EFFICIENT OPTIMIZATION WITH INTEGRATED GRADIENT APPROXIMATIONS, PART I: ALGORITHMS

J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen

SOS-86-12-R

December 1986

EFFICIENT OPTIMIZATION WITH INTEGRATED GRADIENT APPROXIMATIONS

PART I: ALGORITHMS

J.W. Bandler, Fellow, IEEE, S.H. Chen, Student Member, IEEE,

S. Daijavad, Member, IEEE, and K. Madsen

Abstract A flexible and effective algorithm is proposed for efficient optimization with integrated gradient approximations. It combines the techniques of perturbations, the Broyden update and the special iterations of Powell. Perturbations are used to provide an initial approximation as well as regular corrections. The approximate gradient is updated using Broyden's formula. The special iterations of Powell are utilized to generate strictly linearly independent directions. A modification to Broyden's formula is introduced to exploit possible sparsity of the Jacobian. Utilizing this algorithm, powerful gradient-based nonlinear optimization tools for circuit CAD can be employed without the effort of calculating exact derivatives. Computational efficiency is greatly improved as compared to estimating derivatives entirely by numerical differentiation. Part I describes the theoretical aspects of the algorithm and some test problems. Implementation and examples of practical applications are presented in Part II.

This work was supported in part by the Natural Sciences and Engineering Research Council of Canada under Grants A7239 and G1135.

J.W. Bandler and S.H. Chen are with the Simulation Optimization Systems Research Laboratory and the Department of Electrical and Computer Engineering, McMaster University, Hamilton, Canada L8S 4L7.

J.W. Bandler is also with Optimization Systems Associates Inc., 163 Watson's Lane, Dundas, Ontario, Canada L9H 6L1.

S. Daijavad is with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, California 94720, U.S.A.

K.Madsen is with the Institute for Numerical Analysis, Technical University of Denmark, Building 302, DK-2800 Lyngby, Denmark.

I. INTRODUCTION

Many powerful algorithms for nonlinear optimization have been developed and applied to circuit CAD problems. For example, algorithms for linearly constrained ℓ_1 and minimax optimization have been described by Bandler, Kellermann and Madsen [1],[2]. One difficulty in extending their application to a wide range of practical problems, however, is that exact gradients of all functions with respect to all variables must be made available. In many situations, either an explicit expression of the exact gradient is not available or the actual implementation of such an expression is very tedious and time-consuming. With only the function values available, as is the case for typical circuit CAD packages, one usually resorts to the method of perturbations (numerical differentiation) for the gradient. However, except for rather simple problems, the computational labor for estimating gradients entirely by perturbation is very expensive.

In this paper, we propose a flexible and effective approach to gradient approximation for nonlinear optimization. It is a hybrid method which utilizes perturbations, the Broyden update [3] and the special iterations of Powell [4]. Perturbations are used to provide an initial approximation and some subsequent corrections. The Broyden rank-one formula has been used in conjunction with the special iterations of Powell to update the approximate gradients. See, for example, Madsen [5] and Zuberek [6]. Such an update does not require extra function evaluations but its accuracy may not be satisfactory for some highly nonlinear problems or for a certain stage of the optimizer. In our algorithm, perturbations are introduced in a flexible manner to provide regular corrections. A suitable compromise between accuracy and computