direction $\underset{\sim}{s}$ is the same as (2.42); however, the set $J(\underset{\sim}{x})$ now becomes

$$J(\underset{\sim}{x}) = \{i: (1 - |e_i(x)/e_m)| < \delta\}$$

2.6 CONCLUSIONS

The minimax optimization problem is studied from a mathematical point of view, and it is shown that a direction of steepest descent for the minimax objective function can be found through linear programming. Using the results, an algorithm based on the algorithms of Fletcher and Fletcher-Powell is proposed for solving such problems.

It is also shown that least-p-th optimization with large values of p can be tackled in a simple manner. The limit of the negative of the gradient of the least-p-th performance function turns out to be an average of the negative of the gradient vectors for the components equal to the maximum. This direction is used in conjunction with Fletcher-Powell's variable metric method to find the search direction. However, this average of the negative of the gradients is shown not to give a descent direction for the true minimax performance function in

general. The most significant advantage of using the near-minimax approach is the ease with which the direction of search is found, even if it is not always a descent direction.

In both the methods, the fact that the gradients need be computed for only a few components of the vector of performance functions (namely those where the maximum is attained within some tolerance), is of prime significance compared with least-p-th techniques.

CHAPTER 3

APPLICATIONS OF MINIMAX OPTIMIZATION TO FILTER DESIGN AND DEVICE MODELLING

3.1 INTRODUCTION

In this Chapter, the feasibility of the true-minimax algorithm presented in Chapter 2 is established by applying it to solve two kinds of design problems, namely, (i) design of filters, and (ii) device modelling.

The examples on the design of filters consist of microwave n-section filters, a passive lumped LC filter, and a lumped distributed RC active filter. These examples have already been tackled by others by using various minimax approaches. The discrete frequency points at which the networks are analyzed to compute the minimax performance function are also taken to be the same as by others [4,5,9], to be able to have a meaningful comparison of our method for efficiency and speed.

The example on device modelling is that of a Schottky-clamped transistor which has not been tackled by the minimax or the least-squares approach so far in the literature. In fact, a computer-aided modelling of such a device has not yet been attempted by others. However, a dc model for a Schottky-clamped transistor should prove useful because of its significance in high speed switching circuits [24, 34]. The various equivalent circuits for analysis in this case are nonlinear and much more complex than in the filter design case, and hence it requires more effort than the straightforward application of a minimax algorithm. These aspects are discussed in more detail at appropriate places.

3.2 APPLICATION TO FILTER DESIGN

Three kinds of filter design problems are tackled in this chapter, and the true-minimax algorithm of Chapter 2 is applied to solve each of those problems.

Example 1 [29] is concerned with the design of 2-section and 3-section transmission line transformers. The problem is to minimize the maximum reflection coefficient ρ of the composite transformer in a certain frequency band. One may like to have minimum ρ for fixed line lengths, and varying the characteristic impedance z_i of each section, or the problem may be to find the minimum reflection coefficient by making each characteristic impedance z_i and each section length ℓ_i as a design parameter.

Example 2 [22] involves the design of an LC matching network. It is required to minimize the maximum reflection coefficient of the circuit for given source and load resistances. The design parameters are all the inductors and capacitors.

Example 3 [30] consists of the design of a lumped-distributed RC active filter, where upper and lower bounds on the response in different frequency regions are specified (namely the pass band and the stop band). It is required to minimize the maximum violation of the specifications in the frequency range. The design parameters are the different resistances and capacitances in the circuit, and the gain of the amplifier.

The gradients of the various components of the performance function vector are computed by the method of adjoint networks [13]. The design parameters are constrained to be positive by taking the actual i-th optimization parameter p_i as

$$p_i = x_i^{(i)}$$

where x_i is the corresponding i-th design parameter.

3.2.1 Examples on Filter Design

Example 1: Quarter-Wave Transmission Line Transformer

The problem considered is the design of 2-section and 3-section 10 Ω to 1 Ω transmission line transformers over a 100 per cent relative bandwidth centred at 1 GHz. The objective is to minimize $\max_i |\rho_i(\underline{x})|$ over 11 frequency points in the band 0.5-1.5 GHz at steps of 0.1 GHz for the network as shown in Fig. 3.1, where $\rho_i(\underline{x})$ is the reflection coefficient at the i-th frequency point.

In the 2-section case the variables are the characteristic impedances z_1 and z_2. The results are shown in Table 3.1 for different starting values of z_1 and z_2. The algorithms are terminated when 0.01 per cent of the optimum value is reached. The lengths of each section are taken as unity when normalized with respect to the quarter wave-length ℓ_q at the centre frequency. For the 3-section case the 11 example points are taken at

{0.5, 0.6, 0.7, 0.77, 0.9, 1.0, 1.1, 1.23, 1.3, 1.4, 1.5} GHz.

The design results with the characteristic impedances z_1, z_2, and z_3

as parameters are given in Table 3.2. Next, the characteristic impedances z_1, z_2, and z_3, as well as the line lengths ℓ_1, ℓ_2, and ℓ_3 are taken as design parameters, and the corresponding results are shown in Table 3.3. The lengths are again normalized with respect to the length ℓ_q at the centre frequency. These problems are also tackled in [4, 5].

Example 2: Lumped LC Transformer

The problem considered is the design of a 3-section lumped LC transformer to match a 1 ohm load to a 3 ohm generator over the normalized frequency range of 0.5-1.79 radiams/sec. The structure of the network is shown in Fig. 3.2. The function to be minimized is

$$f(x) = \max_{1 \le i \le 21} |\rho_1(\underline{x})|$$

over 21 uniformly spaced frequencies ω_i in the pass band, and the design parameter vector is given by

$$x = [L_1 \;\; C_2 \;\; L_3 \;\; C_4 \;\; L_5 \;\; C_6]$$

The results are shown in Table 3.4. The initial values of all the parameters are taken to be 1. The problem is also tackled in [37].

Example 3. Lumped-Distributed Active Filter

The design of a third order lumped-distributed-active lowpass filter [5, 30] as shown in Fig. 3.3 is considered next. It is desired to have an attenuation and ripple in the normalized pass band [0, 0.7] rad/sec. of less than 1 db, while the attenuation in the stop band

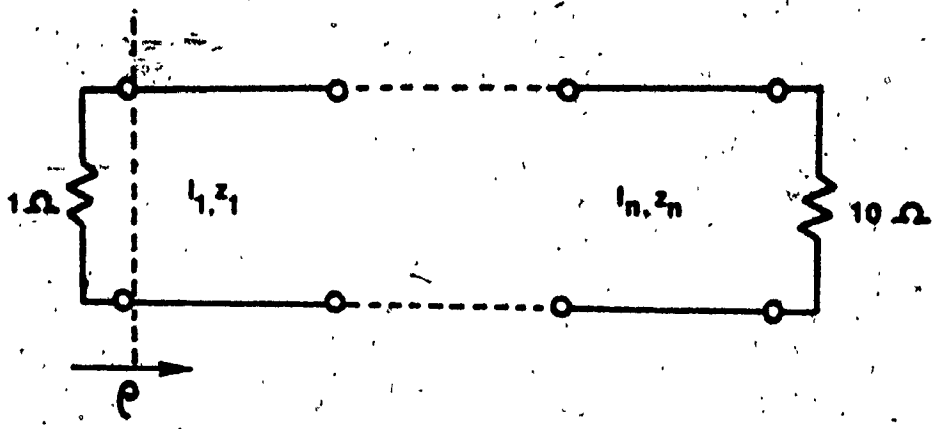


Fig. 3.1 An m-section Resistively Terminated Cascade of Transmission Lines.

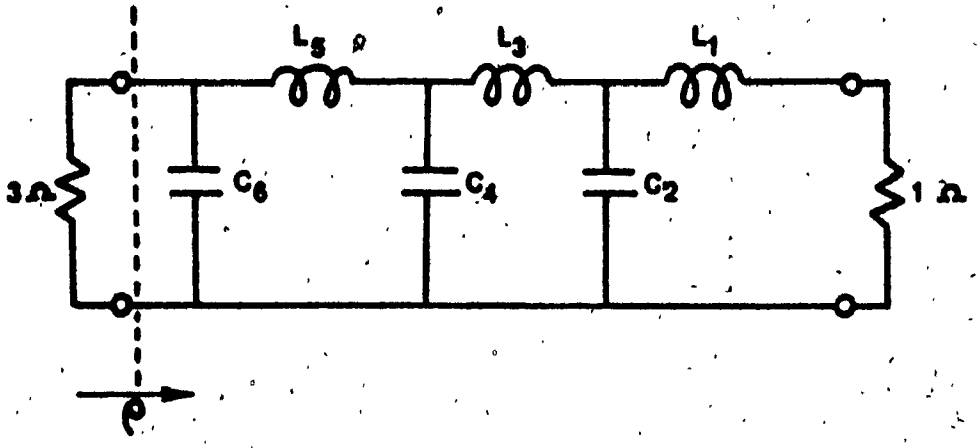


Fig. 3.2 A 3-section Resistively Terminated LC Transformer.

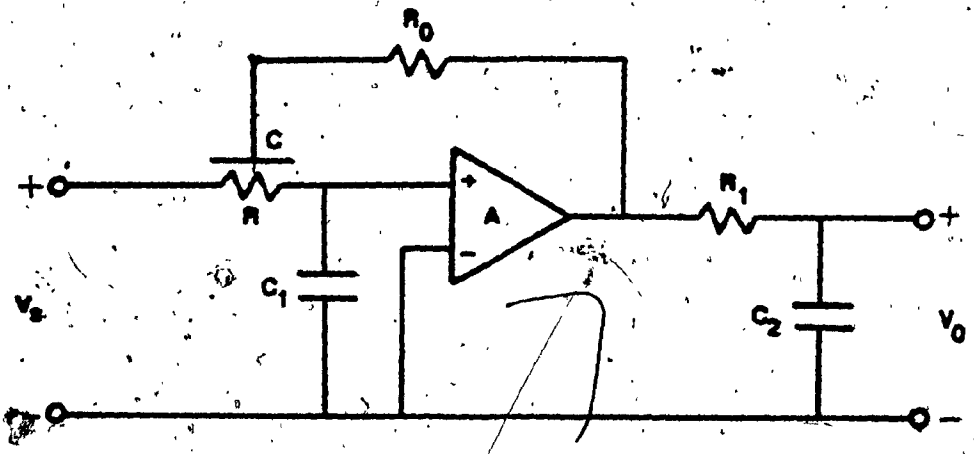


Fig. 3.3 A Lumped-distributed-active Lowpass Filter.

TABLE 3.1: TWO-SECTION TRANSMISSION LINE TRANSFORMER

Initial parameter values	Initial value of max $\lvert\rho_1\rvert$	CPU time in seconds	Function evaluations
$z_1 = 3.5$ $z_2 = 6.0$	0.54574	2.5	42
$z_1 = 1.0$ $z_2 = 3.0$	0.70954	2.5	42

Optimum max $|\rho_1| = 0.42857$

Final parameter values: $z_1 = 2.2353$, $z_2 = 4.4706$

Number of function evaluations in [5] = 56 and 118 respectively

Number of function in [4] = 95 and 126 respectively

TABLE 3.2: THREE-SECTION TRANSMISSION LINE WITH FIXED LENGTHS

Initial parameter values	Initial value of max $\|\rho_1\|$	CPU time in seconds	Function evaluations
$z_1 = 3.0$ $z_2 = 6.0$ $z_3 = 1.5$	0.24785	3.1	25
$z_1 = 10.0$ $z_2 = 3.16228$ $z_3 = 1.0$	0.70930	5.3	41

Optimum max $|\rho_1| = 0.19729$

Final parameter values: $z_1 = 3.17079$, $z_2 = 6.13763$, $z_3 = 1.64014$

Number of function evaluations in [4] = 140 and 175 respectively

TABLE 3.3: THREE-SECTION TRANSMISSION LINE WITH VARIABLE LENGTHS

Initial parameter values	Initial value of max $\|\rho_i\|$	CPU time in seconds	Function evaluations
$z_1 = 1.5$, $\ell_1 = 0.8$, $z_2 = 3.0$, $\ell_2 = 1.2$, $z_3 = 6.0$, $\ell_3 = 0.8$	0.3881	21.8	156
$z_1 = 1.0$, $\ell_1 = 1.0$ $z_2 = 3.16228$, $\ell_2 = 1.0$, $z_3 = 10.0$, $\ell_3 = 1.0$	0.70930	22.3	159

Optimum max $|\rho_i| = 0.19729$

Final parameter values: $z_1 = 1.6299$, $\ell_1 = 0.9982$, $z_2 = 3.1535$, $\ell_2 = 1.0$, $z_3 = 6.0994$, $\ell_3 = 1.0018$

Number of function evaluations in [5] = 645 and 668 respectively

Number of function evaluations in [4] = 498 and 696 respectively

Number of function evaluations in [9] = 160 and 178 respectively

TABLE 3.4: THREE-SECTION LC TRANSFORMER

Final parameter values	Initial value of max $\|\rho_1\|$	CPU time in seconds	Function evaluations
$L_2 = 0.9525$			
$C_2 = 0.9756$			
$L_3 = 0.4249$	0.65508	28	297
$C_4 = 0.7760$			
$L_5 = 0.3392$			
$C_6 = 0.3442$			

Optimum value of max $|\rho_1| = 0.07576$

Number of function evaluations in [37] = 1316

TABLES 3.5: DISTRIBUTED RC-ACTIVE FILTER

Initial parameter values	Final parameter values	Maximum violation in specs. in db		CPU time in seconds	Function evaluations
		Initial	Final		
A = 1.142	1.0034				
R =17.786	21.5716				
C = 0.427	0.4787				
R_0= 1.0	0.6171	1.1533	0.02935	131	935
R_1= 1.0	0.9507				
C_1= 0.067	0.0454				
C_2= 2.62	kept fixed				

CPU time reported in [1] for one least-p-th optimization = 60 secs.,

Total number of least-p-th optimizations in [1] = 5

[1.415, ∞] rad/sec. is desired to be at least 30db. Nine sample points are taken in the pass band, which are {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.65, 0.7}, and 17 sample points in the stop band are taken as {1.415, 1.5, 1.6, 1.7, 1.8, 1.9, 2.0, 2.1, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 2.8, 2.9, 3.0}. The design parameter vector is

$$\underset{\sim}{x} = [A \quad R \quad C \quad R_0 \quad R_1 \quad C_1]^T$$

where A is the gain of the amplifier, R and C are the total resistance and capacitance of the transmission line and, R_0, R_1, and C_1 are as shown in Fig. 3.3. The results obtained are shown in Table 3.5.

3.3 MODELLING OF A SCHOTTKY-CLAMPED TRANSISTOR

3.3.1 Introduction

Schottky barrier diodes are of great significance in the design of logic circuits, since they do not store charge; this is particularly useful in transistor-transistor logic (TTL) circuits. In a TTL circuit they are used to clamp transistors at the edge of saturation, thereby eliminating the delay caused by minority-carrier charge storage [24,34]. The range of operating conditions over which a Schottky-clamped transistor (SCT) is useful depends strongly on the parasite resistances. Thus obtaining a network model for an SCT based on input-output characteristics is of much practical significance, since it is difficult to measure the transistor parameters directly.

The purpose of this example is to illustrate the feasibility of computer-aided modelling of an SCT by using the true minimax algorithm.

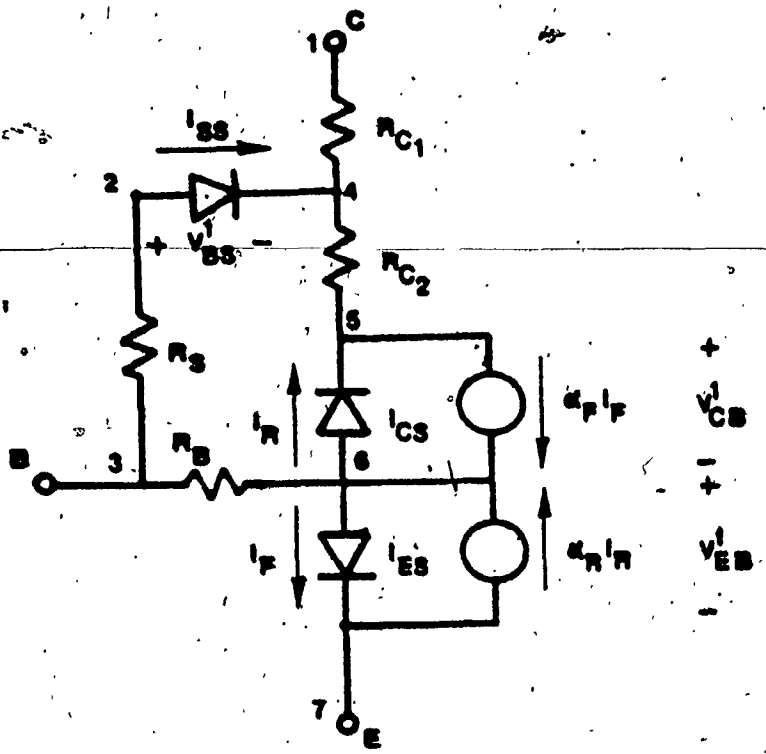
A minimax optimization approach to device modelling insures that the deviation in the model response from the measured response does not exceed a maximum bound. On the other hand with mean-square error minimization, it is possible to have good matching in general at the cost of large mismatching at certain points. Thus a minimax modelling approach should be more desirable compared to a mean-square error minimization.

3.3.2 Initial Model and Measurements

The initial model chosen for the SCT is illustrated in Fig. 3.4. To obtain an accurate estimate of the model parameter values, the model response must be the same as the actual circuit response under varying operating conditions and excitations. With this in mind the following set of measurements are considered. The particular device under test is known as SCH-1, manufactured by Microsystems International Limited, Ottawa, Canada.

Measurement 1. A power supply E_{CE} of 5 volts is applied in series with a load resistance of R_L of 500 ohms. Now, the base is excited by a current source which varies from 0.5 μa to 3.16 μa, driving the device well beyond saturation. The corresponding collector to emitter voltage V_{CE}, and base to emitter voltage V_{BE} are shown in Table 3.6.

Measurement 2. A voltage V_{BE} ranging from 100 mV to 700 mV is applied across base and emitter and the corresponding emitter currents when collector-base is open (I_{EO}) and when collector base is shorted (I_{ES}) are shown in Table 3.7.



$$I_R = I_{CS}\,[\exp(qV'_{CB}/n_c kT)-1]$$

$$I_F = I_{ES}\,[\exp(qV'_{EB}/n_e kT)-1]$$

$$I_S = I_{SS}\,[\exp(qV'_{BS}/n_s kT)-1]$$

Fig. 3.4 A Schottky-Clamped Transistor Model.

Measurement 3. A voltage V_{BC} ranging from 100 mV to 700 mV is applied across base and collector and the corresponding collector currents when emitter-base is open (I_{CO}), and when emitter-base is shorted (I_{CS}) are shown in Table 3.8.

3.3.3 The Performance Function

The overall minimax performance function after being normalized may be expressed as

$$
\begin{aligned}
f_M(\underset{\sim}{x}) = \max \{ & \left| [\hat{V}_{CE}(\underset{\sim}{x}, I_{B_i}) - V_{CE}(I_{B_i})]/V_{CE}(I_{B_i}) \right| , \\
& \left| [\hat{V}_{BE}(\underset{\sim}{x}, I_{B_i}) - V_{BE}(I_{B_i})]/V_{BE}(I_{B_i}) \right| , \\
& \left| [\hat{I}_{ES}(\underset{\sim}{x}, V_{BE_j}) - I_{ES}(V_{BE_j})]/I_{ES}(V_{BE_j}) \right| , \\
& \left| [\hat{I}_{EO}(\underset{\sim}{x}, V_{BE_j}) - I_{EO}(V_{BE_j})]/I_{EO}(V_{BE_j}) \right| , \\
& \left| [\hat{I}_{CS}(\underset{\sim}{x}, V_{BC_k}) - I_{CS}(V_{BC_k})]/I_{CS}(V_{BC_k}) \right| , \\
& \left| [\hat{I}_{CO}(\underset{\sim}{x}, V_{BC_k}) - I_{CO}(V_{BC_k})]/I_{CO}(V_{BC_k}) \right| \} \qquad (3.1) \\
& i \in I, \; j \in J, \text{ and } k \in K,
\end{aligned}
$$

where I, J, and K are the sets of excitation points for I_{B_i}, V_{BE_j}, and V_{BC_k} respectively. The voltages $\hat{V}_{CE}(\underset{\sim}{x}, I_{B_i})$, $\hat{V}_{BE}(\underset{\sim}{x}, I_{B_i})$, and the currents $I_{ES}(\underset{\sim}{x}, V_{BE_j})$, $\hat{I}_{EO}(\underset{\sim}{x}, V_{BE_j})$, $\hat{I}_{CS}(\underset{\sim}{x}, V_{BC_k})$, and $I_{CO}(\underset{\sim}{x}, V_{BC_k})$ are the responses of the circuit model for the corresponding excitations of $V_{CE}(I_{B_i})$, $V_{BE}(I_{B_i})$, $I_{ES}(V_{BE_j})$, $I_{EO}(V_{BE_j})$, $I_{CS}(V_{BC_k})$, and $I_{CO}(V_{BC_k})$. The performance function as given by (3.1) puts equal weights to each component, however, if one desires to put varying weights, each component may be multiplied by a suitable scale factor W_i.

3.3.4 Initial Guess for the Model Parameters

The exact values of the various parameters depend on the make of the device to a certain extent. This in turn depends on the device material and also on the device geometry. Some of the parameters can roughly be guessed by these factors and the device characteristics. The various parameters to be estimated and their initial guesses for the initial model are shown in Table 3.9. However, as may be noted from the table, the parameter values vary in magnitude from 10^{-15} to 10^{2}. This may create performance function surfaces with sharp valleys [23, Ch. 2], which will make the convergence of an optimization algorithm very slow. Thus, the model parameters are scaled such that they are all of the same order of magnitude.

Let c_i be the value of the initial guess for the i-th model parameter, and y_i be the scaled parameter to be optimized. The scaling is done such that the value of y_i become unity for each parameter. Thus

$$x_i = c_i\, y_i \tag{3.2}$$

where now y_i is the optimization parameter, x_i is the model parameter and c_i is the scale factor. In order to insure that the model parameters are all positive, we actually choose

$$x_i = c_i\, p_i^2 \tag{3.3}$$

where $y_i = p_i^2$

3.3.5 Transformation of Parameters

It is known that the forward current amplification factor α_F of the transistor is very close to 1, $(.99<\alpha_F<1.0)$, thus the network model response is highly sensitive to any variations in α_F. It was found that for α_F as a parameter, the optimization algorithm did not converge at all. Therefore, a new parameter is defined as

$$\beta_F = \alpha_F/(1-\alpha_F) \qquad (3.4)$$

such that the range of β_F is from 0 to ∞ for α_F ranging from 0 to 1. Similar transformation is done for α_R.

3.3.6 Evaluation of the Performance Function

As evident from (1), to evaluate the performance function $f_M(\underline{x})$ at a certain value of $\underline{x}$, the model must be analyzed at all the excitation points I_{B_i}, V_{BE_j}, V_{BC_k}, $\forall\ i\varepsilon I$, $j\varepsilon J$, and $k\varepsilon K$. It should be noted however, that for each excitation V_{BE_j}, the emitter current is measured with collector to base shorted, and collector to base opened. Similarly, for each excitation V_{BC_k}, the emitter current is measured with emitter to base shorted and emitter to base opened. Thus for each excitation V_{BE_j} and V_{BC_k}, two different networks are analyzed, so that in all there are five different nonlinear networks to be analyzed. A modified version of the Newton-Raphson method is used to analyze the nonlinear networks by defining a companion network for each network [12, Ch.5]. A program is written to analyze the networks under all the excitations and the corresponding responses and the diode conductances are stored in different arrays to be used for gradient calculations.

Computation of the descent direction of $f_M(\underset{\sim}{x})$ as given in (3.1) involves the calculation of the sensitivities of the excitations with respect to the parameter vector $\underset{\sim}{x}$. The method of adjoint networks [13] is used to evaluate the sensitivities. The linear adjoint network at any excitation is easily obtained since all the nonlinear conductance values are stored during the analyses of the original networks.

3.3.7 Optimization of the Initial Model

Initially the model as shown in Fig. 3.4 is optimized with the variable parameters as shown in Table 3.9. It is found that the matching is not very good in certain regions. The optimized response results are given in Tables 3.10 to 3.15.

3.3.8 Improvement of the Model by Growing Elements

It is expected that in Fig. 3.4, a resistance between nodes 1 and 2 should be created to account for the base-width modulation. Initially, a conductance $G = 1/R$, with zero value is assumed between the nodes and is taken to be a new parameter. Now, with the optimized parameter values of section 3.4.7, the performance function is evaluated and the search direction found. It is found that the search direction due to G is negative and large in magnitude. It is therefore concluded that there is a strong tendency for G to grow in value [14]. Hence, a nonlinear conductance

$$G = r_o I_{ES}[\exp(V_{BE}/V_T)-1]$$

is introduced between nodes 1 and 2, where $V_T = kT/q \approx 0.026$ at 300°K, and

r_o is made a design parameter.

Similarly, R_{C_2} is replaced by $R^1_{C_2}$ [24], such that

$$R^1_{C_2} = R_{C_2} + \alpha I_C$$

and it is found that the constant α has a strong tendency to grow. Hence, α is also made a design parameter. It should be noted that to be more precise, $R^1_{C_2}$ should be treated as a nonlinear function of I_C, however, a linear dependence is assumed for the sake of simplicity. The results of the optimization are given in Tables 3.9 to 3.15.

The final parameter values for both the models are given in Table 3.16.

3.3.9 Discussion

Minimax optimization is applied to the problem of modelling of a Schottky-clamped transistor. The initial model chosen is a very simple one. It is found that the optimized model is a much better match than the initial model whose parameter values are guessed. However, as is evident from the results, the match is not good at some test points. We then attempt to improve upon the model by "growing" certain elements at certain places. It is found that the response of the improved model is slightly better than that of the initial model.

The model may be improved further by adding more nonlinear elements at appropriate places. However, for a model to be of significance, the match should be quite close at all test points, which may not be

obtained by using only lumped elements in the model. Hence, for significantly better results, a distributed model should be taken. Moreover, if it is desired to have a better match at certain points than at others, it can be achieved by multiplying each component in (3.1) by suitable weighting factors. This freedom gives a great deal of flexibility in the modelling.

The minimax performance function f_M as defined in (3.1) finds the maximum deviation from the desired response for all the excitation values over each kind of excitation. However, this kind of a performance function has one drawback, that is, if the error in measurement at a single point is significantly higher than all the other points, the optimum result may be drastically affected by that. However, if we define f_M as

$$f_M = \max_i \{E_i\}$$

where E_i is the least-squares performance function for the i-th excitation source, the effect of an isolated error in measurement is considerably reduced. For the modelling problem under consideration, it is a viable alternative and worth looking into.

TABLE 3.6: MEASUREMENT 1

I_B	V_{CE}	V_{BE}
0.5 μa	4.977 v	.605 v
1.0	4.950	.627
2.0	4.900	.650
5.0	4.710	.678
10.0	4.380	.700
20.0	3.660	.723
50.0	1.580	.758
53.7	1.450	.762
100.	.735	.768
200.	.690	.780
500.	.630	.818
1.0 ma	.410	.866
1.846 ma	.335	.957
3.13	.311	.989
2.66	.280	1.04
3.16	.256	1.1

TABLE 3.7: MEASUREMENT 2

V_{BE}	I_{ES}	I_{EO}
100 mV	$.400 \times 10^{-9}$ A	0 A
150	$.120 \times 10^{-8}$	$.800 \times 10^{-11}$
200	$.180 \times 10^{-8}$	$.100 \times 10^{-10}$
250	$.230 \times 10^{-8}$	$.590 \times 10^{-10}$
300	$.480 \times 10^{-8}$	$.300 \times 10^{-9}$
350	$.170 \times 10^{-7}$	$.210 \times 10^{-9}$
400	$.650 \times 10^{-7}$	$.150 \times 10^{-7}$
450	$.220 \times 10^{-6}$	$.940 \times 10^{-7}$
500	$.780 \times 10^{-6}$	$.740 \times 10^{-6}$
550	$.363 \times 10^{-5}$	$.350 \times 10^{-5}$
600	$.203 \times 10^{-4}$	$.203 \times 10^{-4}$
650	$.149 \times 10^{-3}$	$.149 \times 10^{-3}$
700	$.167 \times 10^{-3}$	$.168 \times 10^{-2}$

TABLE 3.8: MEASUREMENT 3

V_{BC}	I_{CS}	I_{CO}
100 mV	205 ma	205 - na
150	520 na	520
200	1.3 µa	1.3 µa
250	4.2	4.2
300	18.	18.
350	73.	73.
400	215.	215.
450	454.	454.
500	777.	777.
550	1.143 ma	1.143 ma
600	1.561	1.561
650	2.04	2.04
700	2.68	2.68

TABLE 3.9: STARTING PARAMETER VALUES FOR THE INITIAL MODEL

Parameters	Initial estimate
I_{SS}	7×10^{-9} A
n_s	1.1
I_{CS}	10^{-15} A
n_c	1.4
I_{ES}	10^{-13} A
n_e	1.1
α_F	0.99
α_R	0.50
R_{C1}	16Ω
R_{C2}	10Ω
R_S	15Ω
R_B	200Ω

TABLE 3.10: OPTIMIZATION RESULTS FOR V_{BE}

V_{BE} measured	V_{BE} initial	V_{BE} optimized (initial model)	V_{BE} optimized (improved model)
0.605 V	.573 V	.647 V	.643 V
0.627	.593	.669	.664
0.650	.613	.691	.686
0.678	.639	.721	.716
0.700	.660	.744	.739
0.723	.682	.768	.763
0.758	.714	.802	.801
0.762	.717	.805	.804
0.768	.738	.810	.810
0.780	.739	.811	.811
0.818	.741	.813	.814
0.866	.744	.815	.817
0.957	.748	.819	.822
0.989	.749	.821	.824
1.040	.752	.823	.827
1.100	.754	.825	.830

TABLE 3.11: OPTIMIZATION RESULTS FOR V_{CE}

V_{CE} measured	V_{CE} initial	V_{CE} optimized (initial model)	V_{CE} optimized (improved model)
4.977 v	4.975 v	4.962 v	4.944 v
4.950	4.950	4.924	4.890
4.900	4.900	4.848	4.782
4.710	4.752	4.622	4.469
4.380	4.505	4.245	3.977
3.660	4.009	3.491	3.092
1.580	2.525	1.228	1.028
1.450	2.342	.950	.824
0.735	0.663	.527	.531
0.690	0.602	.469	.475
0.630	0.563	.413	.418
0.410	0.537	.371	.376
0.335	0.510	.329	.332
0.311	0.503	.318	.320
0.280	0.492	.301	.300
0.256	0.482	.285	.284

TABLE 3.12: OPTIMIZATION RESULTS FOR I_{ES}

I_{ES} measured	I_{ES} initial	I_{ES} optimized (initial model)	I_{ES} optimized (improved model)
$.400\times10^{-9}$A	$.320\times10^{-11}$A	$.226\times10^{-11}$A	$.233\times10^{-11}$ A
$.120\times10^{-8}$	$.156\times10^{-10}$	$.984\times10^{-11}$	$.103\times10^{-10}$
$.180\times10^{-8}$	$.904\times10^{-10}$	$.482\times10^{-10}$	$.512\times10^{-10}$
$.230\times10^{-8}$	$.519\times10^{-9}$	$.234\times10^{-9}$	$.253\times10^{-9}$
$.480\times10^{-7}$	$.298\times10^{-8}$	$.114\times10^{-8}$	$.125\times10^{-8}$
$.170\times10^{-7}$	$.171\times10^{-7}$	$.553\times10^{-8}$	$.615\times10^{-8}$
$.650\times10^{-7}$	$.985\times10^{-7}$	$.268\times10^{-7}$	$.303\times10^{-7}$
$.220\times10^{-6}$	$.566\times10^{-6}$	$.130\times10^{-6}$	$.149\times10^{-6}$
$.780\times10^{-6}$	$.391\times10^{-5}$	$.633\times10^{-6}$	$.737\times10^{-6}$
$.363\times10^{-5}$	$.224\times10^{-4}$	$.354\times10^{-5}$	$.416\times10^{-5}$
$.203\times10^{-4}$	$.127\times10^{-3}$	$.172\times10^{-4}$	$.205\times10^{-4}$
$.149\times10^{-3}$	$.706\times10^{-3}$	$.833\times10^{-4}$	$.101\times10^{-3}$
$.167\times10^{-3}$	$.336\times10^{-2}$	$.399\times10^{-3}$	$.488\times10^{-3}$

TABLE 3.13: OPTIMIZATION RESULTS FOR I_{EO}

I_{EO} measured	I_{EO} initial	I_{EO} optimized (initial model)	I_{EO} optimized (improved model)
0 A	$.320\times10^{-11}$ A	$.226\times10^{-11}$ A	$.233\times10^{-11}$ A
$.800\times10^{-11}$	$.188\times10^{-10}$	$.112\times10^{-10}$	$.118\times10^{-10}$
$.100\times10^{-10}$	$.108\times10^{-9}$	$.556\times10^{-10}$	$.589\times10^{-10}$
$.590\times10^{-10}$	$.625\times10^{-9}$	$.270\times10^{-9}$	$.291\times10^{-9}$
$.300\times10^{-9}$	$.359\times10^{-8}$	$.131\times10^{-8}$	$.143\times10^{-8}$
$.210\times10^{-8}$	$.206\times10^{-7}$	$.637\times10^{-8}$	$.705\times10^{-8}$
$.150\times10^{-7}$	$.118\times10^{-6}$	$.309\times10^{-7}$	$.346\times10^{-7}$
$.940\times10^{-7}$	$.681\times10^{-6}$	$.150\times10^{-6}$	$.169\times10^{-6}$
$.740\times10^{-6}$	$.391\times10^{-5}$	$.730\times10^{-6}$	$.828\times10^{-6}$
$.350\times10^{-5}$	$.224\times10^{-4}$	$.354\times10^{-5}$	$.405\times10^{-5}$
$.230\times10^{-4}$	$.128\times10^{-3}$	$.172\times10^{-4}$	$.198\times10^{-4}$
$.149\times10^{-3}$	$.706\times10^{-3}$	$.833\times10^{-4}$	$.965\times10^{-4}$
$.168\times10^{-3}$	$.336\times10^{-2}$	$.399\times10^{-3}$	$.464\times10^{-3}$

TABLE 3.14: OPTIMIZATION RESULTS FOR I_{CS}

I_{CS} measured	I_{CS} initial	I_{CS} optimized (initial model)	I_{CS} optimized (improved model)
$.205 \times 10^{-6}$ A	$.889 \times 10^{-7}$ A	$.365 \times 10^{-7}$ A	$.374 \times 10^{-7}$ A
$.520 \times 10^{-6}$	$.109 \times 10^{-5}$	$.153 \times 10^{-6}$	$.160 \times 10^{-6}$
$.130 \times 10^{-5}$	$.755 \times 10^{-5}$	$.454 \times 10^{-6}$	$.482 \times 10^{-6}$
$.420 \times 10^{-5}$	$.418 \times 10^{-4}$	$.132 \times 10^{-5}$	$.142 \times 10^{-5}$
$.180 \times 10^{-4}$	$.202 \times 10^{-3}$	$.399 \times 10^{-5}$	$.437 \times 10^{-5}$
$.730 \times 10^{-4}$	$.683 \times 10^{-3}$	$.115 \times 10^{-4}$	$.127 \times 10^{-4}$
$.215 \times 10^{-3}$	$.155 \times 10^{-2}$	$.326 \times 10^{-4}$	$.366 \times 10^{-4}$
$.454 \times 10^{-3}$	$.266 \times 10^{-2}$	$.904 \times 10^{-4}$	$.102 \times 10^{-3}$
$.777 \times 10^{-3}$	$.392 \times 10^{-2}$	$.235 \times 10^{-3}$	$.261 \times 10^{-3}$
$.114 \times 10^{-2}$	$.526 \times 10^{-2}$	$.543 \times 10^{-3}$	$.585 \times 10^{-3}$
$.156 \times 10^{-2}$	$.665 \times 10^{-2}$	$.107 \times 10^{-2}$	$.111 \times 10^{-2}$
$.204 \times 10^{-2}$	$.808 \times 10^{-2}$	$.181 \times 10^{-2}$	$.182 \times 10^{-2}$
$.268 \times 10^{-2}$	$.955 \times 10^{-2}$	$.272 \times 10^{-2}$	$.267 \times 10^{-2}$

TABLE 3.15: OPTIMIZATION RESULTS FOR I_{CO}

I_{CO} measured	I_{CO} initial	I_{CO} optimized (initial model)	I_{CO} optimized (improved model)
$.205 \times 10^{-6}$ A	$.224 \times 10^{-6}$ A	$.385 \times 10^{-7}$ A	$.384 \times 10^{-7}$ A
$.520 \times 10^{-6}$	$.132 \times 10^{-5}$	$.173 \times 10^{-6}$	$.166 \times 10^{-6}$
$.130 \times 10^{-5}$	$.755 \times 10^{-5}$	$.454 \times 10^{-6}$	$.485 \times 10^{-6}$
$.420 \times 10^{-5}$	$.418 \times 10^{-4}$	$.132 \times 10^{-5}$	$.143 \times 10^{-5}$
$.180 \times 10^{-4}$	$.202 \times 10^{-3}$	$.399 \times 10^{-5}$	$.437 \times 10^{-5}$
$.730 \times 10^{-4}$	$.686 \times 10^{-3}$	$.115 \times 10^{-4}$	$.127 \times 10^{-4}$
$.215 \times 10^{-3}$	$.155 \times 10^{-2}$	$.326 \times 10^{-4}$	$.366 \times 10^{-4}$
$.454 \times 10^{-3}$	$.266 \times 10^{-2}$	$.904 \times 10^{-4}$	$.102 \times 10^{-3}$
$.777 \times 10^{-3}$	$.392 \times 10^{-2}$	$.235 \times 10^{-3}$	$.261 \times 10^{-3}$
$.114 \times 10^{-2}$	$.526 \times 10^{-2}$	$.543 \times 10^{-3}$	$.585 \times 10^{-3}$
$.156 \times 10^{-2}$	$.665 \times 10^{-2}$	$.107 \times 10^{-2}$	$.111 \times 10^{-2}$
$.204 \times 10^{-2}$	$.808 \times 10^{-2}$	$.181 \times 10^{-2}$	$.182 \times 10^{-2}$
$.268 \times 10^{-2}$	$.955 \times 10^{-2}$	$.272 \times 10^{-2}$	$.267 \times 10^{-2}$

TABLE 3.16: THE OPTIMIZED PARAMETER VALUES

Parameters	optimized values (initial model)	optimized values (improved model)
I_{CS}	$.699 \times 10^{-9}$ A	$.699 \times 10^{-9}$ A
n_s	1.408	1.790
I_{CS}	$.101 \times 10^{-15}$ A	$.100 \times 10^{-15}$ A
n_c	1.4	1.409
I_{ES}	$.101 \times 10^{-12}$ A	$.101 \times 10^{-12}$ A
n_e	1.16	1.209
α_F	.997	.993
α_R	.499	.499
R_{C_1}	15.1Ω	14.0Ω
R_{C_2}	9.4Ω	9.0Ω
R_S	11.	14.158Ω
R_B	19.8Ω	202Ω
R.	x	$1.\times 10^{15}\Omega$
α	x	.999

3.4 CONCLUSIONS

The true-minimax algorithm proposed in Chapter 2 is applied to design different kinds of filters and to the modelling of a Schottky-clamped transistor. Fletcher's method and a modified version of the Fletcher-Powell algorithm are used in the optimization algorithm. It is found that the algorithm is quite efficient and reliable, and compares favourably with the existing methods [4, 5, 9, 25, 35]. It has also been found that the optimization can be carried out by using only the modified version of Fletcher-Powell method without any appreciable loss of efficiency. The fact that the gradients need be computed only at a few components of the performance function vector is of significance for many design problems, particularly so in device modelling where many nonlinear networks are to be analyzed.

CHAPTER 4

CONSTRAINED MINIMAX OPTIMIZATION WITH APPLICATION TO NONLINEAR PROGRAMMING

minimax problem. The basic idea here is the same as in [6]. We then proceed beyond the results in [6] by showing how equality constraints can also be incorporated in the same general framework.

Consider the following problem:

4.2.1 Nonlinear Programming Problem with Inequality Constraints (NLP1):

Given f_i; $R^n \to R$, $i = 0, \ldots, m$, f_i continuously differentiable.

Minimize $f_0(\underset{\sim}{x})$ subject to $f_i(\underset{\sim}{x}) \leq 0$, $i = 1, \ldots, m$. (4.1)

With this problem one can associate the following minimax problem.

4.2.2 Unconstrained Minimax Optimization Problem (UMP1):

$$\text{Minimize } F(\underset{\sim}{x}) \triangleq \max_{1 \leq i \leq m} f_0(\underset{\sim}{x}),\ f_0(\underset{\sim}{x}) + \alpha_i f_i(\underset{\sim}{x}) \tag{4.2}$$

where α_i, $i = 1, \ldots, m$ are given positive constants.

Theorem 4.1. Any solution of UMP1, if it is feasible for NLP1, solves NLP1. If the α_i's are sufficiently large, any point satisfying the Kuhn-Tucker conditions for NLP1 also satisfies the necessary condition (2.27) applied to UMP1.

Proof: Suppose $\bar{\underset{\sim}{x}}$ solves UMP1 and is feasible for NLP1. Then $f_i(\bar{\underset{\sim}{x}}) \leq 0$ for $1 \leq i \leq m$; therefore $F(\bar{\underset{\sim}{x}}) = f_0(\bar{\underset{\sim}{x}})$. Now suppose $\underset{\sim}{x}$ is any feasible point for NLP1. Since $f_i(\underset{\sim}{x}) \leq 0$ for $1 \leq i \leq m$, we have

$$\begin{aligned} f_0(\underset{\sim}{x}) &= F(\underset{\sim}{x}) \quad \text{(by the feasibility of } \underset{\sim}{x}) \\ &\geq F(\bar{\underset{\sim}{x}}) \quad \text{(since } \bar{\underset{\sim}{x}} \text{ solves UMP1)} \\ &\geq f_0(\bar{\underset{\sim}{x}}) \quad \text{(by the feasibility of } \bar{\underset{\sim}{x}}) \end{aligned} \tag{4.3}$$

Hence $\bar{\underset{\sim}{x}}$ solves NLP1.

To prove the second part of the theorem, suppose that, at some $\underset{\sim}{x} \in R^n$, there exist nonnegative constants $\lambda_1, \ldots, \lambda_m$ such that

$$\nabla f_0(\underset{\sim}{x}) + \sum_{i \in I(\underset{\sim}{x})} \lambda_i \nabla f_i(\underset{\sim}{x}) = 0 \tag{4.4}$$

where $I(\underset{\sim}{x})$ is the index set of the active constraints at $\underset{\sim}{x}$. Define $\beta_i = \lambda_i / \alpha_i$, then (4.4) can be rewritten as

$$\nabla f_0(\underset{\sim}{x}) + \sum_{i \in I(\underset{\sim}{x})} \beta_i \alpha_i \nabla f_i(\underset{\sim}{x}) = 0 \tag{4.5}$$

By making the α_i's sufficiently large, we can insure that

$$\sum_{i \in I(\underset{\sim}{x})} \beta_i < 1 \tag{4.6}$$

in which case (4.5) can be rewritten as

$$(1 - \sum_{i \in I(\underset{\sim}{x})} \beta_i) \nabla f_0(\underset{\sim}{x}) + \sum_{i \in I(\underset{\sim}{x})} \beta_i (\nabla f_0(\underset{\sim}{x}) + \alpha_i \nabla f_i(\underset{\sim}{x})) = 0 \tag{4.7}$$

But this readily implies that

$$\min_{||\underset{\sim}{h}||=1} \max_{i \in I(\underset{\sim}{x})} \langle \nabla f_0(\underset{\sim}{x}), \underset{\sim}{h} \rangle, \langle \nabla f_0(\underset{\sim}{x}) + \alpha_i \nabla f_i(\underset{\sim}{x}), \underset{\sim}{h} \rangle \geq 0 \tag{4.8}$$

It is routine to verify that (4.8) is the necessary condition contained in (2.27), for $\underset{\sim}{x}$ to be a solution of UMP1. ⊠

The second part of Theorem 4.1 is similar to the results of [6], except that in [6], the authors stop with the necessary conditions (4.7). The first part of Theorem 4.1, though perhaps obvious, does not appear to have been previously stated so explicitly.

Theorem 4.1 suggests a method of solving a nonlinear programming problem with inequality constraints by first transforming it to an

unconstrained minimax problem. However, the biggest stumbling block in this technique is the choice of the parameters α_i. If the α_i's are chosen too small, UMP1 may not have any solution at all (i.e. the infimum is approached as $||\underset{\sim}{x}|| \to \infty$, but is not actually attained), or the solution may not be feasible. On the other hand, if the α_i's are chosen too large, ill-conditioning may result. The inequality (4.6) provides a guideline for selecting the α_i's: they should be large enough that (4.6) holds, but not so large that the problem becomes ill-conditioned.

Let us now turn to the case of both equality and inequality constraints being present.

4.2.3 Nonlinear Programming Problem with Equality and Inequality Constraints (NLP2)

Given $f_i: R^n \to R$, $i = 0, \ldots, m$, $h_i: R^q \to R$, $i = 1, \ldots, k$, f_i and h_i continuously differentiable.

Minimize $f_0(\underset{\sim}{x})$ subject to $f_i(\underset{\sim}{x}) \leq 0$, $i = 1, \ldots, m$, $h_i(\underset{\sim}{x}) = 0$, $i = 1, \ldots, k$. The only apparent way of extending Theorem 4.1 to cover the case of equality constraints is the rather dubious one of separating each equality constraint into a pair of inequality constraints, i.e. by trying to minimize

$$\max_{1 \leq i \leq n,\ 1 \leq j \leq k} f_0(\underset{\sim}{x}), f_0(\underset{\sim}{x}) + \alpha_i f_i(\underset{\sim}{x}), f_0(\underset{\sim}{x}) + \beta_i h_i(\underset{\sim}{x}), f_0(\underset{\sim}{x}) - \gamma_i h_i(\underset{\sim}{x})$$

However, this procedure in practice runs into severe computational difficulties.

An alternative procedure is to transform NLP2 into a minimax problem with equality constraints, to be described next.

4.3 MINIMAX OPTIMIZATION PROBLEM WITH EQUALITY CONSTRAINTS (CMP1)

4.3.1 Introduction

Minimize $F(\underline{x})$, defined in (4.3), subject to the constraints $h_i(\underline{x}) = 0$, $i = 1, \ldots, k$.

In this section a procedure is given whereby CMP1 is transformed into a series of unconstrained minimax problems. The procedure given below can be thought of as an extension, to the minimax case, of the results of [27,31]. Before giving this procedure, however, we prove some preliminary results relating the solutions of CMP1 to those of NLP2.

Fact 1: A necessary condition for $\underline{x}$ to be a solution of CMP1 is that there exist real constants $\lambda_1, \ldots, \lambda_k$ such that

$$\delta F(\underline{x};\underline{v}) + \sum_{j=1}^{k} \lambda_j \nabla h_j(\underline{x})\, \underline{v} \geq 0 \quad \forall\, \underline{v} \in R^n \tag{4.9}$$

where $\delta F(\underline{x};\underline{v})$ denotes the directional derivative of F at $\underline{x}$ in the direction $\underline{v}$.

Fact 1 is simply an extension of the standard Lagrange multiplier technique to the case where the performance function is just directionally differentiable. The proof of Fact 1 can be carried out exactly as in [33, Ch. 3] and is hence omitted.

Theorem 4.2 below relates the solutions of CMP1 with those of NLP1.

Theorem 4.2: Suppose $\underline{x}$ solves CMP1 and is feasible for NLP2; then $\underline{x}$ solves NLP2. Suppose $\underline{x}$ satisfies the Kuhn-Tucker necessary conditions for NLP2; then, provided the constants a_i, $i = 1, \ldots, m$ are sufficiently large, $\underline{x}$ also satisfies the necessary condition (4.9) for CMP1.

Proof: The first part of the theorem is very easy to prove. To prove the second part, suppose $\underset{\sim}{x}$ satisfies the Kuhn-Tucker necessary conditions for NLP2; i.e., suppose there exist nonnegative constants β_i and real constants γ_j, such that

$$\nabla f_0(\underset{\sim}{x}) + \sum_{i \in I(\underset{\sim}{x})} \beta_i \nabla f_i(\underset{\sim}{x}) + \sum_{j=1}^{k} \gamma_j \nabla h_j(\underset{\sim}{x}) = 0 \tag{4.10}$$

where

$$I(\underset{\sim}{x}) = \{i: f_i(\underset{\sim}{x}) = 0\} \tag{4.11}$$

denotes the index set of the active constraints at $\underset{\sim}{x}$. We prove that (4.9) holds by contradiction. Note first of all that since $\underset{\sim}{x}$ is feasible for NLP2, we have, by (2.25) that

$$\delta F(\underset{\sim}{x};\underset{\sim}{v}) = \max_{i \in I(\underset{\sim}{x})} \nabla f_0(\underset{\sim}{x})\underset{\sim}{v}, (\nabla f_0(\underset{\sim}{x}) + \alpha_i \nabla f_i(\underset{\sim}{x}))\underset{\sim}{v} \tag{4.12}$$

Now suppose the α_i's are sufficiently large, so that

$$\sum_{i \in I(\underset{\sim}{x})} \beta_i/\alpha_i < 1 \tag{4.13}$$

and define

$$\eta_i = \beta_i/\alpha_i,\ i \in I(\underset{\sim}{x});\ \eta_0 = 1 - \sum_{i \in I(\underset{\sim}{x})} \eta_i \tag{4.14}$$

Then (4.10) can be rewritten as

$$\eta_0 \nabla f_0(\underset{\sim}{x}) + \sum_{i \in I(\underset{\sim}{x})} \eta_i(\nabla f_0(\underset{\sim}{x}) + \alpha_i \nabla f_i(\underset{\sim}{x})) + \sum_{j=1}^{k} \gamma_i \nabla h_i(\underset{\sim}{x}) = 0 \tag{4.15}$$

Next, suppose by way of contradiction that (4.9) does not hold. This means that for a particular choice of $\underset{\sim}{v}$, (4.9) fails to hold no matter what values are assigned to the constants λ_j. Let $\underset{\sim}{v}$ be so chosen, and

let $\lambda_j = \gamma_j/\mu$, where μ equals the number of elements in $I(x)$ plus one. Then the violation of (4.9) implies that

$$\nabla f_0(\underset{\sim}{x})\underset{\sim}{v} + \sum_{j=1}^{k} \lambda_j \nabla h_j(\underset{\sim}{x})\underset{\sim}{v} < 0 \tag{4.16}$$

$$(\nabla f_0(\underset{\sim}{x}) + \alpha_i \nabla f_i(\underset{\sim}{x}))\underset{\sim}{v} + \sum_{j=1}^{k} \lambda_j \nabla h_j(\underset{\sim}{x})\underset{\sim}{v} < 0 \quad \forall\ i \in I(\underset{\sim}{x}) \tag{4.17}$$

However, (4.16) and (4.17) together violate (4.15). Hence (4.9) holds. ∎

Before converting CNP1 to a sequence of unconstrained problems following the approach of Morrison [31], we summarize this method first.

4.3.2 Morrison's Algorithm

A nonlinear programming problem with equality constraints may be expressed as

Problem 1: Minimize $f(\underset{\sim}{x})$ subject to the constraints $h_j(\underset{\sim}{x}) = 0$, $j = 1, \ldots, \ell$.

Let $\bar{\underset{\sim}{x}}$ be the optimum solution of Problem 1, and consider the associated problem.

Problem 2: Minimize $[f(\underset{\sim}{x}) - M]^2$ subject to the constraints $h_j(\underset{\sim}{x}) = 0$, $j = 1, \ldots, \ell$.

Now it can be shown that if $M \leq f(\bar{\underset{\sim}{x}})$, any solution of Problem 1 is a solution of Problem 2 and vice versa. Problem 2 is now converted to an unconstrained problem by introducing a penalty function, as follows:

Problem 3: Minimize $P(\underset{\sim}{x}, M) = [f(\underset{\sim}{x}) - M]^2 + \sum_{j=1}^{\ell} \omega_j [h_j(\underset{\sim}{x})]^2$ (4.18a)

where $\omega_j \geq 0$.

The algorithm as given in [31] is as follows:

(a) Start with an estimate M_1 with $f(\underset{\sim}{\bar{x}}) \geq M_1$, that is, start with an optimistic guess for M_1. Let $k = 1$.

(b) Solve Problem 3, with $M = M_k$, obtaining a solution $\underset{\sim}{x}_k$. Then

$$P(\underset{\sim}{x}_k, M_k) = [f(\underset{\sim}{x}_k) - M_k]^2 + \sum_{j=1}^{\ell} \omega_j [h_j(\underset{\sim}{x}_k)]^2 \qquad (4.18b)$$

$$= \inf_{\underset{\sim}{x}} P(\underset{\sim}{x}, M_k)$$

(c) Let $M_{k+1} = M_k + P^{\dagger}$ (4.19a)

(d) If $M_{k+1} = M_k$, stop. Otherwise go back to (b) with $k + 1$ replacing k.

The basis for the algorithm is that (i) if M_1 is an exact estimate, i.e., if $M_1 = f(\underset{\sim}{\bar{x}})$, then $\underset{\sim}{x}_1$ is a solution of Problem 2, and (ii) more generally if M_1 is an optimistic guess, then M_k is an optimistic guess for all k, and M_k's converge monotonically to $f(\underset{\sim}{\bar{x}})$ under some mild conditions on f and $\underset{\sim}{h}$.

Kowalik et al. [27] have shown that under certain conditions a much faster updating formula than (4.1) can be applied in step (c). Denoting the new updating parameter by M^T (tangent parameter), it is expressed as

$$M^T_{k+1} = M^T_k + P \;/\; [f(\underset{\sim}{x}_k) - M_k] \qquad (4.19b)$$

Comparing (4.2) with (4.1), it is apparent that since $[f(\underset{\sim}{x}_k) - M_k]^2 \leq P(\underset{\sim}{x}_k, M_k)$, we have $M^T_k \geq M_k$ for all k, so that the M^T_k's converge faster than the M_k's.

4.3.3 Constrained Minimax as a Sequence of Unconstrained Minimax Problems (UMP2)

Morrison's algorithm, as described in section 4.3.2, can be applied as is to a minimax problem with equality constraints, by redefining $P(\underset{\sim}{x}_k, M_k)$ of (4.18) as

$$P(\underset{\sim}{x}, M_k) = [F(\underset{\sim}{x}) - M_k]^2 + \sum_{j=1}^{\ell} \omega_j [g_j(\underset{\sim}{x})]^2 \qquad (4.20)$$

$$1 \leq i \leq m$$

where

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

where $e_i(\underset{\sim}{x})$ is the same as in section 2.2.2. Any of the algorithms in Chapter 2 to minimize an unconstrained minimax function can be applied to minimize $P(\underset{\sim}{x}, M_k)$. It is easy to show that Morrison's algorithm with the updating formula of (4.19a) is valid in this case. However, it is not apparent whether the faster updating formula of (4.19b) is also valid. We now show that a similar formula is valid for the present UMP2, even though $P(\underset{\sim}{x}, M_k)$ is no longer differentiable. First, three preliminary results are proven.

Lemma 4.1. Suppose $\bar{\underset{\sim}{x}}$ is a solution to CMP1. Then there exist nonnegative constants λ_0, λ_i, $i \in I(\bar{\underset{\sim}{x}})$, and real constants γ_j, $j = 1, \ldots, k$, such that $\lambda_0 = 0$ if $\bar{\underset{\sim}{x}}$ is nonfeasible for NLP2, and

$$\lambda_0 + \sum_{i \in I(\bar{\underset{\sim}{x}})} \lambda_i = 1 \qquad (4.21)$$

$$\lambda_0 \nabla f_0(\bar{\underset{\sim}{x}}) + \sum_{i \in I(\underset{\sim}{x})} \lambda_i (\nabla f_0(\bar{\underset{\sim}{x}}) + \alpha_i \nabla f_i(\bar{\underset{\sim}{x}})) + \sum_{j=1}^{k} \gamma_j \nabla h_j(\bar{\underset{\sim}{x}}) = 0 \qquad (4.22)$$

Proof: CMP1 is equivalent to the following problem:

Minimize $f = \phi$ subject to

$$-\phi+f_0(\underset{\sim}{x})\leq 0,\ -\phi+f_0(\underset{\sim}{x})+\alpha_i f_i(\underset{\sim}{x})\leq 0,\ i=1,\ldots,m$$

$$h_j(\underset{\sim}{x}) = 0,\ j=1,\ldots,k$$

Applying the Kuhn-Tucker conditions [26, Ch.1] to the above problem yields (4.21) and (4.22)

Lemma 4.2. Suppose $\bar{\underset{\sim}{x}}_\ell$ is a solution of UMP2 with $M = M_\ell$. Then there exist nonnegative constants $\lambda_{0,\ell}$, $\lambda_{i,\ell}$, $i\in I(\bar{\underset{\sim}{x}}_\ell)$ such that

$$\lambda_{0,\ell} + \sum_{i\in I(\bar{\underset{\sim}{x}}_\ell)} \lambda_{i,\ell} = 1 \tag{4.23}$$

$$\lambda_{0,\ell}\nabla f_0(\bar{\underset{\sim}{x}}_\ell) + \sum_{i\in I(\bar{\underset{\sim}{x}}_\ell)} \lambda_{i,\ell}(\nabla f_0(\underset{\sim}{x}_\ell) + \alpha_i\nabla f_i(\underset{\sim}{x}_\ell)) + \sum_{j=1}^{k} \{\omega_j h_j(\underset{\sim}{x}_\ell)/(F(\underset{\sim}{x}_\ell)-M_\ell)\}\nabla h_j(\underset{\sim}{x}_\ell) = 0 \tag{4.24}$$

Proof: UMP2 can be formulated as the following nonlinear programming problem:

Minimize $(\phi-M_\ell)^2 + \sum_{j=1}^{k} \omega_j h_j^2(\underset{\sim}{x})$, subject to

$$-\phi + f_0(\underset{\sim}{x}) \leq 0,\ -\phi + f_0(\underset{\sim}{x}) + \alpha_i f_i(\underset{\sim}{x}) \leq 0$$

Applying the Kuhn-Tucker conditions to the above problem reveals that there are nonnegative constants β_0, $\beta_{i,\ell}$, $i\in I(\bar{\underset{\sim}{x}}_\ell)$ such that

$$2(\phi-M_\ell) - \beta_0 - \sum_{i\in I(\bar{\underset{\sim}{x}}_\ell)} \beta_i = 0 \tag{4.25}$$

$$\sum_{j=1}^{k} 2\omega_j h_j(\bar{\underset{\sim}{x}}_\ell)\nabla h_j(\bar{\underset{\sim}{x}}_\ell) + \beta_0\nabla f_0(\bar{\underset{\sim}{x}}_\ell) + \sum_{i\in I(\bar{\underset{\sim}{x}}_\ell)} \beta_i(\nabla f_0(\bar{\underset{\sim}{x}}_\ell) + \alpha_i\nabla f_i(\bar{\underset{\sim}{x}}_\ell)) = \underset{\sim}{0} \tag{4.26}$$

Equations (4.23) and (4.24) now follow by defining

$$\lambda_{0,\ell} = \beta_0/(g(\bar{x}_\ell)-M_\ell);\ \lambda_{i,\ell} = \beta_i/(g(\bar{x}_\ell)-M_\ell) \tag{4.27}$$

where we have made use of the obvious fact that, at the optimum, ϕ equals $F(\bar{x}_\ell)$.

Lemma 4.3. Let $\bar{x}$ be the solution of CMP1, and suppose $\nabla f_0(\bar{x})$, $\nabla f_i(\bar{x})$, $i\in I(\bar{x})$, and $\nabla h_j(\bar{x})$, $j = 1,\ldots,k$ are linearly independent. Let $\{M_\ell\}$, $\{\bar{x}_\ell\}$ be sequences such that $\bar{x}_\ell$ solves UMP2 with $M_\ell \to F(\bar{x})$ and $\bar{x}_\ell \to \bar{x}$ as $\ell \to \infty$. Then $\lambda'_{0,\ell} \to \lambda_0$, $\lambda'_{i,\ell} \to \lambda_i$ for $i\in I(\bar{x})$ as $\ell \to \infty$, and moreover

$$h_j(\bar{x}_\ell)/(F(\bar{x}_\ell) - M_\ell) \to \gamma_j$$

as $\ell \to \infty$, where all constants are as in Lemmas 4.1 and 4.2.

Proof: The proof is self evident, once it is observed that $I(\bar{x}_\ell)$ is a subset of $I(\bar{x})$ for $\bar{x}_\ell$ sufficiently close to $\bar{x}$.

Finally, we are ready to prove the validity of the faster updating formula.

Theorem 4.3. Let $L(x, \gamma)$ denote the Lagrangian

$$L(x,\gamma) = F(x) + \sum_{j=1}^{k} \gamma_j h_j(x) \tag{4.28}$$

Let $\bar{x}$ denote the solution of CMP1, and let $\bar{\gamma}_j$, $j = 1,\ldots,k$ be the constants such that (4.22) is satisfied. Let S be a set in $R^n \times R^k$ containing $\bar{x}$, $\bar{\gamma}$ in its interior such that the following two conditions hold.

(i) L has the saddle point property in S; i.e.,

$$L(\bar{x},\gamma) = L(\bar{x},\bar{\gamma}) \le L(x,\bar{\gamma}) \tag{4.29}$$

(11) Whenever $\gamma^* \in R^k$ has the property that $(\underset{\sim}{x},\gamma^*) \in S$ for some $\underset{\sim}{x} \in R^n$, there exists a unique $\underset{\sim}{x}^* \in R^n$ such that

(a) $(\underset{\sim}{x}^*,\gamma^*) \in S$

(b) $L(\underset{\sim}{x}^*,\gamma^*) \leq L(\underset{\sim}{x},\gamma^*)$ whenever $(\underset{\sim}{x},\gamma^*) \in S$

(c) the conditions (4.22) and (4.23) hold (with $\bar{\underset{\sim}{x}}$ replaced by $\underset{\sim}{x}^*$) for some nonnegative constants λ_0^*, λ_i^*, $i \in I(\underset{\sim}{x}^*)$.

Under these conditions, whenever $M_\ell < F(\bar{\underset{\sim}{x}})$, and is sufficiently close to $F(\bar{\underset{\sim}{x}})$, the following expression is valid:

$$M_\ell + P(\bar{\underset{\sim}{x}}_\ell, M_\ell)/[F(\underset{\sim}{x}) - M_\ell] \leq F(\bar{\underset{\sim}{x}}) \qquad (4.30)$$

The proof is omitted since it exactly follows that in [27].

Example: As an illustration of the procedure, the following nonlinear programming problem is solved.

Minimize $f(\underset{\sim}{x}) = 4x_1 - x_2^2 - 12$

Subject to

$h_1(\underset{\sim}{x}) \triangleq 25 - x_1^2 - x_2^2 = 0$

$g_2(\underset{\sim}{x}) \triangleq x_1^2 + x_2^2 - 10x_1 - 10x_2 + 34 \leq 0$

$g_3(\underset{\sim}{x}) \triangleq -x_1 \leq 0$

$g_4(\underset{\sim}{x}) \triangleq -x_2 \leq 0$

The starting point was chosen as

$x_1 = x_2 = 1$, $M_1 = -50$, $\alpha_i = 10$ for $i = 1,2,3$.

The final result, produced after three steps (using the updating formula in (4.19a)) was

$M_3 = -31.992$ (= optimum value)

$\bar{x}_1 = 1.001$, $\bar{x}_2 = 4.899$

$h_1(\bar{x}) < 2.16 \times 10^{-11}$

The near minimax method given in Chapter 2 was used as the minimization algorithm, where the total number of search directions was 32. The CPU time was 0.4 second.

By way of comparison, the same problem when solved by SUMT [23,Ch.5] gives the following results.

The ϵ-parameter was changed from 1 to 1/16384 in eight stages. The final result was

$h_1(x_0) = -3.95 \times 10^{-3}$

$f(x_0) = -31.990$

The CPU time was about 1.0 second.

4.4 MINIMAX OPTIMIZATION PROBLEM WITH BOTH EQUALITY AND INEQUALITY CONSTRAINTS (CMP2)

It is well known that the inequality constraints can be formally converted to equality constraints [26, Ch. 6] by using the Heaviside function $H(t)$, where $H(t) = 1$, $t > 0$, $H(t) = 0$, $t \leq 0$. For example, the inequality constraint

$$g(x) \leq 0 \tag{4.31}$$

is equivalent to the equality constraint

$$h(x) = g(x)\, H[g(x)] = 0 \tag{4.32}$$

The above approach has been used by Kowalik et al. [27] to extend Morrison's method [31] to include inequality constraints.

Note that $h(\underset{\sim}{x})$ in (4.32) need not be everywhere differentiable, however, if we define

$$h^2(\underset{\sim}{x}) = g^2(\underset{\sim}{x})\ H[g(\underset{\sim}{x})] \qquad (4.33)$$

then $h^2(\underset{\sim}{x})$ can easily be shown to be continuously differentiable. Thus, the method discussed in section 4.3 for CMP1 can be easily extended to a minimax problem involving both equality and inequality constraints.

4.5 CONCLUSIONS

In this chapter, several interrelationships between minimax optimization problems and nonlinear programming problems have been presented. One of the important aspects is a mathematical justification for adapting the algorithms of [27,31] to the case of nondifferentiable minimax objective functions. Thus, any constrained minimax optimization problem can be solved as a sequence of unconstrained minimax optimization problems. As a result, a procedure has been derived for transforming nonlinear programming problems into a sequence of unconstrained minimax optimization problems, which appears to be competitive with other existing techniques in NLP.

CHAPTER 5

CONSTRAINED MINIMAX PROBLEM AS A SEQUENCE OF LEAST-SQUARES TYPE OPTIMIZATIONS

5.1 INTRODUCTION

The problem of minimax optimization under constraints has not yet been satisfactorily resolved in the literature. One idea that has been put forward [8] is to first convert the constrained minimax problem into a constrained nonlinear programming problem, and then to convert back to an unconstrained minimax problem - which is quite a cumbersome and unreliable procedure. The situation is particularly bad in the case of equality constraints, which in [8] are converted into two inequality constraints.

In this chapter we propose a method whereby a constrained minimax problem is converted to a sequence of problems involving the unconstrained minimization of a sequence of least-squares type of objective functions. This procedure offers two advantages: (i) the original constrained problem is transformed into a sequence of unconstrained problems, and (ii) the original nondifferentiable minimax performance function is replaced by least-squares type performance functions that are everywhere differentiable; hence any efficient gradient technique can be used to carry out the sequence of unconstrained minimizations. Based on this method, two algorithms are proposed for constrained or unconstrained minimax optimization. In general, the second algorithm converges faster than the first. But, when applied to unconstrained minimax problems, both algorithms converge much faster than previously proposed techniques [4, 9, 25, 35].

5.2 BASIC ALGORITHM

An unconstrained minimax problem can be stated as

Problem 5.1 Minimize

$$F(\underline{x}) = \max_{i \in I} f_i(\underline{x}) \tag{5.1}$$

where $I = \{1, \ldots, n\}$ is a finite set of integers.

A constrained problem can be stated as

Problem 5.2 Minimize $F(\underline{x})$, subject to the constraints

$$g_j(\underline{x}) \leq 0, \quad j \in J$$

$$h_\ell(\underline{x}) = 0, \quad \ell \in L$$

Problem 5.2 is equivalent to the following nonlinear programming problem [41];

Problem 5.3 Minimize ϕ, subject to

$$f_i(\underline{x}) - \phi \leq 0, \quad i \in I$$

$$g_j(\underline{x}) \leq 0, \quad j \in J$$

$$h_\ell(\underline{x}) = 0, \quad \ell \in L$$

Let us now consider the following problem:

Problem 5.4 Minimize

$$P(\underline{x}, \phi) = \sum_{i \in I(\underline{x})} [f_i(\underline{x}) - \phi]^2 + \sum_{j \in J(\underline{x})} \omega_j g_j^2(\underline{x}) + \sum_{\ell \in L} v_\ell h_\ell^2(\underline{x}) \tag{5.2}$$

where ϕ is a prespecified constant, ω_j, $j \in J$ and v_i, $i \in I$ are preassigned weights, and

$$I(\underset{\sim}{x}) = \{i \in I: f_i(\underset{\sim}{x}) > \phi\}$$

$$J(\underset{\sim}{x}) = \{j \in J: g_j(\underset{\sim}{x}) > 0\}$$

By convention, empty sums are taken to equal zero.

It can be easily shown that, for each constant ϕ, $P(\underset{\sim}{x}, \phi)$ is continuously differentiable with respect to $\underset{\sim}{x}$ [19]. A result is now proven that forms the basis of the algorithm.

Proposition 5.1 Let $\bar{\underset{\sim}{x}}$ denote the solution of Problem 5.2, and let the constant ϕ_0 be chosen such that $\phi_0 \leq F(\bar{\underset{\sim}{x}})$. Let $\bar{\underset{\sim}{x}}_0$ denote the solution of Problem 5.4 with $\phi = \phi_0$. Then

$$[P(\bar{\underset{\sim}{x}}_0, \phi_0)/n]^{\frac{1}{2}} \leq F(\bar{\underset{\sim}{x}}) - \phi_0 \tag{5.3}$$

where n is the number of integers in the set I.

Proof Since $\bar{\underset{\sim}{x}}_0$ solves Problem 5.4,

$$P(\bar{\underset{\sim}{x}}_0, \phi_0) \leq P(\bar{\underset{\sim}{x}}, \phi_0)$$

$$= \sum_{i \in I(\bar{\underset{\sim}{x}})} [f_i(\bar{\underset{\sim}{x}}) - \phi_0]^2 \tag{5.4}$$

since all the constraints of Problem 5.2 are satisfied at $\bar{\underset{\sim}{x}}$. Now

$$\sum_{i \in I(\bar{\underset{\sim}{x}})} [f_i(\bar{\underset{\sim}{x}}) - \phi_0]^2 \leq m\,[F(\bar{\underset{\sim}{x}}) - \phi_0]^2$$

$$\leq n\,[F(\bar{\underset{\sim}{x}}) - \phi_0]^2 \tag{5.5}$$

where m is the number of integers in the set $I(\bar{\underset{\sim}{x}})$. Combining (5.4) and

(5.5) gives

$$P(\bar{x}_0, \phi_0) \leq n^{\frac{1}{2}} [F(\bar{x}) - \phi_0]^2 \tag{5.6}$$

clearly (5.6) is equivalent to (5.3). □

We now reformulate Problem 5.2 as a sequence of unconstrained minimization problems, as follows:

Problem 5 Choose $\phi_0 \leq F(\bar{x})$, and minimize

$$P(x, \phi_k) = \sum_{i \in I(x)} [f_i(x) - \phi_k]^2 + \sum_{j \in J(x)} \omega_j g_j^2(x) + \sum_{\ell \in L} v_\ell h_\ell^2(x) \tag{5.7}$$

where the constant ϕ_k is updated according to the formula

$$\phi_{k+1} = \phi_k + [P(\bar{x}_k, \phi_k)/n]^{\frac{1}{2}} \tag{5.8}$$

where $\bar{x}_k$ minimizes $P(x, \phi_k)$.

In view of Proposition 5.1, it is clear that if ϕ_0 is an optimistic guess, i.e. $\phi_0 \leq F(\bar{x})$, then ϕ_k is an optimistic guess for all k. Thus $\{\phi_k\}$ is a monotonically increasing sequence that is bounded above, and hence has a definite limit as $k \to \infty$. We now show that, under a relatively mild assumption, ϕ_k in fact converges to $F(\bar{x})$ as $k \to \infty$.

Assumption 5.1 Let $\bar{x}(q)$ denote the solution to the problem

Minimize $F(x)$, subject to

$$g_j(x) \leq q_j, \quad j \in J$$

$$h_\ell(x) = q_\ell \quad \ell \in L$$

The assumption is that $F(\bar{x}(q))$ is a continuous function of q at $q=0$.

Proposition 5.2 If Assumption 5.1 holds, then $\phi_k \to F(\bar{x})$ as $k \to \infty$.

Proof Assumption 5.1 implies that, given any $\epsilon > 0$, there exists a $\delta>0$ such that

$$|F(\bar{x}(q)) - F(\bar{x}(0))| < \epsilon \text{ whenever } ||q|| < \delta. \tag{5.9}$$

Note that $\bar{x}=x(0)$. To prove that $\phi_k \to F(\bar{x})$, let $\epsilon>0$ be any given constant. We shall show that, for k sufficiently large, we have $F(\bar{x}) - \phi_k < 2\epsilon$. First of all, given ϵ, pick $\delta>0$ such that (5.9) is satisfied, and then pick k_o sufficiently large that

$$\phi_{k+1} - \phi_k < \min \{\epsilon, \delta(\omega/n)^{\frac{1}{2}}\} \quad \forall k \geq k_o \tag{5.10}$$

where

$$\omega = \min_{j\in J,\ \ell\in L} \omega_j, \nu_\ell$$

Such a k_o can always be found since $\{\phi_k\}$ is a Cauchy sequence. Now, (5.10) implies that

$$[P_k(\bar{x}_k, \phi_k)/n]^{\frac{1}{2}} < \delta(\omega/n)^{\frac{1}{2}}$$

or, in other words,

$$\sum_{i\in I(\bar{x}_k)} [f_i(\bar{x}_k) - \phi_k]^2 + \sum_{j\in J(\bar{x}_k)} \omega_j\, g_j^2(\bar{x}_k) + \sum_{\ell\in L} \nu_\ell\, h_\ell^2(\bar{x}_k) < \delta^2\omega$$

Hence, in particular, we have

$$g_j(\bar{x}_k) < \delta \qquad \forall j\in J \tag{5.11a}$$

$$h_\ell(\bar{x}_k) < \delta \qquad \forall \ell\in L \tag{5.11b}$$

similarly, we also have

$$f_i(\bar{x}_k) - \phi_k < \varepsilon \qquad \forall i \in I(\bar{x}_k)$$

which in turn implies that

$$F(\bar{x}_k) - \phi_k < \varepsilon \tag{5.12}$$

Finally, by Assumption 5.1, (5.11) implies that

$$|F(\bar{x}_k) - F(\bar{x})| < \varepsilon \tag{5.13}$$

The inequalities (5.12) and (5.13) together establish that

$$F(\bar{x}) - \phi_k < 2\varepsilon$$

Hence the result is proved. ⊠

Corollary 5.1 As $k \to \infty$, $F(\bar{x}_k) \to F(\bar{x})$, and $P_k(\bar{x}_k, \phi_k) \to 0$.

Proof Obvious.

For the sake of clarity, we again summarize the algorithm for constrained minimax problems.

ALGORITHM 1

Step 1. Choose $\phi_0 \leq F(\bar{x})$, where $\bar{x}$ is a solution to Problem 2. set k=0.

Step 2. Minimize $P(x, \phi_k)$. Denote the solution by $\bar{x}_k$.

Step 3. Set $\phi_{k+1} = \phi_k + [P(\bar{x}_k, \phi_k)/n]^{\frac{1}{2}}$

Step 4. If $\phi_{k+1} - \phi_k < \varepsilon$, where ε is a prespecified small number, STOP. Otherwise set $k \to k+1$ and go to step 2.

5.3 A FASTER ALGORITHM

In this section, we present a different updating formula for the ϕ_k's, and it is shown that, at least as ϕ_k becomes close to $F(\bar{x})$, the new formula accelerates the convergence of ϕ_k to $F(\bar{x})$.

In order to bring out the basic idea clearly, and to avoid confusion, we repeat here the updating formula given in section 5.2. Given $\phi_0 < F(\bar{x})$, the procedure is to minimize $P(x, \phi_0)$, and to set

$$\phi_1 = \phi_0 + [P(\bar{x}_0, \phi_0)/n]^{\frac{1}{2}} \tag{5.14}$$

where $\bar{x}_0$ minimizes $P(x, \phi_0)$. By contrast, the new updating formula suggested here is to set

$$\psi_1 = \phi_0 + P(\bar{x}_0, \phi_0)/\{\sum_{i \in I(\bar{x}_0)} [f_i(\bar{x}_0) - \phi_0]\} \tag{5.15}$$

First of all, Proposition 5.3 below shows that $\psi \geq \phi_1$, so that (5.15) is indeed a faster updating formula than (14).

Proposition 5.3 $\psi_1 \geq \phi_1$.

Proof We show instead that $\psi_1 - \phi_0 \geq \phi_1 - \phi_0$, we have

$$\phi_1 - \phi_0 = [P(\bar{x}_0, \phi_0)/n]^{\frac{1}{2}}$$

$$\psi_1 - \phi_0 = P(\bar{x}_0, \phi_0)/\{\sum_{i \in I(\bar{x}_0)} [f_i(\bar{x}_0) - \phi_0]\}$$

Hence showing that $\psi_1 - \phi_0 \geq \phi_1 - \phi_0$ is equivalent to showing that

$$\sum_{i \in I(x_0)} f_i(\bar{x}_0) - \phi_0 \leq [n\, P(\bar{x}_0, \phi_0)]^{\frac{1}{2}}$$

However, Schwarz's inequality [1, Appendix 1] shows that

$$\sum_{i\in I(\underset{\sim}{x}_0)} f_i(\underset{\sim}{\bar{x}}_0) - \phi_0 \leq n^{\frac{1}{2}} \cdot \left[\sum_{i\in I(\underset{\sim}{x}_0)} (f_i(\underset{\sim}{\bar{x}}_0) - \phi_0)^2\right]^{\frac{1}{2}}$$

Since the second term on the right-hand side is less than or equal to $P(\underset{\sim}{\bar{x}}_0, \phi_0)$, the result is proved. ☒

Hence the updating formula (5.15) gives a higher value than (5.14). However, it is not always true that if $\phi_0 \leq F(\underset{\sim}{\bar{x}})$, then $\psi_1 \leq F(\underset{\sim}{\bar{x}})$. In other words, the updating formula (15) may overshoot $F(\underset{\sim}{\bar{x}})$. The next series of propositions is devoted to showing that, under some relatively mild conditions, this does not happen, and that ψ_1 is also an optimistic estimate.

Proposition 5.4 At the solution $\underset{\sim}{\bar{x}}$ of problem 2, there exist non-negative constants $\bar{\lambda}_i$, $i\in I$, $\bar{\gamma}_j$, $j\in J$, and real constants $\bar{\beta}_\ell$, $\ell\in L$, such that

$$\sum_{i\in I_0(\underset{\sim}{\bar{x}})} \bar{\lambda}_i \nabla f_i(\underset{\sim}{\bar{x}}) + \sum_{j\in J_0(\underset{\sim}{\bar{x}})} \bar{\gamma}_j \nabla g_j(\underset{\sim}{\bar{x}}) + \sum_{\ell\in L} \bar{\beta}_\ell \nabla h_\ell(\underset{\sim}{\bar{x}}) = \underset{\sim}{0} \qquad (5.16)$$

$$\sum_{i\in I_0(\underset{\sim}{\bar{x}})} \bar{\lambda}_i = 1$$

where

$$I_0(\underset{\sim}{\bar{x}}) = \{i\in I : f_i(\underset{\sim}{\bar{x}}) = F(\underset{\sim}{\bar{x}})\}$$

$$J_0(\underset{\sim}{\bar{x}}) = \{j\in J : g_j(\underset{\sim}{\bar{x}}) = 0\}$$

Proof Immediate by applying the Kuhn-Tucker condition to Problem 5.2 [2].

Proposition 5.5 Suppose $\bar{x}_k$ is a solution to Problem 5.4 corresponding to $\phi=\phi_k$. Then

$$\sum_{i\in I(\bar{x}_k)} \nabla f_i(\bar{x}_k)(f_i(\bar{x}_k) - \phi_k) + \sum_{j\in J(\bar{x}_k)} \omega_j g_j(\bar{x}_k) \nabla g_j(\bar{x}_k)$$

$$+ \sum_{\ell\in L} v_\ell h_\ell(\bar{x}_k) \nabla h_\ell(\bar{x}_k) = 0 \qquad (5.17)$$

Proof Equation (5.17) simply states that the gradient of $P(x, \phi_k)$ is zero at $x=x_k$.

Proposition 5.6 Suppose $\{\phi_k\}$ is a sequence converging to $F(\bar{x})$ and that the corresponding sequence $\{\bar{x}_k\}$ converges to $\bar{x}$. Suppose further that $\nabla f_i(\bar{x})$, $i\in I_0(\bar{x})$, $\nabla g_j(\bar{x})$, $j\in J_0(\bar{x})$, $\nabla h_\ell(\bar{x})$, $\ell\in L$ form a linearly independent set of vectors. Then

$$[f_i(\bar{x}_k) - \phi_k]/p_k \to \lambda_i$$

$$\omega_j g_j(\bar{x}_k)/p_k \to \gamma_j$$

$$v_\ell h_\ell(\bar{x}_k)/p_k \to \beta_\ell$$

where

$$p_k = \sum_{i\in I(\bar{x}_k)} [f_i(\bar{x}_k) - \phi_k]$$

Proof The result follows immediately by comparing (5.16) and (5.17), and by observing that for $\bar{x}_k$ sufficiently close to $\bar{x}$, $I(\bar{x}_k)$ is a subset of $I_0(\bar{x})$, and $J(\bar{x}_k)$ is a subset of $J_0(\bar{x})$.

Proposition 5.7 Let

$$L(\underset{\sim}{x}, \lambda, \gamma, \beta) = \sum_{i\in I(\underset{\sim}{x})} \lambda_i [f_i(\underset{\sim}{x}) - \phi_k] + \sum_{j\in J(\underset{\sim}{x})} \gamma_j g_j(\underset{\sim}{x}) + \sum_{\ell\in L} \beta_\ell h_\ell(\underset{\sim}{x}) \tag{5.18}$$

Suppose that, whenever $\underset{\sim}{x}^*$ is the unique solution of the equation

$$\nabla L(\underset{\sim}{x}, \underset{\sim}{\lambda}^*, \underset{\sim}{\gamma}^*, \underset{\sim}{\beta}^*) = 0 \tag{5.19}$$

corresponding to fixed $\underset{\sim}{\lambda}^*$, $\underset{\sim}{\gamma}^*$, $\underset{\sim}{\beta}^*$, we have

$$L(\underset{\sim}{x}^*, \underset{\sim}{\lambda}^*, \underset{\sim}{\gamma}^*, \underset{\sim}{\beta}^*) \le L(\underset{\sim}{x}, \underset{\sim}{\lambda}^*, \underset{\sim}{\gamma}^*, \underset{\sim}{\beta}^*) \tag{5.20}$$

Then

$$\phi_{k+1} = \phi_k + P(\underset{\sim}{x}_k, \phi_k)/p_k \le F(\bar{x}) \tag{5.21}$$

where

$$p_k = \sum_i [f_i(\underset{\sim}{x}_k) - \phi_k], \quad i\in I(\underset{\sim}{x}_k)$$

Proof A necessary condition at the minimum of $P(\underset{\sim}{x}, \phi_k)$ is given by (5.17) which can be rewritten as

$$\sum_{i\in I(\underset{\sim}{x}_k)} \{[f_i(\underset{\sim}{x}_k) - \phi_k]/p_k\} \nabla f_i(\underset{\sim}{x}_k) + \sum_{j\in J(\underset{\sim}{x}_k)} \{\omega_j g_j(\underset{\sim}{x}_k)/p_k\} \nabla g_j(\underset{\sim}{x}_k)$$

$$+ \sum_{\ell\in L} \{v_\ell h_\ell(\underset{\sim}{x}_k)/p_k\} \nabla h_\ell(\underset{\sim}{x}_k) = 0 \tag{5.22}$$

Now, (5.22) is of the form

$$\nabla L(\underset{\sim}{x}_k, \underset{\sim}{\lambda}_k, \underset{\sim}{\gamma}_k, \underset{\sim}{\beta}_k) = 0$$

where

$$\lambda_{k_i} = [f_i(\underset{\sim}{x}_k) - \phi_k]/p_k$$

$$\gamma_{k_j} = \omega_j g_j(\bar{x}_k)/p_k$$

$$\beta_{k_\ell} = v_\ell h_\ell(\bar{x}_k)/p_k$$

Hence, by the assumption in (5.19),

$$L(x_k, \lambda_k, \gamma_k, \beta_k) \leq L(\bar{x}, \lambda_k, \gamma_k, \beta_k) \tag{5.23}$$

Moreover, it can easily be verified that

$$L(\bar{x}, \lambda_k, \gamma_k, \beta_k) \leq L(\bar{x}, \lambda_k, \bar{\gamma}, \bar{\beta}) \tag{5.24}$$

Thus from (5.23) and (5.24)

$$L(x_k, \lambda_k, \gamma_k, \beta_k) \leq L(\bar{x}, \lambda_k, \bar{\gamma}, \bar{\beta})$$

$$= \sum_{i \in I} \lambda_{k_i} [f_i(\bar{x}) - \phi_k] \tag{5.25}$$

Again from Proposition 5.4,

$$\sum_{i \in I} \lambda_{k_i} = 1$$

and hence (5.25) implies

$$L(x_k, \lambda_k, \gamma_k, \beta_k) \leq F(\bar{x} - \phi_k) \tag{5.26}$$

It is now verified by direct substitution that

$$L(x_k, \lambda_k, \gamma_k, \beta_k) = P(x_k, \phi_k)/p_k$$

Therefore,

$$P(x_k, \phi_k)/p_k \leq F(\bar{x}) - \phi_k$$

which concludes the proof. ∎

Remarks. In general the condition (5.20) may not be satisfied everywhere, unless f_i, g_j, and h_ℓ are all convex functions. However, in the neighbourhood of the minimum it should hold in most nondegenerate cases, hence, the faster updating formula would be valid at least near the minimum.

Proposition 5.8. Let $F_0 \triangleq \sup_{\underset{\sim}{x} \in S} F(\underset{\sim}{x})$, where

$$S = \{\underset{\sim}{x}: g_j(\underset{\sim}{x}) \leq 0,\ h_\ell(\underset{\sim}{x}) = 0\}$$

Then for $F_0 \geq \phi \geq F(\underset{\sim}{\bar{x}})$, we have

$$\inf_{\underset{\sim}{x}} P(\underset{\sim}{x}, \phi) = 0 .$$

The proof is very simple and therefore omitted.

The result of Proposition 8 is applied in detecting whether ϕ_k shoots over the optimum value. Validity of the faster updating formula is assumed when $P_{k-1}(\underset{\sim}{x}_{k-1}, \phi_{k-1})$ is small. Based on the foregoing results, a general minimization algorithm valid for either constrained or unconstrained minimax problems is given next.

ALGORITHM 2

Step 1. Set $B_\ell \leq F(\underset{\sim}{\bar{x}})$, where $F(\underset{\sim}{\bar{x}})$ is the optimum, and B_ℓ is a lower bound on $F(\underset{\sim}{\bar{x}})$.

Step 2. Set $\phi_1 \leftarrow B_\ell$, and $k \leftarrow 1$.

Step 3. Minimize $P(\underset{\sim}{x}, \phi_k)$, call the solution $\underset{\sim}{x}_k$.

Step 4. Set $P_o \leftarrow P(\underset{\sim}{x}_k, \phi_k)$

Step 5. Set $B_u \leftarrow \min \{F(\underset{\sim}{x}_o), F(\underset{\sim}{x}_k)\}$, where B_u is an upper bound on $F(\bar{\underset{\sim}{x}})$.

Step 6. Calculate

$$M^M \leftarrow \phi_k + \{P(\underset{\sim}{x}_k, \phi_k)/n\}^{\frac{1}{\alpha}},$$

$$M^T \leftarrow \phi_k + P(\underset{\sim}{x}_k, \phi_k)/\Sigma[f_i(\underset{\sim}{x}_k) - \phi_k], \; i \in I(\underset{\sim}{x}_k)$$

where

$$I(\underset{\sim}{x}_k) = \{i: f_i(\underset{\sim}{x}_k) > \phi_k\}$$

Step 7. If $M^T < B_u$, set $\phi_{k+1} \leftarrow M^T$, otherwise set $\phi_{k+1} \leftarrow M^M$. Also set $\phi_o \leftarrow \phi_{k+1} - \phi_k$

Step 8. Set $B_\ell \leftarrow M^M$, and $S \leftarrow P(\underset{\sim}{x}_k, \phi_k)/P_o$

Step 9. Set $k \leftarrow k+1$

Step 10. Minimize $P(\underset{\sim}{x}, \phi_k)$, call the solution $\underset{\sim}{x}_k$.

Step 11. If $(B_u - B_\ell)$ or $\phi_o \leq \epsilon$, STOP (machine test of a virtual zero)

Step 12. If $P(\bar{\underset{\sim}{x}}_k, \phi_k) > \delta$, go to Step 6 (machine test of a virtual zero). Otherwise, if $S <$ SMAL, STOP (SMAL is a positive constant signifying the closeness of $P(\bar{\underset{\sim}{x}}_{k-1}, \phi_{k-1})$ to zero). If none is true then set $B_u \leftarrow \phi_k$, and $\phi_k \leftarrow B_\ell$, and go to Step 10.

5.4 DISCUSSION

An important result of the investigation is that if the value of the parameter ϕ is less than or equal to the supremum of the function value in the feasible region but greater than or equal to the optimum value of the minimax function, the infimum of $P(\underset{\sim}{x}, \phi)$ equals zero, as stated in Proposition 5.8. This property is the key to the second algorithm. The same fact can be utilized for other purposes, some of which are discussed next.

5.4.1 Obtaining a Lower Bound on the Minimax Optimum

As mentioned earlier, in our algorithms it is necessary to have an optimistic guess for the optimum value $F(\bar{\underset{\sim}{x}})$. In many practical situations, the functions $f_i(\underset{\sim}{x})$ are nonnegative valued, in which case an optimistic guess of $\phi = 0$ would be valid.

However, in some minimax problems, an optimistic guess cannot be made a priori. In such cases a lower bound can be obtained by utilizing the result of Proposition 5.8 as follows:

Step 1. Choose an initial parameter vector $\underset{\sim}{x}_0$ and an initial guess for ϕ_1. Set $k \leftarrow 1$.

Step 2. Minimize $P(\underset{\sim}{x}, \phi_k)$ to find $\bar{\underset{\sim}{x}}_k$.

Step 3. If $P(\bar{\underset{\sim}{x}}_k, \phi_k) > \delta$ (a test for zero), STOP. Otherwise set $\phi_{k+1} \leftarrow \phi_k - \Delta\phi_k$, where $\Delta\phi_k$ is a suitable positive step size.

Step 4. Set $k \leftarrow k+1$, and go to Step 2.

The final value of ϕ_k obtained by the above algorithm would be optimistic.

5.4.2 Application in Filter Design

An important application of Proposition 5.8 is in the design of filters. If we are investigating whether a particular structure will satisfy the design specifications (upper and lower bounds on the response), the fact would be revealed in the first step.

It should also be noted that the gradients need not be computed at each sample point in every step of the algorithms. In general, as ϕ_k approaches the optimum value, fewer and fewer elements of $f_i(x_k)$ will satisfy the condition $f_i(x_k) > \phi_k$, and gradients need be computed only for those elements. This fact may also be verified from the results of the examples tackled in the next section.

The minimax algorithm given in this chapter may also be viewed as a generalization of the nonlinear programming method proposed by Kowalik et al. [26, pp. 180-190, 27], so that a general purpose package program can be written for both kinds of optimizations.

5.5 EXAMPLES

Several examples are presented to illustrate the effectiveness of the approach. The second algorithm as given in section 5.3 is used to solve the problems. The examples taken can be broadly classified into two categories. In the first category are the unconstrained and constrained optimization of the maximum value of a set of functions. In the second category is the minimax design of a three-section

transmission line transformer with and without constraints. Fletcher's optimization algorithm is used to minimize $P(\underset{\sim}{x}_k, \phi_k)$.

Example 5.1. Unconstrained Problem

Minimize the maximum of

$$f_1 = x_1^4 + x_2^2$$

$$f_2 = (2-x_1)^2 + (2-x_2)^2$$

$$f_3 = 2 \exp(-x_1+x_2)$$

The results are given in Table 5.1. This example has also been tackled in [10].

Example 5.2

(5.2-a). Unconstrained Problem

Minimize the maximum of

$$f_1 = x_1^2 + x_2^4$$

$$f_2 = (2-x_1)^2 + (2-x_2)^2$$

$$f_3 = 2 \exp(-x_1+x_2)$$

The results are given in Table 5.2. This problem has also been solved in [10].

(5.2-b). Equality Constraints

Minimize the function of Example (5.2-a) with the following equality constraint:

$$x_1 + x_2 - 2 = 0$$

The results are given in Table 5.3.

5.2-c). Equality and Inequality Constraints

Minimize the function of Example (5.2-a) with the following constraints:

$x_1 + x_2 - 2 = 0$, and

$-x_1^2 - x_2^2 + 2.25 \leq 0$.

The results are given in Tables 5.4 and 5.5 with the starting point at a feasible and an infeasible region respectively.

Example 5.3. Quarter-Wave Transmission Line Transformer

The problem considered here is the same as the three-section transmission line transformer tackled in Examples 1 of Chapter 2. The design parameters are the characteristic impedances z_1, z_2, z_3, and the lengths ℓ_1, ℓ_2, and ℓ_3. The lengths have been normalized with respect to the quarter wave-length ℓ_q at the centre frequency. The sample points are the same as in Chapter 3.

The following problems are tried with two starting points which are the same as the two starting points of Example 1 in Chapter 2, and shown in Table 3.3, henceforth called starting point number 1 and 2 respectively.

(5.3-a): Unconstrained Optimization

The problem as described above is tried with the two starting points. The design parameters are constrained to be positive by taking the actual i-th optimization parameter p_i as $p_i = x_i^{(2)}$, where x_i is the corresponding i-th design parameter. This example is also tackled in

[4, 5, 9]. The final results are given in Table 5.6.

(5.3-b). Equality Constraint

The problem of Example (3-a) is now solved with an added equality constraint on the characteristic impedances, given by

$$z_1 + z_2 + z_3 - 10 = 0 \qquad (5.27)$$

The weighting factor ω_j is taken to be unity, and the example is run with the two starting points as in Example (5.3-a). It is found that for the second starting point the solution converges to a higher value of the $\max|\rho_i|$ than with the first set. However, the necessary conditions for a constrained minimax optimum are found to be satisfied at the solution. The results are shown in Table 5.7. The example is also tried from a starting point where all the parameters are taken to be unity. The results are the same as with starting point number 1.

(5.3-c). Inequality Constraints

The following equality constraints are added on to the characteristic impedances in the original problem of Example (5.3-a):

$$0 \leq z_i \leq 5 \;, \quad i = 1, 2, 3. \qquad (5.28)$$

The example is tried with the same two starting points and the results are given in Table 5.8. The upper and the lower specifications of (5.28) can easily be expressed in the standard form. Hence, by (5.28) we in fact introduce six inequality constraints. The weighting factors ω_i's are all taken to be unity.

The optimum parameter values for all the examples are given in Table 5.9.

TABLE 5.1: EXAMPLE (5.1), UNCONSTRAINED PROBLEM

Step k	ϕ_k	$P(\underline{x}_k, \phi_k)$	$\underline{x}_k$	
			x_1	x_2
1	0	11.2959	1.01702	.820541
2	1.96665	2.8719×10^{-3}	1.00228	.993012
3	1.99993	1.189×10^{-8}	1.00000	.999985
4	2.0	7.606×10^{-19}	1.00000	1.00000

Starting point: $x_1 = x_2 = 2.0$, $F(\underline{x}_0) = 404$, $F(\bar{\underline{x}}) = 2.0$

CPU time = 0.8 sec., Function evaluations = 28

TABLE 5.2: EXAMPLE (5.2-a), UNCONSTRAINED PROBLEM

Step k	ϕ_k	$P(\underline{x}_k, \phi_k)$	$\underline{x}_k$	
			x_1	x_2
1	0	9.50	1.24176	.774005
2	1.81603	.036420	1.13350	.896903
3	1.95218	5.662×10^{-9}	1.13776	.900555
4	1.95223	4.1×10^{-16}	1.13776	.900556

Starting point: $x_1 = x_2 = 2.0$, $F(\underline{x}_0) = 404$, $F(\underline{\bar{x}}) = 1.95223$

CPU time = 0.5 sec., Function evaluations = 21

TABLE 5.3: EXAMPLE (5.2-b), EQUALITY CONSTRAINT

Step k	ϕ_k	$P(x_k, \phi_k)$	x_k	
			x_1	x_2
1	0	9.50113	1.24110	.773632
2	1.81610	3.72023×10^{-2}	1.12753	.899301
3	1.95599	5.375×10^{-4}	1.06843	.950663
4	1.99116	1.638×10^{-5}	1.01186	.991731
5	1.99978	9.333×10^{-9}	1.00028	.999806
6	2.0	9.309×10^{-14}	1.00023	.999768

Starting point: $x_1 = x_2 = 2.0$, $F(\underset{\sim}{x}_0) = 408$, $F(\underset{\sim}{\bar{x}}) = 2.0$

CPU time = 0.5 sec., Function evaluations = 47

TABLE 5.4: EXAMPLE (5.2-c), EQUALITY AND INEQUALITY CONSTRAINTS (FEASIBLE STARTING POINT)

Step k	ϕ_k	$P(\underline{x}_k, \phi_k$	$\underline{x}_k$	
			x_1	x_2
1	0	9.51	1.25063	.775888
2	1.81824	5.108×10^{-2}	1.19113	.862352
3	1.99820	3.522×10^{-3}	1.27550	.780453
4	2.25	1.18×10^{-12}	1.35355	.646449

Starting point: $x_1 = x_2 = 2.0$, $F(\underline{x}_0) = 408$, $F(\bar{\underline{x}}) = 2.25$

CPU time = 1.1 sec., Function evaluations = 35

TABLE 5.5: EXAMPLE (5.2-c), EQUALITY AND INEQUALITY CONSTRAINTS (INFEASIBLE STARTING POINT)

Step k	ϕ_k	$P(x_k, \phi_k)$	x_k	
			x_1	x_2
1	0	9.51	1.25063	.775888
2	1.81824	5.108×10^{-2}	1.19113	.862352
3	1.99820	3.522×10^{-3}	1.2755	.780453
4	2.25	7.79×10^{-13}	1.35355	.646448

Starting point: $x_1 = x_2 = 0.5$, $F(x_0) = 28.41$, $F(\bar{x}) = 2.25$

CPU time = 1.1 sec., Function evaluations = 37

TABLE 5.6: EXAMPLE (5.3-a), THE UNCONSTRAINED PROBLEM

Step k	Starting point 1		Starting point 2		Maximum no. of elements in $I_k(x)$	
	ϕ_k	$P(x_k, \phi_k)$	ϕ_k	$P(x_k, \phi_k)$	1*	2*
1	0	1.9532×10^{-1}	0	1.9532×10^{-1}	11	11
2	.15978	5.0293×10^{-3}	.15978	5.0293×10^{-3}	6	6
3	.19519	1.5439×10^{-5}	.19519	1.5439×10^{-5}	4	4
4	.19729	1.1993×10^{-19}	.19729	1.1992×10^{-19}	2	2
CPU time	5.9 secs.		5.5 secs.		*Nos. 1 and 2 signify starting points	
Function evaluations	72		68			

TABLE 5.7: EXAMPLE (5.3-b), EQUALITY CONSTRAINTS

Step k	Starting point 1		Starting point 2		Maximum no. of elements in $I_k(x)$	
	ϕ_k	$P(x_k, \phi_k)$	ϕ_k	$P(x_k, \phi_k)$	1*	2*
1	0	2.3233×10^{-1}	0	1.1336	11	11
2	.16418	5.9467×10^{-3}	.35482	3.2621×10^{-2}	6	7
3	20222	2.2296×10^{-5}	.44059	2.5564×10^{-16}	4	3
4	.20475	3.9686×10^{-16}	x	x	2	x
CPU time	4.5 secs.		4.3 secs.		*Nos. 1 and 2 signify starting points	
Function evaluations	68		54			

TABLE 5.8: EXAMPLE (5.3-c), INEQUALITY CONSTRAINTS

Step k	Starting point 1		Starting point 2		Maximum no. of elements in $I_k(\underset{\sim}{x})$	
	ϕ_k	$P(\underset{\sim}{x}_k, \phi_k)$	ϕ_k	$P(\underset{\sim}{x}_k, \phi_k)$	1*	2*
1	0	3.7366×10^{-1}	0	3.7366×10^{-1}	11	11
2	.19701	4.0247×19^{-3}	.19701	4.0247×10^{-3}	7	7
3	.22689	4.5547×10^{-5}	.22689	4.5547×10^{-5}	4	4
4	.23056	1.5564×10^{-12}	.23056	1.5564×10^{-12}	2	2
CPU time	5.5 secs.		5.5 secs.		*Nos. 1 and 2 signify starting points	
Function evaluations	96		95			

TABLE 5.9: OPTIMUM PARAMETER VALUES FOR EXAMPLE 5.3

Parameters x_i	Example (5.3-a)	Example (5.3-b)		Example (5.3-c)
		Starting point 1	Starting point 2	
ℓ_1/ℓ_q	1.0000	.9986	.9998	.9994
z_1	1.6347	1.7184	1.5106	1.3829
ℓ_1/ℓ_q	1.0	1.2093	.9999	.9997
z_2	3.1628	3.1730	2.8879	2.6285
ℓ_3/ℓ_q	1.0	3.6746	1.0001	1.0003
z_3	6.1174	5.1086	5.6015	5.0000

5.6 CONCLUSIONS

A fast and efficient approach is proposed to solve the general constrained minimax problem. The constraints are taken care of in a very simple and straightforward manner by converting the constrained problem into a sequence of unconstrained least-squares type optimizations. Based on the approach, two algorithms are proposed, of which the second algorithm should prove faster in general, whereas the first algorithm is simpler to program.

Many problems have been solved using the faster algorithm, and it is found that this algorithm is extremely fast and robust compared to the other existing good algorithms [4, 9, 25, 35]. It is believed that the simplicity and flexibility of the algorithms presented here will lend much freedom to the designer in tackling any minimax problem.

It should also be noted that the method is still valid if the minimax optimization is generalized to a sequence of least-p-th type problems.

CHAPTER 6

DESIGN OF NONLINEAR DC TRANSISTOR CIRCUITS WITHOUT SOLVING NETWORK EQUATIONS

6.1 INTRODUCTION

Recently, several methods have been given for the computer-aided design of several types of nonlinear dc circuits [7,14,17,32]. These basically involve the selection of the adjustable parameters, namely, a subset of the resistances in the circuit, to achieve a desired set of bias voltages and currents. In the case of logic gates the voltage sources may also be included as design parameters.

An automated design algorithm of the type presented in [7,14,17,32] usually employs an iterative circuit optimization technique to systematically adjust the circuit design parameters to find a set of acceptable values for the parameters which satisfies, as closely as possible, the different voltage level and current level specifications [38]. The basic steps in the execution of such an algorithm may be summarized as follows:

1. A performance function is defined which reflects the difference between the desired values and the actual values of the specified voltages and currents.

2. The nonlinear network is analyzed to compute the performance function.

3. The gradients of the performance function with respect to the parameters are computed, which can be found by the method described by Director and Rohrer [13].

4. A function minimization (optimization) algorithm uses the performance function and sensitivity information to generate improved designs.

Thus, an efficient algorithm requires the knowledge of the first partial derivatives of the performance function with respect to the design parameters. At each design iteration, the minimization technique uses the knowledge of the present and past gradients of the performance function to vary the design parameters for getting a lower value for the performance function.

However, one performance function and gradient evaluation requires as many analyses of the original network and its adjoint as there are excitation-response situations [13]. A great deal of the computation time for a design is spent on the nonlinear circuit analyses [12], making any design algorithm efficiency dependent on the efficiency of the analysis algorithm.

In this chapter, a method is proposed for the optimization of nonlinear dc transistor circuits without solving network equations. This is achieved by treating the network equations as equality constraints on the parameters. Thus, a penalty function is defined to transform the constrained problem into a sequential unconstrained problem [27,31], which is then solved by Morrison's algorithm [31], with an improvement proposed by Kowalik et al. [27].

The performance function can be of either the least-squares type or the minimax type. However, for worst-case design, the maximum deviation from the desired values is a better criterion than a weighted sum of the deviations of the actual response from the desired response. Thus, the minimization problem becomes one of minimax optimization with equality constraints, rather than least-squares optimization with

equality constraints. Applying the method proposed in Chapter 4 to such an optimization, a numerical example is taken to show that it is sometimes more desirable to optimize a minimax performance function than a least-squares one.

6.2 PROBLEM FORMULATION

6.2.1 Performance Function:

Given a design specification in terms of the various voltages and currents appearing in the circuit, an error term is associated with each of them. Such a term measures the difference between the actual and desired values. The overall performance index may be written as shown below:

$$f = \sum_{i=1}^{N_V} [W_{V_i} (V_i - \hat{V}_i)]^2 + \sum_{j=1}^{N_I} [W_{I_j} (I_j - \hat{I}_j)]^2 \qquad (6.1)$$

where V_i, I_j are the actual voltages and currents, and $\hat{V}_i$, $\hat{I}_j$ are the corresponding specified values; N_V, N_I are the number of voltage and current specifications, respectively, while W_{V_i}, W_{I_j} are positive scaling parameters. If $V_j \neq 0$, the parameter W_{V_i} is chosen as

$$W_{V_i} = \hat{W}_{V_i} / |\hat{V}_i|$$

where $\hat{W}_{V_i}$ is another scaling parameter. This has the effect of normalizing the corresponding error term in (6.1). On the other hand, if $\hat{V}_i = 0$, such a normalization is not possible. In this case, the scaling factor W_{V_i} is chosen so as to make all of the terms in (6.1) the same order of magnitude. Similar consideration apply to scaling factors W_{I_j}.

Now, let the network equations be defined by $\underset{\sim}{h} = \underset{\sim}{0}$. Then the optimization problem can be stated as

minimize f, subject to the constraint

$\underset{\sim}{h} = \underset{\sim}{0}$

6.2.2 Worst-Case Design:

The parameters of a transistor are functions of the temperature, which also vary from one transistor to another produced in the same batch. Thus the operating point of a transistor circuit may change because of changes in temperature or because one transistor is replaced by another. It may also fluctuate due to fluctuations in the source excitation values. With this background, a set of transistor parameters and source excitation values may be called an environmental condition, and is clear that the operating point of a transistor circuit is dependent on the environmental condition.

The cost function of (6.1) yields a design under a nominal environmental condition. It can easily be extended to include several environmental conditions simultaneously such that the bias point fluctuation is minimized (worst case design). A cost function like (6.1) is then defined for each environmental condition, and the overall cost function F is the algebraic sum of all such functions.

However, in a dc amplifier design, the biasing of each stage is dependent on the preceding stage, unlike ac design. Moreover, the temperature stability is far poorer than in ac design, since the drift of the first stages are amplified by succeeding stages. Thus the operating point may change because of change in temperature or because

one transistor is replaced by another. It may also fluctuate due to a fluctuation in the supply voltage. Thus, for a dc amplifier operating under changing environmental conditions, it would be impossible to determine whether an output voltage change is due to the input signal or due to a drift in the environmental condition. Design of the first stage is therefore very critical in dc amplifiers. Thus in such a case it is more appropriate to define a performance function of the type (6.1) for each environmental condition and minimize the maximum deviation over all such conditions, instead of minimizing the algebraic sum of all such functions.

The performance function F may then be defined as

$$F = \max_{m} f_m$$

where m is the index of environmental conditions, and f_m is the m-th performance function of the type (6.1). Hence, a minimax optimization should be done instead of minimizing the algebraic sum of the performance functions at all the environmental conditions.

6.2.3 Constraint on the Conductances:

As it is desired to have passive conductances, they must be constrained to be positive. This is done by redefining the parameters as $x_i = p_i^2$, where x_i's are the conductances and p_i's are the corresponding design parameters.

6.2.4 Scaling:

It should be noted that in general there are two kinds of parameters involved in the minimization process, namely, p_i's, i.e., the square root

of the conductances; and the voltages. The two kinds of parameters are very much different in orders of magnitude. Hence, for better convergence we redefine the parameters due to the conductances as

$$x_i = \alpha p_i^2 \qquad (6.2)$$

where 'α' is a scale factor depending on the magnitudes of the conductances and node voltages. Thus, p_i's and V_i's are made of the same order of magnitude.

6.2.5 Minimizing Power Dissipation

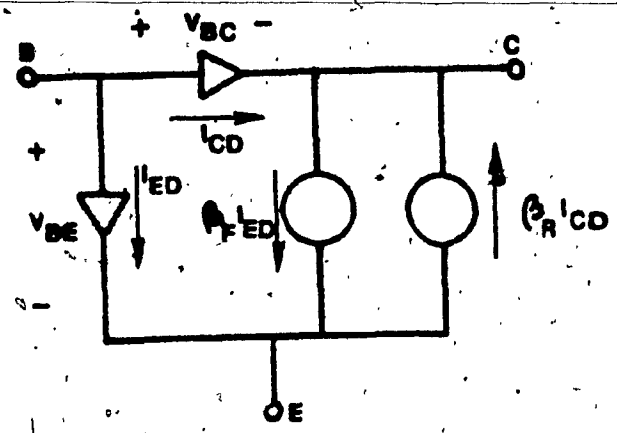
To minimize power dissipation, terms of the type $W_p\ I_p\ V_p$ are added to the performance index F in (6.1) for each power supply, where V_p is the power supply voltage, I_p is the power supply current, and W_p is a weighting factor used to have a trade off between the accuracy of specified voltage and current, and the power consumption.

6.2.6 DC Transistor Model

For the formulation of the nodal equations, the Ebers-Moll transistor model [12] replaces the transistors. Such a model for an npn transistor is shown in Figure 6.1.

6.2.7 Optimization Algorithm

Morrison's algorithm as described in Chapter 4 (section 4.3.2) is used to solve the problem. In the present case, $\underset{\sim}{h}(\underset{\sim}{x})$ is taken to be the set of network equations while F is the performance function. The penalty term is now given by $\omega\Sigma h_i^2 = \omega \underset{\sim}{h}^T \underset{\sim}{h}$, where ω is the scaling factor to give appropriate weightage to the penalty term. Thus the penalty augmented function is now given by



$$I_{ED} = \frac{I_{ES}}{\beta_F+1}\left[\exp\left(q\frac{V_{BE}}{kT}\right) - 1\right]$$

$$I_{CD} = \frac{I_{CS}}{\beta_R+1}\left[\exp\left(q\frac{V_{BC}}{kT}\right) - 1\right]$$

Fig. 6.1 Ebers-Moll Model for npn Transistor.

$$P = (F - M)^2 + \omega \sum_i h_i^2 \qquad (6.3)$$

It is clear that the performance function F given by (6.1) is always greater than or equal to zero. Hence, any negative number may be chosen as the initial guess for M.

Flow-Chart of the Algorithm

A flow-chart based on Morrison's algorithm applicable to our design problem is given in Figure 6.2.

6.3 COMPUTATION OF GRADIENTS

Most efficient minimizing algorithms require the knowledge of the gradient vector of the function to be minimized with respect to the design parameters. Thus for a fast design algorithm, it is essential that the gradients be found easily and with a minimum of computations. Let $\underset{\sim}{x}$ be a vector of the design parameters. Differentiating the penalty augmented function P in (6.3) with respect to $\underset{\sim}{x}$, we have,

$$\frac{\partial P}{\partial \underset{\sim}{x}} = \nabla P = 2(f - M) \frac{\partial F}{\partial \underset{\sim}{x}} + 2\omega \left(\frac{\partial \underset{\sim}{h}}{\partial \underset{\sim}{x}}\right)^T \underset{\sim}{h} \qquad (6.4)$$

where, from equation (6.1),

$$\frac{\partial F}{\partial \underset{\sim}{x}} = 2 \sum_{i=1}^{N_V} W_{V_i} (V_i - \hat{V}_i) \frac{\partial V_i}{\partial \underset{\sim}{x}} + 2 \sum_{j=1}^{N_I} W_{I_j} (I_j - \hat{I}_j) \frac{\partial I_j}{\partial \underset{\sim}{x}} \qquad (6.5)$$

In addition to the design parameters due to the designable conductances and node voltages, the vector $\underset{\sim}{x}$ may also contain a subset of the independent voltage sources and current sources as in example 3 to be

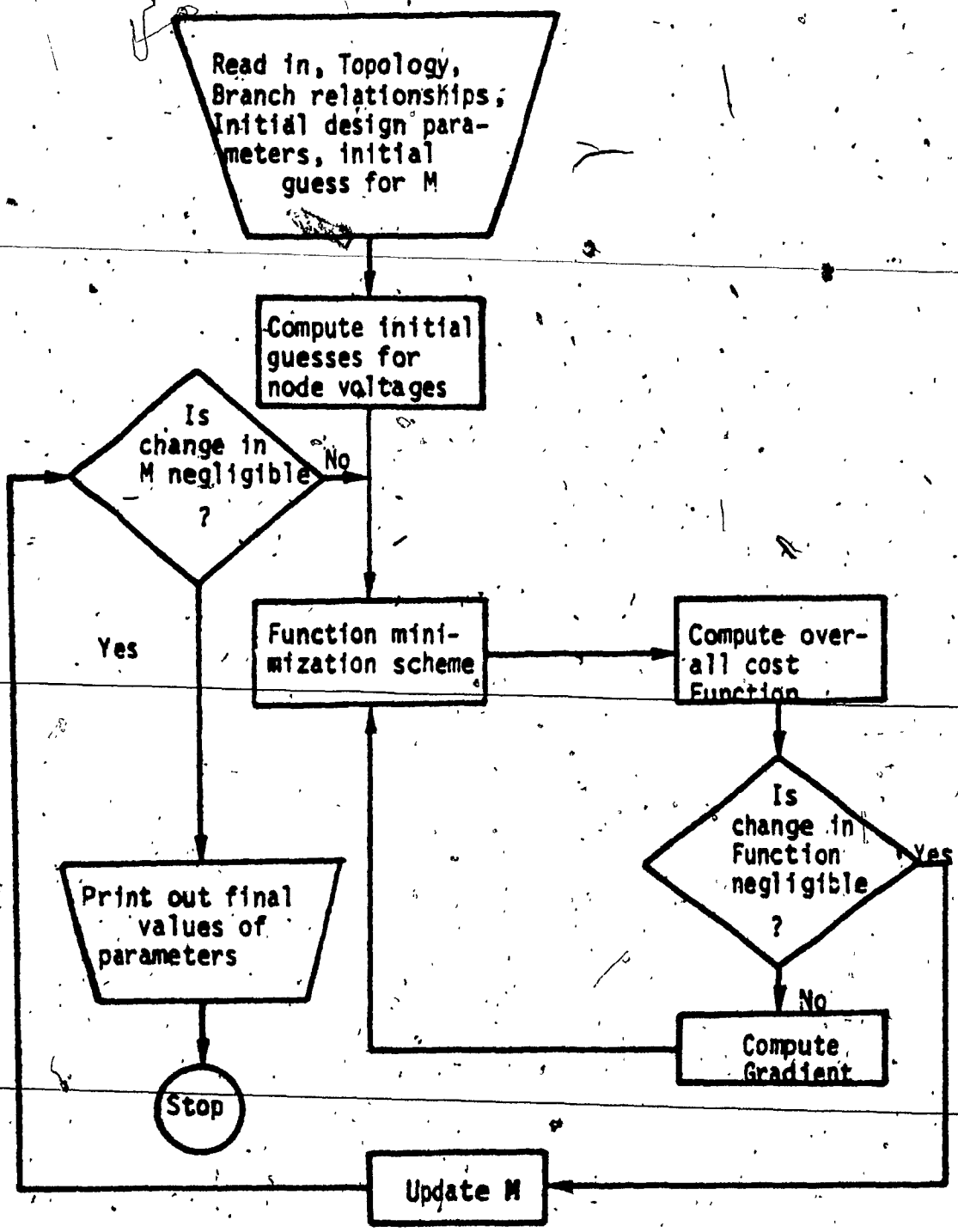


Fig. 6.2 Flowchart of Design Algorithm.

discussed later. To derive the expression for $\frac{\partial F}{\partial \underset{\sim}{x}}$ and $\frac{\partial h}{\partial \underset{\sim}{x}}$ in (6.4) we proceed as follows:

For a network where the controlling current flows through a passive element, the current i_e may be expressed as $i_e = Y_e v_e$ where Y_e is the admittance of the element and v_e the voltage. Thus all the controlled sources finally become voltage controlled. Now, without any loss of generality all the controlled sources can be assumed to be voltage controlled current sources for nodal formulation of the network equations.

A generalized i^{th} branch in a network is as shown in Figure 6.3, where the subscript i indicates the i^{th} branch. A network is an interconnection of such branches, so the different branch parameters for the whole network may be expressed in vector form as follows:

$\underset{\sim}{p}_e$ → branch element "parameter" vector,

$\underset{\sim}{V}_e$ → element voltage vector,

$\underset{\sim}{I}_e$ → element current vector,

$\underset{\sim}{E}_s$ → voltage source vector,

$\underset{\sim}{I}_s$ → current source vector,

$\underset{\sim}{I}_B$ → branch current vector,

$\underset{\sim}{V}_B$ → branch voltage vector.

Now, in a network with controlled sources, $\underset{\sim}{I}_e$, $\underset{\sim}{E}_s$, and $\underset{\sim}{I}_s$ may be expressed as functions of $\underset{\sim}{V}_e$ and $\underset{\sim}{p}_e$. Representing them in vector form, we get

$$\underset{\sim}{I}_e = f_1(\underset{\sim}{V}_e, \underset{\sim}{p}_e) \qquad (6.6a)$$

$$\underset{\sim}{E}_s = f_2(\underset{\sim}{V}_e, \underset{\sim}{p}_e) \qquad (6.6b)$$

$$\underset{\sim}{I}_s = f_3(\underset{\sim}{V}_e, \underset{\sim}{p}_e) \qquad (6.6c)$$

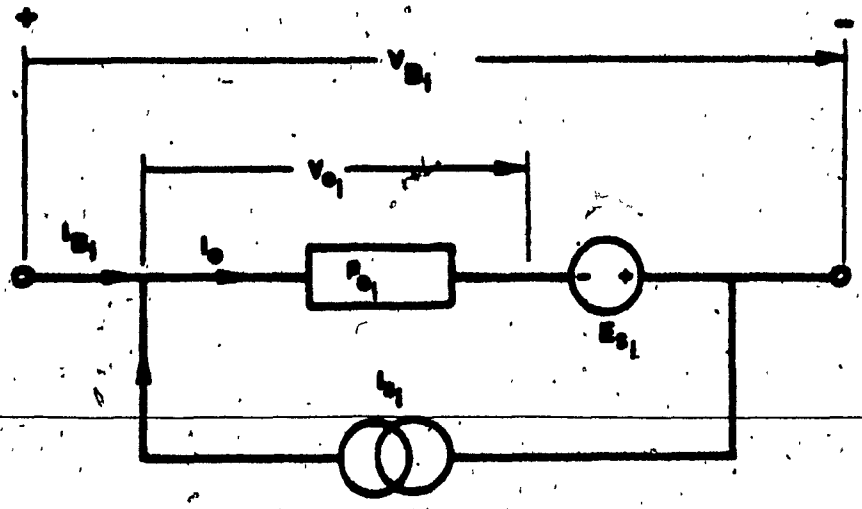


Fig. 6.3 Generalized i-th Branch.

From Figure 6.3,

$$\underset{\sim}{I}_B = \underset{\sim}{I}_e - \underset{\sim}{I}_s$$
$$= f_4(\underset{\sim}{V}_e, \underset{\sim}{P}_e)$$

and Kirchoff's current law yields,

$$\underset{\sim}{A}\underset{\sim}{I}_B \triangleq \underset{\sim}{h} = \underset{\sim}{0}$$

The gradients of the penalty augmented function P with respect to the parameters have a contribution from the penalty function term, and also from the performance index term as evident from (6.4). The two parts are dealt with separately as shown below:

6.3.1 Gradients of the Penalty Function Term:

(a) Gradients with respect to the node voltages

The penalty term is given by

$$\sum_j h_j^2 = \underset{\sim}{h}^T \underset{\sim}{h} \triangleq Q$$

Denoting the node voltage vector by $\underset{\sim}{V}_n$, Kirchoff's voltage law yields,

$$\underset{\sim}{V}_B = \underset{\sim}{A}^T \underset{\sim}{V}_n$$

however,

$$\underset{\sim}{V}_e = \underset{\sim}{V}_B + \underset{\sim}{E}_s$$
$$= \underset{\sim}{A}^T\underset{\sim}{V}_n + \underset{\sim}{E}_s$$

therefore,

$$\underset{\sim}{h} = \underset{\sim}{A}\, f_4(\underset{\sim}{A}^T \underset{\sim}{V}_n + \underset{\sim}{E}_s, \underset{\sim}{P}_e)$$

Now,

$$\frac{\partial Q}{\partial \underset{\sim}{V}_n} = \frac{\partial (\underset{\sim}{h}^T \underset{\sim}{h})}{\partial \underset{\sim}{V}_n}$$

$$= 2 \left(\frac{\partial \underset{\sim}{h}}{\partial \underset{\sim}{V}_n}\right)^T \underset{\sim}{h},$$

$$= 2\underset{\sim}{A} \left[\left. \frac{\partial \underset{\sim}{f}_4(\underset{\sim}{V}_e, \underset{\sim}{p}_e)}{\partial \underset{\sim}{V}_e} \right|_{\underset{\sim}{V}_e = \underset{\sim}{A}^T \underset{\sim}{V}_n + \underset{\sim}{E}} \right]^T \left(\underset{\sim}{A}^T + \frac{\partial \underset{\sim}{E}_s}{\partial \underset{\sim}{V}_n}\right)\underset{\sim}{h} \qquad (6.7)$$

In the case where each branch current source and each branch voltage source is controlled by a single parameter or by a single voltage source, (6.7) can be considerably simplified. Specifically, suppose the sources in the i-th branch depend only on V_{e_j} and p_{e_j}. Then (6.6) becomes

$$I_{e_i} = f_1(V_{e_j}, p_{e_j}) \qquad (6.8a)$$

$$E_{s_i} = f_2(V_{e_j}, p_{e_j}) \qquad (6.8b)$$

$$I_{s_i} = f_3(V_{e_j}, p_{e_j}) \qquad (6.8c)$$

Note that the above situation prevails in dc transistor circuits. Further, in transistor model in Figure 6.1, we have $\frac{\partial E_s}{\partial V_n} = 0$. With these simplifications, (6.7) becomes

$$\frac{\partial Q}{\partial \underset{\sim}{V}_n} = 2\,\underset{\sim}{A}\,\underset{\sim}{J}^T\,\underset{\sim}{A}^T\,\underset{\sim}{h},$$

where the Jacobian $\underset{\sim}{J}$ is given by

$$\underset{\sim}{J} = \left. \frac{\partial \underset{\sim}{f}_4(\underset{\sim}{V}_e, \underset{\sim}{p}_e)}{\partial \underset{\sim}{V}_e} \right|_{\underset{\sim}{V}_e = \underset{\sim}{A}^T \underset{\sim}{V}_n + \underset{\sim}{E}_s}$$

Thus the Jacobian is the same as the branch admittance matrix of a transistor ac equivalent circuit, where the diodes are replaced by equivalent conductances given by,

$$G_j = \frac{\partial\, I_s[e^{\frac{q}{kT} V_{e_j}} - 1]}{\partial V_{e_j}}$$

$$= \frac{I_s\, q}{kT}\, e^{\frac{q}{kT} V_{e_j}}$$

Denoting the branch conductance matrix by $\underset{\sim}{Y}_B$, we therefore get

$$\frac{\partial Q}{\partial \underset{\sim}{V}_n} = 2\, \underset{\sim}{A}\, \underset{\sim}{Y}_B^T\, \underset{\sim}{A}^T\, \underset{\sim}{h}$$

$$= 2\, \underset{\sim}{Y}^T\, \underset{\sim}{h}$$

where $\underset{\sim}{Y}$ is the nodal admittance matrix of the new network. It should be noted that if the nodal admittance matrix of the linearized adjoint network [7] is denoted as $\underset{\sim}{Y}_a$, then $\underset{\sim}{Y}_a = \underset{\sim}{Y}^T$, so that

$$\frac{\partial Q}{\partial \underset{\sim}{V}_n} = 2\, \underset{\sim}{Y}_a\, \underset{\sim}{h} \qquad (6.9)$$

(b) <u>Gradients with respect to the design parameters:</u>

Differentiating h w.r.t. the vector $\underset{\sim}{p}_e$, we have

$$\frac{\partial \underset{\sim}{h}}{\partial \underset{\sim}{p}_e} = \underset{\sim}{A}\, \frac{\partial \underset{\sim}{f}_4(\underset{\sim}{V}_e, \underset{\sim}{p}_e)}{\partial \underset{\sim}{p}_e}\Bigg|_{\underset{\sim}{V}_e = \underset{\sim}{A}^t \underset{\sim}{V}_n + \underset{\sim}{E}_s}$$

However, when the design parameters are lumped conductances, f_4 becomes $f_4(\underset{\sim}{V}_e, \underset{\sim}{Y}_e)$, and the above expression may be simplified as follows:

Let the incidence matrix $\underset{\sim}{A}$ be partitioned as $\underset{\sim}{A} = [\underset{\sim}{A}_1 | \underset{\sim}{A}_2]$, where the columns of $\underset{\sim}{A}_1$ correspond to the conductances to be optimized, and

those of $\underset{\sim}{A}_2$ to the remaining branches. Accordingly, the linearized branch admittance matrix $\underset{\sim}{Y}_b$ is partitioned as-

$$\underset{\sim}{Y}_b = \begin{bmatrix} \underset{\sim}{Y}_{b_{11}} & \underset{\sim}{Y}_{b_{12}} \\ \underset{\sim}{Y}_{b_{21}} & \underset{\sim}{Y}_{b_{22}} \end{bmatrix}$$

where $\underset{\sim}{Y}_{b_{11}}$ is diagonal. Therefore,

$$\frac{\partial Q}{\partial \underset{\sim}{p}_e} = 2 \left(\frac{\partial \underset{\sim}{h}}{\partial \underset{\sim}{Y}_e}\right)^T \underset{\sim}{h}$$

$$= 2 \left(\underset{\sim}{A}_1 \; \underset{\sim}{A}_2\right) \left.\frac{\partial f_4(\underset{\sim}{V}_e, \underset{\sim}{Y}_e)}{\partial \underset{\sim}{Y}_e}\right|_{\underset{\sim}{V}_e = \underset{\sim}{A}^T \underset{\sim}{V}_n + \underset{\sim}{E}_s} \underset{\sim}{h}$$

$$= 2 \frac{\partial}{\partial \underset{\sim}{Y}_e} \{\underset{\sim}{A}_1 \underset{\sim}{Y}_{b_{11}} (\underset{\sim}{A}_1^T \underset{\sim}{V}_n + \underset{\sim}{E}_s)\}^T \underset{\sim}{h}$$

Thus the i^{th} component of $\frac{\partial Q}{\partial \underset{\sim}{p}_e}$ is given by

$$\frac{\partial Q}{\partial p_{e_i}} = 2 \underset{\sim}{h}^T \underset{\sim}{A}_1 \underset{\sim}{M}_1 (\underset{\sim}{A}^T \underset{\sim}{V}_n + \underset{\sim}{E}_s) , \qquad (6.10)$$

where

$$\underset{\sim}{M}_1 = i \begin{bmatrix} 0 & & & & i \\ & 0 & & & \\ & & \ddots & & \\ & & & i & \\ & & & & \ddots \\ & & & & & 0 \end{bmatrix}$$

(c) <u>Gradients with respect to the independent voltage sources:</u>

$$\frac{\partial Q}{\partial \underset{\sim}{E}_s} = 2 \underset{\sim}{A} \left.\frac{\partial f_4(\underset{\sim}{V}_e, \underset{\sim}{Y}_e)}{\partial \underset{\sim}{E}_s}\right|_{\underset{\sim}{V}_e = \underset{\sim}{A}^T \underset{\sim}{V}_n + \underset{\sim}{E}_s} \underset{\sim}{h}$$

$$= 2 \underset{\sim}{A} \underset{\sim}{J}_1 \underset{\sim}{h} \qquad (6.11)$$

where the Jacobian J_1 is the branch admittance matrix of the branches involving the independent voltage sources as parameters, with the other branch admittances set equal to zero.

6.3.2 Gradients of the Performance Function:

Computation of $\frac{\partial f}{\partial \underset{\sim}{x}}$ involves the computation of $\frac{\partial V_j}{\partial \underset{\sim}{x}}$ and $\frac{\partial I_j}{\partial \underset{\sim}{x}}$, as is evident from (6.5). The computation of $\frac{\partial V_j}{\partial \underset{\sim}{x}}$ is straightforward as V_j can be readily expressed in terms of the node voltages $\underset{\sim}{V}_n$ and the source voltages $\underset{\sim}{E}_s$. But in order to compute $\frac{\partial I_j}{\partial \underset{\sim}{x}}$, I_j must first be expressed in terms of the design parameters, node voltages, and other network parameters (such as conductances whose values are not subject to design). I_j may be the j^{th} branch current I_{B_j}, or just the j^{th} element current I_{e_j} or the source current I_{s_j}. As shown in Figure 6.3, I_{B_j} may be expressed as

$$\begin{aligned} I_{B_j} &= I_{e_j} - I_{s_j} \\ &= Y_{e_j} V_{e_j} - I_{s_j} \\ &= Y_{e_j} (V_{B_j} + E_{s_j}) - I_{s_j} \end{aligned}$$

where Y_{e_j} is the conductance of the element p_{ej}. Now, if the j^{th} branch is connected between the $\ell^{th}_{(j)}$ and $k^{th}_{(j)}$ nodes, V_{B_j} may be expressed as the difference between $\ell^{th}_{(j)}$ and $k^{th}_{(j)}$ node voltages, i.e.,

$$V_{B_j} = V_{n_\ell} - V_{n_k}$$

In this case

$$I_{B_j} = Y_{e_j} [(V_{n_\ell} - V_{n_k}) + E_{s_j}] - I_{s_j}$$

Thus

$$\frac{\partial I_{B_j}}{\partial \underset{\sim}{x}} = \frac{\partial Y_{e_j}}{\partial \underset{\sim}{x}} [(V_{n_\ell} - V_{n_k}) + E_{s_j}] + Y_{e_j} [\frac{\partial V_{n_e}}{\partial \underset{\sim}{x}} - \frac{\partial V_{n_k}}{\partial \underset{\sim}{x}} + \frac{\partial E_{s_j}}{\partial \underset{\sim}{x}}] - \frac{\partial I_{s_j}}{\partial \underset{\sim}{x}} \tag{6.12}$$

(Note that in a dc transistor circuit $\frac{\partial E_{s_j}}{\partial \underset{\sim}{x}} = \underset{\sim}{0}$, since it contains no controlled voltage sources). All the gradients involved in (6.12) can be readily calculated except $\frac{\partial I_{s_j}}{\partial \underset{\sim}{x}}$. As mentioned previously, we assume that all current sources are voltage controlled. Therefore, I_{s_j} can be expressed as a function of the node voltages and the design parameters. Once this is done, the computation of $\frac{\partial I_j}{\partial \underset{\sim}{x}}$ is trivial.

In conclusion, all the gradients can be computed in a very straightforward manner with only the knowledge of the network topology and branch relationships.

6.4 EXAMPLES

Three examples are presented to illustrate the effectiveness of the approach. The first two examples are on the design of biasing circuits taken from [17]. Example 3 is taken from [12], and involves the design of an emitter-coupled logic gate.

Initial Choice of the Node Voltage Values:

In all these examples, it is found that the convergence to the desired optimum occurs readily if the initial node voltage values are chosen in the neighbourhood of the feasible region. This is done as follows. The transistors in the network are first replaced by simplified Ebers-Moll models, i.e., the diode between base to collector and the

corresponding current source in Figure 6.1 is ignored. Whenever a voltage across a branch or between two nodes is given as a design specification, this branch is replaced by a voltage source equal to the specified value. For each collector current specification, the base to emitter diode is replaced by a voltage source equivalent to the voltage drop in the diode. Each of the other diodes in the circuit are replaced by a voltage source v_{be} of magnitude in the feasible range (say $0.2V \leq v_{be} \leq 0.6V$) taking care that Kirchoff's voltage law is not violated. The resultant linear network, which may be termed the "modified network", is then analysed and the node voltages thus obtained are taken as the initial node voltage values for the optimization algorithm. (Note that the number of independent nodes reduce drastically in the modified network).

The minimization algorithm used is due to Fletcher and Powell [20] as in [17], and the linear search to obtain the step size is the method of Golden Sections. The examples are run on a CDC 6400 computer available. The scale factor k for the network equations is chosen to be 10^4 for all examples. Similarly, the scale factor 'α' for the parameters p_i due to the conductances is selected as 10^2. The termination criterion in all the examples is $M_{i+1} - M_i \leq 10^{-15}$.

Example 6.1. Gain Stages for Operational Amplifier

The circuit diagram for the example is shown in Figure 6.4. The adjustable parameters are the five conductances G_1 to G_5. The bias point specifications are as follows: $I_{C_1} = 0.5$ mA, $I_{C_2} = 0.5$ mA, $I_{C_3} = 1.0$ mA, $I_{C_4} = 1.0$ mA, $E_1 = 1.5V$, and $E_2 = 1.5V$. The Ebers-Moll parameters of all the transistors are taken to be $I_{ES} = 10^{-14}$A,

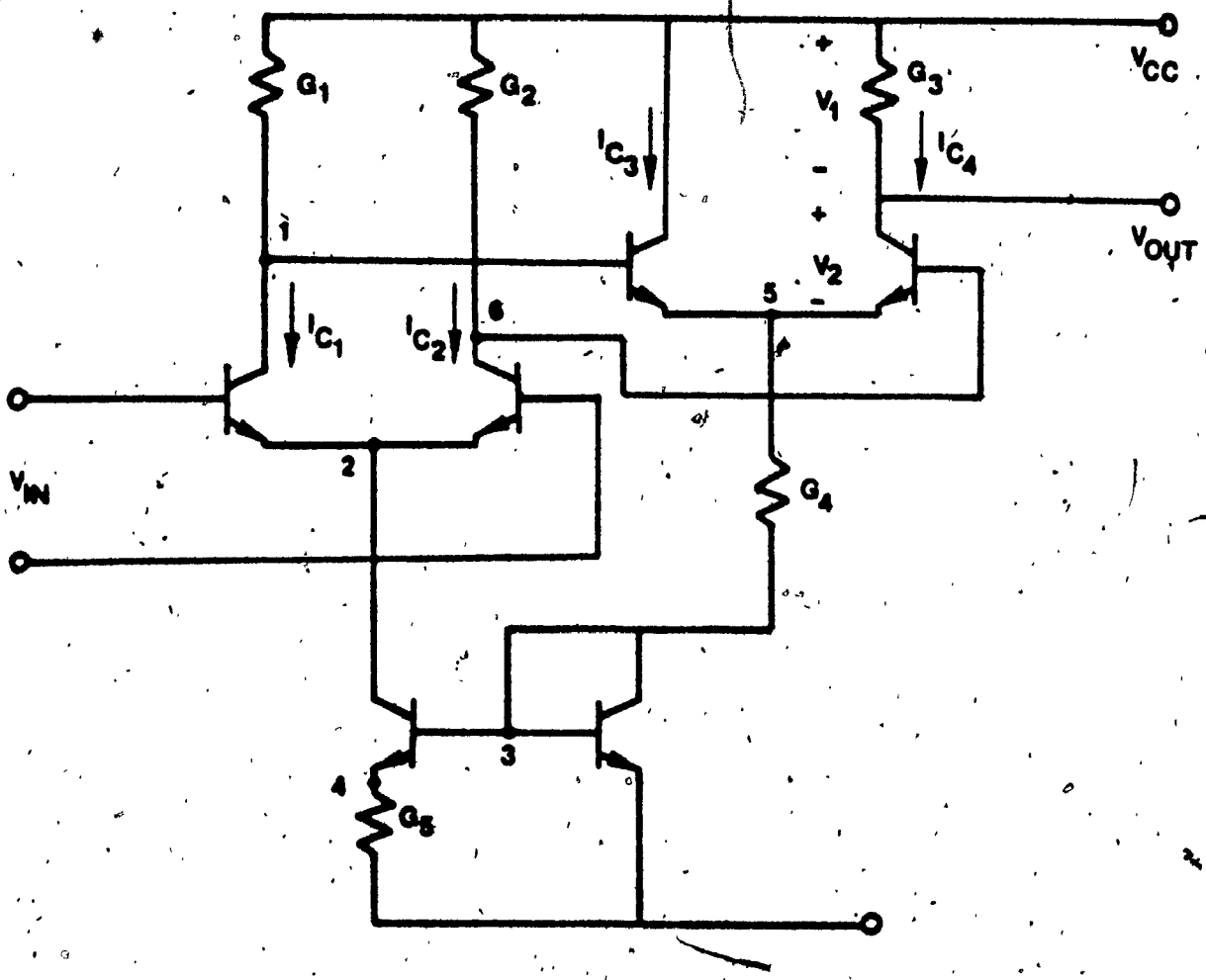


Fig. 6.4 A Two-Stage Operational Amplifier.

$I_{CS} = 10^{-14}$ A, $\beta_F = 100$, $\beta_R = 5$, and $T = 300°K$. The supply voltages are $V_{CC} = 18V$, and $V_{EE} = -18V$. Now, the performance index after normalizing is given by

$$F = (-I_{C_1} - \beta_R I_{C_1} + \beta_F I_{E_1} - 0.5\times10^{-3})^2 \times 4\times10^6$$
$$+ (-I_{C_2} - \beta_R I_{C_2} + \beta_F I_{E_2} - 0.5\times10^{-3})^2 \times 4\times10^6$$
$$+ (-I_{C_5} - \beta_R I_{C_5} + \beta_F I_{E_5} - 1\times10^{-3})^2 \times 10^6$$
$$+ (-V_7 G_3 + V_C G_3 - 1\times10^{-3})^2 \times 10^6$$
$$+ (V_C - V_7 - 1.5)^2 / 2.25$$
$$+ (V_7 - V_5 - 1.5)^2 / 2.25$$

As the number of specifications is equal to the number of designable conductances, F does not include a term representing the power dissipation.

As shown in the figure, the network has seven nodes, so that there are seven unknown node voltages. Thus there are seven equality constraints to be satisfied along with the design specifications. Now, the problem is converted to an unconstrained one according to the modified Morrison's algorithm,

$$P = (F - M)^2 + \omega \sum_{j=1}^{7} h_j^2$$

The function P now involves seven node voltages as variables in addition to the five variables due to the five conductances. The initial values of the conductances are chosen to be the same as in [17]. In Run 1, the initial guesses for the seven node voltage values are obtained by

analyzing the modified network as mentioned earlier; in Runs 2 to 4 the initial node voltage values are varied slightly from the above values to show that the convergence is not much affected even if the modified network is not analyzed very accurately. The initial values as well as the final values are shown in Table 6.1, where the conductances are in mhos and the voltages are in volts. Table 6.2 shows the accuracy to which the network equations and the voltage and current specifications are satisfied in Run 1, when the parameters assume the final values of Table 6.1. They are satisfied to the same order of accuracy in Runs 2 to 4. The initial and the final values of P and the modified Morrison's parameter M are as follows:

	Run 1	Run 2	Run 3	Run 4
Initial Value:	P = 22.31 M = - 1.0	P = 26.30 M = - 1.0	P = 38.30 M = - 1.0	P = 38.47 M = - 1.0
Final Value:	$P=0.21\times10^{-17}$ M = 0	$P=0.87\times10^{-16}$ M = 0	$P=0.73\times10^{-16}$ M = 0	$P=0.12\times10^{-15}$ M = 0.0

The CPU time required on CDC 6400 computer as reported in [17], with the same initial values for the conductances, is 30 seconds.

TABLE 6.1: FINAL RESULTS OF EXAMPLE 6.1

Parameters	Run 1	Run 2	Run 3	Run 4	Final Values
G_1	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.218×10^{-3}
G_2	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.218×10^{-3}
G_3	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.666×10^{-3}
G_4	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.1×10^{-3}	0.625×10^{-3}
G_5	0.1	0.1	0.1	0.1	0.578×10^{-1}
V_1	15.5	15.5	15.5	15.5	15.659
V_2	-0.5	-0.5	-0.5	-0.5	-0.640
V_3	-17.5	-17.5	-17.5	-17.5	-17.323
V_4	-18.0	-17.6	-17.6	-17.8	-17.982
V_5	15.0	15.0	15.0	15.0	15.0
V_6	15.5	15.5	15.5	15.5	15.659
V_7	16.5	16.5	16.5	16.5	16.5
CPU Time	5.0 secs.	4.2 secs.	4.7 secs.	3.5 secs.	
No.of Fletcher-Powell iterations	42	39	44	33	

TABLE 6.2: FINAL RESULTS OF EXAMPLE 6.1 (CONTINUED)

Network Equations	Specified Currents and Voltages
$h_1 = -0.584 \times 10^{-19}$	I_{C_1} -0.5 mA = -0.579×10^{-10}
$h_2 = -0.266 \times 10^{-18}$	I_{C_2} -0.5 mA = -0.579×10^{-10}
$h_3 = -0.253 \times 10^{-18}$	I_{C_3} -1 mA = -0.756×10^{-9}
$h_4 = -0.711 \times 10^{-18}$	I_{C_4} -1 mA = -0.263×10^{-10}
$h_5 = 0.1[illegible] \times 10^{-18}$	V_1 - 1.5 V = -0.528×10^{-8}
$h_6 = -0.416 \times 10^{-18}$	V_2 - 1.5 V = -0.162×10^{-8}
$h_7 = -0.278 \times 10^{-19}$	

Example 6.2. Single Transistor Stage: Two Environmental Conditions

The circuit to be designed is shown in Figure 6.5. It is desired to have minimum deviation in the bias point at two environmental conditions, as well as minimum power dissipation. The desired operating point is as follows: I_C = 1 mA, and V_{CE} = 5 V. The final values of the designable conductances G_1 to G_3 should be such that the operating point I_C and V_{CE} are not too sensitive to changes in the transistor parameters; while minimizing the total dc power consumption at the same time. The transistor parameters at the two environmental conditions are as follows:

Environmental Condition 1	Environmental Condition 2
$\beta_F = 25$	$\beta_F = 100$
$\beta_R = 5$	$\beta_R = 5$
$I_{CS} = 10^{-14}$ A	$I_{CS} = 10^{-14}$A
$I_{ES} = 10^{-14}$ A	$I_{ES} = 10^{-14}$A

The performance index after normalizing the bias voltage and bias current parts and scaling the power consumption part is given by

$$F = \sum_{i=1}^{2} (V_2^{(i)} - V_3^{(i)} - 5)^2 \times \frac{1}{25} + (-V_2^{(i)} G_3 + 10^{-3})^2 \times 10^6$$

$$+ Wp\,(V_{CC}\,G_1 + V_{CC}\,G_3 - V_1^{(i)}\,G_1 - V_2^{(i)}\,G_3)$$

where the superscript i indicates the environmental condition.

As shown in the circuit, the network has three nodes and hence the network equation vector $\underset{\sim}{h}$ has three elements corresponding to those nodes. The penalty augmented function is then given by

$$P = (F - M)^2 + \omega \sum_{i=1}^{2} \sum_{j=1}^{3} (h_j^{(i)})^2$$

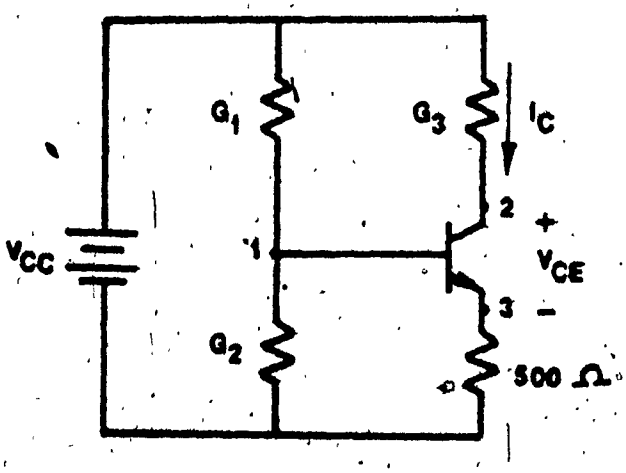


Fig. 6.5 A One Transistor Biasing Circuit.

where the subscript j corresponds to the different nodes and the superscript i indicates the environmental conditions. Hence, the function P contains nine variables; three due to the conductances G_1 to G_3, three node voltages at one environmental condition and three at the other. The initial node voltage values are obtained by analyzing the modified network at one environmental condition, and the results are given in Table 6.3 under Run 1. For Runs 2 and 3 the initial choice of the conductances are varied, and also the initial node voltage values are changed slightly from the analyzed values. Table 6.4 shows for Run 1 the accuracy to which the network equations and the specified voltages and currents are satisfied. They are satisfied to the same order of accuracy in Runs 2 and 3 as well. The superscripts denote the environmental conditions. The initial and the final values of P and the modified Morrison's parameter N are as follows:

Initial value:	P = 0.197 M = 0.0	P = 0.185 M = 0.0	P = 0.265 M = 0.0
Final Value:	$P=0.214\times10^{-14}$ $M=0.199\times10^{-4}$	$P=0.669\times10^{-14}$ $M=0.198\times10^{-4}$	$P=0.207\times10^{-14}$ $M=0.198\times10^{-4}$

In all runs $W_p = 0.01$. The CPU time reported for this example in [17] is 5 seconds.

TABLE 6.3: FINAL RESULTS OF EXAMPLE 6.2

Parameters	Initial Values			Final Values
	Run 1	Run 2	Run 3	
G_1	$.1 \times 10^{-3}$	$.1 \times 10^{-4}$	$.1 \times 10^{-4}$	$.949 \times 10^{-5}$
G_2	$.1 \times 10^{-3}$	$.1 \times 10^{-4}$	$.1 \times 10^{-3}$	$.706 \times 10^{-4}$
G_3	$.1 \times 10^{-3}$	$.1 \times 10^{-3}$	$.1 \times 10^{-3}$	$.184 \times 10^{-3}$
$V_1^{(1)}$	1.1	1.5	1.0	1.08
$V_2^{(1)}$	5.6	6.0	5.5	5.43
$V_3^{(1)}$	0.6	1.0	1.0	0.42
$V_1^{(2)}$	1.1	1.0	1.0	0.76
$V_2^{(2)}$	5.6	6.0	5.5	5.43
$V_3^{(2)}$	0.6	1.0	1.0	0.44
CPU Time	2.4 secs.	2.8 secs.	2.6 secs.	
No. of Fletcher-Powell iterations	44	50	47	

TABLE 6.4: FINAL RESULTS OF EXAMPLE 6.2 (CONTINUED)

Network Equations	Specified Currents and Voltages
$h_1^{(1)} = -0.996 \times 10^{-19}$	$V_{CE}^{(1)}$ -5.0 V = 0.92×10^{-2}
$h_2^{(1)} = 0.214 \times 10^{-17}$	$V_{CE}^{(2)}$ -5.0 V = -0.25×10^{-2}
$h_3^{(1)} = 0.252 \times 10^{-17}$	$I_C^{(1)}$ -1.0 mA = -0.62×10^{-6}
$h_1^{(2)} = 0.826 \times 10^{-18}$	$I_C^{(2)}$ -1.0 mA = 0.61×10^{-6}
$h_2^{(2)} = -0.213 \times 10^{-17}$	Power dissipation$^{(1)}$ = 0.926 mW
$h_3^{(2)} = -0.235 \times 10^{-17}$	Power dissipation$^{(2)}$ = 0.928 mW

Example 6.3. Design of an Emitter-Coupled Logic Gate

The circuit diagram is shown in Figure 6.6. The purpose of this example is to illustrate the feasibility and efficiency of the proposed method for this kind of problems. It is desired to have the property of preserving the dc operating levels of the input signal $V_s(t)$ shown in Figure 6.7. Thus when the input is +0.5V, the output should be +0.5V, and when the input is -0.5V, the output should be -0.5V. The performance index after normalizing is given by

$$F = (V_{OUT}^{(1)} - 0.5)^2 \times 4 + (V_{OUT}^{(2)} + 0.5)^2 \times 4$$

where the superscript 1 corresponds to an input of +0.5V and the superscript 2 is for an input of -0.5V. The penalty augmented function is then given by

$$P = (F - M)^2 + \omega \sum_{i=1}^{2} \sum_{j=1}^{4} (h_j^{(i)})^2$$

where the subscript j denotes the node numbers, while the superscript i corresponds to the two inputs. The variables in F are the three conductances G_1 to G_3, the two voltage sources e_1 and e_2, and four node voltages for each state for a total of eight voltage variables. Hence there are thirteen variables in all. A guess is made at the initial values of the design parameters, and the initial choice for the node voltage values is obtained by analyzing the modified network at each state. The initial and final values are summarized as follows:

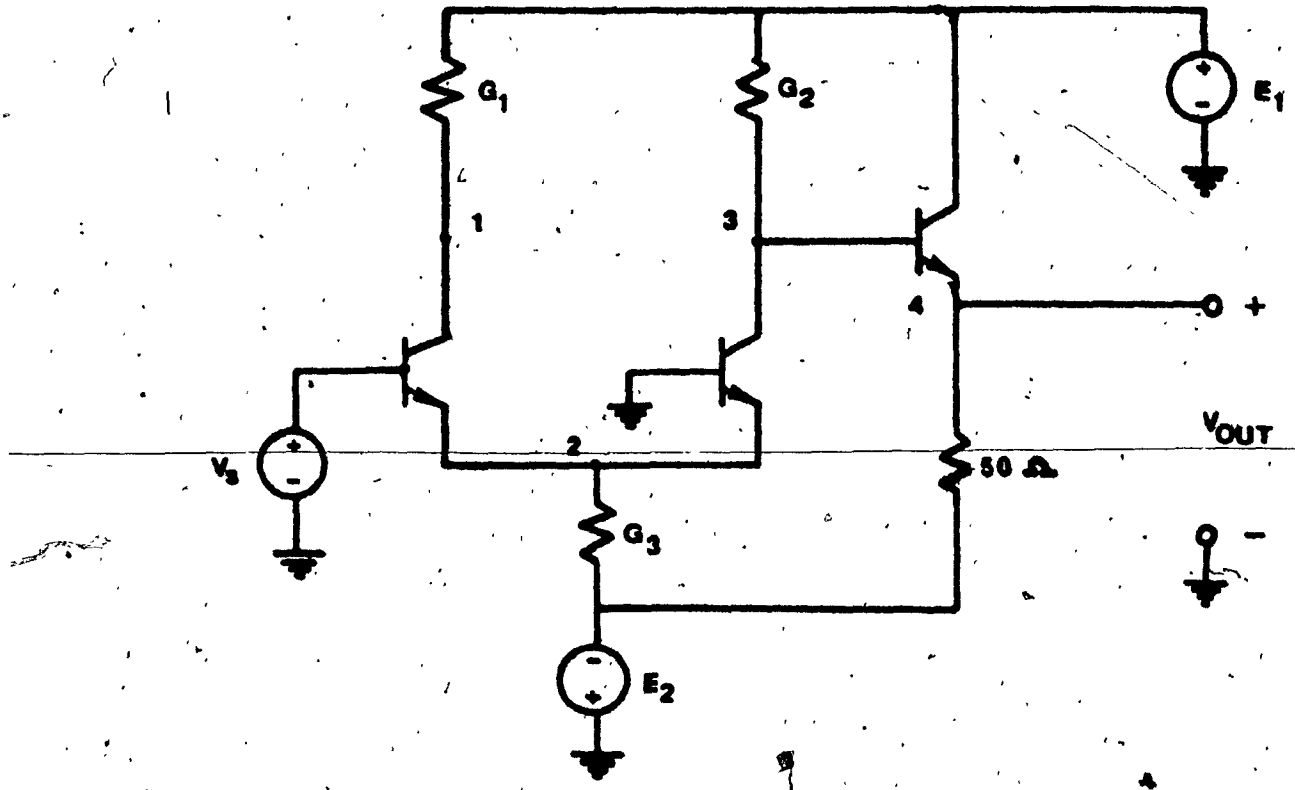


Fig. 6.6 An Emitter-Coupled Logic Gate.

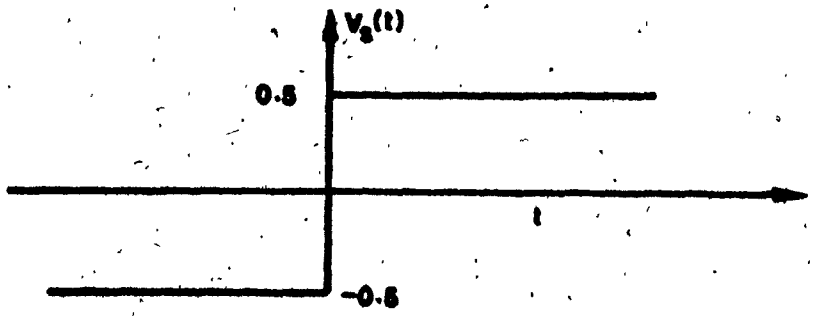


Fig. 6.7 Input Levels for the ECL Gate.

Initial Values		Final Values	
P	= 989.34	P	= 0.3479×10^{-16}
M	= -1.0	M	= 0
G_1	= $.100 \times 10^{-1}$ mho.	G_1	= 0.218×10^{-1} mho.
G_2	= $.100 \times 10^{-1}$ mho.	G_2	= 0.101×10^{-1} mho.
G_3	= $.100 \times 10^{-1}$ mho.	G_3	= 0.153×10^{-1} mho.
e_1	= 1.15xV	e_1	= 1.29 V
e_2	= 2.0 V	e_2	= 1.41 V
$V_1^{(1)}$	= 0.8 V	$V_1^{(1)}$	= 0.47 V
$V_2^{(1)}$	= -0.2 V	$V_2^{(1)}$	= -0.23 V
$V_3^{(1)}$	= 0.8 V	$V_3^{(1)}$	= 1.25 V
$V_4^{(1)}$	= 0.5 V	$V_4^{(1)}$	= 0.5 V
$V_1^{(2)}$	= 1.0 V	$V_1^{(2)}$	= 1.29 V
$V_2^{(2)}$	= -0.8 V	$V_2^{(2)}$	= -0.72 V
$V_3^{(2)}$	= 0.1 V	$V_3^{(2)}$	= 0.23 V
$V_4^{(2)}$	= -0.5 V	$V_4^{(2)}$	= -0.5 V

The network equations are satisfied as follows:

$h_1^{(1)} = 0.757 \times 10^{-11}$

$h_2^{(1)} = 0.844 \times 10^{-11}$

$h_3^{(1)} = -0.557 \times 10^{-10}$

$h_4^{(1)} = -0.109 \times 10^{-11}$

$h_1^{(2)} = -0.638\ 10^{-11}$

$h_2^{(2)} = -0.102\ 10^{-10}$

$h_3^{(2)} = -0.203\ 10^{-10}$

$h_4^{(2)} = 0.223\ 10^{-12}$

The output voltages are exactly matched, i.e., $V_{OUT}^{(1)} = 0.5$ V, and $V_{OUT}^{(2)} = 0.5$ V

No. of Fletcher-Powell iterations = 27

CPU time = 3.5 secs.

Example 6.4. Single Transistor Stage: Two Environmental Conditions (Worst-Case Design)

The example is taken to show that minimax optimization may sometimes yield a better result than the least-squares optimization, in the sense that the maximum deviation of the operating point from the nominal values is less. The same circuit as in example 2 is taken. However, the environmental conditions are as follows:

Environmental Condition 1	Environmental Condition 2
$T = 150°K$	$T = 450°K$
$\beta_F = 25$	$\beta_F = 100$
$\beta_R = 5$	$\beta_R = 5$
$I_{CS} = 10^{-14}$ A	$I_{CS} = 10^{-14}$ A
$I_{ES} = 10^{-14}$ A	$I_{ES} = 10^{-14}$ A

The minimax function to be minimized is given by

$$F = \max f_i, \quad i = 1, 2$$

where

$$f_i = [e_1^{(i)}]^2 \frac{1}{25} + [e_2^{(i)}]^2 \times 10^6 + W_p e_3^{(i)} \qquad (6.13)$$

and

$$e_1^{(i)} = (V_2^{(i)} - V_3^{(i)} - 5) + (-V_2^{(i)} G_3 + 10^{-3})^2 \times 10^6$$

$$e_2^{(i)} = (-V_2^{(i)} G_3 + 10^{-3})$$

$$e_3^{(i)} = (V_{CC} G_1 + V_{CC} G_3 - V_1^{(i)} G_1 - V_2^{(i)} G_3)$$

where the superscript i indicates the environmental condition. The penalty augmented function corresponding to (4.18a) is given by

$$P(\underset{\sim}{x},M) = [\max_{i}(f_i - M)]^2 + \omega \sum_{\ell=1}^{2} \sum_{j=1}^{3} [h_j^{(\ell)}]^2 \,, \quad i = 1,2 \qquad (6.14)$$

where i and ℓ indicate the environmental conditions. Note that the set of equality constraints for each i includes the network equations for all the environmental conditions. The constrained minimax algorithm as proposed in Chapter 4 is used for the optimization.

The example is also run for a least-squares performance function and the results are given in Table 6.5 for comparison. The value of W_p is again chosen to be 0.01. It is found that the maximum deviation in the operating point from the nominal value is more in least-squares case than in the minimax case. At the optimum the network equations in both cases are satisfied to the order of at least 10^{-15}. In Table 6.6, the final values of the specifications ($e_j^{(i)}$'s) realized under the two environmental conditions are given for both methods.

TABLE 6.5: FINAL RESULTS OF EXAMPLE 6.4

Parameters	Initial Values of Parameters	Final Values of Parameters	
		Minimax	Least Squares
G_1	$.1\times10^{-3}$	$.604\times10^{-5}$	$.734\times10^{-5}$
G_2	$.1\times10^{-3}$	$.323\times10^{-4}$	$.350\times10^{-4}$
G_3	$.1\times10^{-3}$	$.175\times10^{-3}$	$.202\times10^{-3}$
$v_1^{(1)}$	1.2	1.37	1.49
$v_2^{(1)}$	6.0	5.58	5.03
$v_3^{(1)}$	1.0	0.392	0.506
$v_1^{(2)}$	1.2	0.742	0.828
$v_2^{(2)}$	6.0	5.47	5.25
$v_3^{(2)}$	1.0	0.415	.498
CPU time in secs		9	6
No. of Fletcher-Powell iterations		108	77

TABLE 6.6: FINAL RESULTS OF EXAMPLE 6.4 (CONTINUED)

Performance Function	$e_1^{(1)}$	$e_2^{(1)}$	$e_3^{(1)}$	$e_1^{(2)}$	$e_2^{(2)}$	$e_3^{(2)}$
Minimax	0.19	-0.17×10^{-4}	$.83\times10^{-3}$	$.51\times10^{-1}$	-0.38×10^{-4}	$.85\times10^{-3}$
Least Squares	-0.48×10^{-1}	$.13\times10^{-4}$	$.11\times10^{-2}$	-0.25	$.57\times10^{-4}$	$.10\times10^{-2}$

6.5 DISCUSSION

A chronic problem with any optimization problem is that the speed of convergence depends to some extent on the initial choice of the parameters. In the existing methods such as in [17], the designer is required to provide an initial guess for the network design parameters. However, in the present approach, an initial guess for the node voltage values must also be given. It has been found that the convergence is slow for an entirely arbitrary choice of the initial node voltage values; in fact, convergence to the desired optimum may not even occur in some cases. However, it is observed that if the initial guess for the node voltage values are in the neighbourhood of the ones obtained by analyzing the modified network as mentioned in Section 6.5, convergence is fast. For the design method to be applicable to a real, large-scale problem, the initial guess for the node voltage values should be automated.

Justification for the final design is based on the fact that the design specifications in the form of a performance index are realized as the vector $\underset{\sim}{h}(\hat{x})$ of the KCL equations becomes zero. However, in reality it is not zero but a vector of very small numbers. In the examples presented here, since the various currents in the circuit are of the order of milliamperes, the fact that the components of the final $\underset{\sim}{h}(\underset{\sim}{x})$ are of the order of 10^{-10} or less is quite satisfactory. To make (6.5) of the same order as the first term $(F - M)^2$, a scale factor of $\omega = 10^4$ is used. However, in general one may normalize the KCL equations with respect to the Euclidean norm of the current vector to obtain a more meaningful measure.

As mentioned in Section 6.2, there are two distinct kinds of parameters involved in the optimization procedure, namely, the square root of the conductances and the node voltages, which are very much different in orders of magnitude in general. Knowing that most of the conductances are in the range of millimhos, and knowing the order of the magnitudes of the node voltages, the scale factor 'α' in (6.2) is chosen as 10^2. However, it was found that the convergence is not very much sensitive to the selection of 'α' and hence its choice is not very critical.

One of the main advantages of the algorithm used is that the updating of the "Morrison's parameter" M for successive iterations is automated. However, the initial choice of M must be optimistic, i.e., $M \leq F(\underset{\sim}{x})$. This presents no problem in the present context as the performance index $F(\underset{\sim}{x})$ is defined such that $F(\underset{\sim}{x}) \geq 0$, so that any negative number is suitable as the initial guess for M.

The efficiency of the present algorithm lies in the fact that unlike the usual practice as applied in [17], it is not necessary to analyze the nonlinear network at each iteration of an optimization algorithm. Even though the parameter space becomes considerably larger, which may require many more iterations within the optimization algorithm, the number of operations required at each iteration is considerably smaller. Thus it is expected that the algorithm will prove faster with more nonlinear elements in the circuit.

6.6 CONCLUSIONS

An algorithm is described for the optimization of nonlinear dc circuits where it is not necessary to analyze the nonlinear network. The network equations are treated as equality constraints on the design parameters, and the optimization is carried out using a penalty function technique. A method is given for the computation of the parameter gradients. The network equations are formulated on the nodal basis, and only the knowledge of the network topology and the branch relationships is enough to compute the overall performance function and its gradient. It is also indicated how the initial choice of the node voltages can be automated.

Secondly, the constraints on the conductance values to be nonnegative has been dealt with in a simple manner which offers no difficulty in the convergence at all. The speed of convergence is improved substantially by scaling the conductance parameters such that they are of the same order of magnitude as the node voltages.

The problem of worst-case design is tackled as a minimax optimization problem with equality constraints. It is shown by an example that in such a situation minimax design may prove to be more desirable than least-squares design.

In designing the transistor circuits it was not found necessary to temporarily simplify the transistor model as mentioned in [17], however, it is expected that such a strategy might speed up the convergence in certain cases.

CHAPTER 7

CONCLUSIONS

7.1 SUMMARY

The thesis proposes a number of algorithms for minimax optimization. The problem of minimax optimization has been studied from a mathematical point of view and it is shown that the negative of the gradient for a least p-th performance function with a large value of p does not always yield a descent direction for the corresponding minimax performance function. This fact is quite revealing since many workers have tried to solve the minimax problem approximately by taking a least p-th performance function with a large value of p, and they have found that in some cases the solution does not converge to the minimax optimum.

A method for solving unconstrained minimax problems is given where the direction of steepest descent is found by solving a linear programming problem. The method is then applied to solve different kinds of filter-design problems and to the modelling of a Schottky clamped transistor, and is found to be efficient and reliable.

A new approach is given for nonlinear programming problems with inequality and equality constraints, where the original problem is converted to a minimax optimization problem with equality constraints. The method of Morrison [3]], and Kowalik et al. [27] is adapted to solve such a problem.

In Chapter 5, a method is given for solving a constrained minimax problem by converting it into minimization of a sequence of continuously

differentiable functions. It is thus possible to apply any of the efficient gradient methods for minimizing an unconstrained function. The method is found to be extremely fast and reliable compared to the existing methods for unconstrained problems too. For constrained minimax problems there is no other satisfactory method existing in the literature.

A method is given for the optimal design of a dc transistor circuit without solving network equations. The network equations are treated as equality constraints on the design parameters. Morrison's method [31] is then applied to convert the constrained problem into a sequence of unconstrained problems. A method is given to derive the gradient of the performance function with respect to all the optimization parameters. It is also shown by an example that for the worst-case design a minimax performance function usually yields a better result than a least-squares one.

7.2 SUGGESTIONS FOR FURTHER INVESTIGATION

The work of this thesis has revealed additional problems for further investigation.

(i) A proof of convergence has not been given for the minimax algorithm presented in Chapter 2. It has been shown in [40] that an algorithm following the direction of steepest descent as the direction of search converges to a local minimum. It should be also possible to show that the algorithm converges when a modified version of Fletcher-Powell method is used to find the direction of search by treating the

direction of steepest descent as if it were the negative of the gradient.

(ii) It is found that in modelling the Schottky-clamped transistor in Chapter 2, analyses of the nonlinear circuits take large amounts of time. Thus the network equations should be treated as equality constraints on the modelling parameters and the constrained problem should be converted to an unconstrained one by using the technique of Chapter 5.

(iii) As pointed out in Chapter 5, a constrained minimax problem is transformed into a sequence of least-squares type functions which are continuously differentiable but the Hessian does not exist everywhere. However, instead of converting the original problem into least-squares type functions, one may convert it into a sequence of least p-th type problems and the method is still valid. It can be easily shown [19] that the converted function is continuously differentiable for $p>1$, and the Hessian exists for $p>2$. It is believed that a unconstrained Quasi-Newton minimization technique should be more efficient if the Hessian exists. Hence, the problems should be tried for $p=4$. Moreover, in large size problems, Newton's method using the Hessian may be more efficient, if the sparsity of the Hessian is exploited in inverting it.

(iv) The method of reduced gradient [23, pp. 274-290] is a very efficient method for nonlinear programming problems, particularly those with linear equality constraints. In the constrained minimax method proposed in Chapter 5, instead of the linear equality constraints being added as penalties, the problem should be treated without the linear equality constraints. The resultant gradient should then be reduced

with respect to the equality constraints, thereby reducing the number of independent variables. This strategy should be investigated and compared by running some examples.

(v) The minimax algorithms presented in this thesis should be extended to the time domain also.

REFERENCES

1. Aoki, M., "Introduction to Optimization Techniques," New York: The MacMillian Company, 1971.

2. Bandler, J.W., "Conditions for Minimax Optimum," IEEE Trans. Circuit Theory, Vol. CT-18, pp. 476-479, 1971.

3. Bandler, J.W., and P.A. MacDonald, "Optimization of Microwave Networks by Razor Search," IEEE Trans. Microwave Theory Tech.," Vol. MTT-17, pp. 552-562, Aug. 1969.

4. Bandler, J.W., T.V. Srinivasan, and C. Charalambous, "Minimax Optimization of Networks by Grazor Search," IEEE Trans. Microwave Theory Tech., Vol. MTT-20, pp. 596-604, Sept. 1972.

5. Bandler, J.W., and C. Charalambous," Practical Least p-th Optimization of Networks," IEEE Trans. Microwave Theory Tech., Vol. MTT-21, pp. 815-818, Dec. 1972.

6. Bandler, J.W., and C. Charalambous, "Nonlinear Programming Using Minimax Techniques," Journal of Optimization Theory and Applications, Vol. 13, No.6, 1974.

7. Broderson, A.J., S.W. Director, and W.A. Bristol, " Simultaneous Automated AC and DC Design of Linear Integrated Circuit Amplifiers," IEEE Trans. Circuit Theory, Vol. CT-18, No.1, pp. 50-58, Jan. 1971.

8. Bandler, J.W., T.V. Srinivasan, "Constrained Minimax Optimization by Grazor Search," Proc. 6-th Hawaii Int. Conf. System Sciences (Honolulu, Hawaii), 1973, pp. 125-128.

9. Charalambous, C., and J.D. Bandler, "New Algorithms for Network Optimization," IEEE Trans. Microwave Theory Tech., Vol. MTT-21, pp. 815-818, Dec. 1973.

10. Charalambous, C., "Nonlinear Least p-th Approximation with Applications," Ph.D. Dissertation, McMaster University, Hamilton, Ontario, Canada. Feb. 1973.

11. Charalambous, C., "A Unified Review of Optimization," IEEE Trans. Microwave Theory Tech., Vol. MTT-22, pp. 289-299, March 1974.

12. Calahan, D.A., "Computer-Aided Network Design," McGraw Hill, New York, 1972.

13. Director, S.W., and R.A. Rohrer, "A Generalized Adjoint Network and Network Sensitivities," IEEE Trans., Vol. CT-16, pp. 330-336, Aug. 1969.

14. Director, S.W., "Computer-Aided Design of Linear Integrated Circuit Amplifiers," Proc. of the IEEE Int. Symp. on Circuit Theory, 1973.

15. Dunford, N., and J.T. Shwarz, "Linear Operators," Part I, "Derivatives of Norms," New York: Interscience Publishers, 1958.

16. Desoer, C.A., and M. Vidyasagar, "Feedback Systems and Input Output Properties," Academic Press, New York, 1975.

17. Dowell, R.I., and R.A. Rohrer, "Automated Design of Biasing Circuits," IEEE Trans. Circuit Theory, Vol. CT-18, pp. 85-89, Jan. 1971.

18. Dutta S.R.K., and M. Vidyasagar, "On the Interrelationships between Nonlinear Programming and Minimax Programming," sent for publication.

19. Dutta, S.R.K., and M. Vidyasagar, "Constrained Minimax Problem as a Sequence of Least-squares Optimizations," Eighteenth Midwest Symposium on Circuits and Systems, August, 1975, R.16.3 .

20. Fletcher, R., and M.J.D. Powell, "A Rapidly Convergent Descent Method for Minimization," Comput. J., Vol.6, pp. 163-168, June, 1963.

21. Fletcher, R., "A New Approach to Variable Metric Algorithms," Comput. J., Vol.13, pp. 317-322, Aug. 1970.

22. Hatley, Jr., W.T., "Computer Synthesis of Wide-band Impedance Matching Networks," Tech. Rept. 6657-2, Stanford Electronics Laboratories, Stanford University, Stanford, California.

23. Himmelblau, D.M., "Applied Nonlinear Programming," New York: McGraw Hill, 1972.

24. Heald, R.A., and D.A. Hodges, "Design of Schottky-Barrier Diode Clamped Transistor Layouts," IEEE J. Solid-State Circuits, Vol. SC-8, pp. 269-275, Aug. 1973.

25. Ishizaki, Y., and H. Watanabe, "An Iterative Chebyshev Approximation Method for Network Design," IEEE Trans. Circuit Theory, Vol. CT-15, pp. 326-336.

26. Jacoby, S.L.S., J.S. Kowalik, and J.T. Pizzo, "Iterative Methods for Nonlinear Optimization Problems," Prentice Hall, Inc., Englewood Cliffs, New Jersey, 1972.

27. Kowalik, J., M.R. Osborne, and D.M. Ryan, "A New Method for Constrained Optimization Problems," Operations Research, pp. 973-983, Vol. 17, 1969.

28. Kowalik, J., and M.R. Osborne, "Methods for Unconstrained Optimization Problems," New York, American Elsevier Publishing Co., 1968.

29. Matthaei, G.L., L. Young, and E.M.T. Jones, "Microwave Filters, Impedance Matching Networks and Coupling Structures," New York: McGraw Hill, 1964, Ch.6.

30. Mokari-Bolhassan, M.E., and T.N. Trick, "Computer-Aided Design of Distributed-Lumped-Active Networks," IEEE Trans. Circuit Theory, Vol. CT-18, pp. 187-190, Jan. 1971.

31. Morrison, D.D., "Optimization by Least-Squares," SIAM J. Numer. Anal., pp. 83-88, Vol.5, No.1, March 1968.

32. Murray-Lasso, M.A., and W.D. Baker, "Computer Design of Multistage Bias Circuits," Proc. Allerton Conference on Circuits and Systems Theory, pp. 557-563, Oct. 1967.

33. Mangasarian, O.L., "Nonlinear Programming," McGraw-Hill Book Company, New York, 1969.

34. Noyce, R.N., R.E. Bohn, and H.T. Chua, "Schottky Diodes Make IC Scene," Electronics, pp. 74-80, July 21, 1969.

35. Osborne, M.R., and G.A. Watson, "An Algorithm for Minimax Approximation in the Non-linear Case," Comput. J., Vol.12, pp. 63-68.

36. Orchard, H.J., and G.C. Temes, "Filter Design Using Transformed Variables," IEEE Trans. Circuit Theory, Vol. CT-15, pp. 385-408, Dec. 1968.

37. Srinivasan, T.V., "Minimax System Modelling and Design," Ph.D. dissertation, McMaster University, Hamilton, Ont., Canada; July 1973.

38. Temes, G.C., and D.A. Calahan, "Computer-Aided Network Optimization The-State-of-the-Art," Proc. IEEE, Vol.55, No.11, Nov. 1967, pp. 1832-1863.

39. Vidyasagar, M., "On the Derivatives of Norms on Finite Dimensional Spaces," to appear in IEEE Trans. Circuits and Systems.

40. Vidyasagar, M., "Convergence of Steepest-Descent Techniques for Directionally Differentiable Functions," sent for publication.

41. Warren, A.D., L.S. Lasdon and D.F. Suchman, "Optimization in Engineering Design," Proc. IEEE, Vol.55, No.11, Nov. 1967, pp. 1885-1897.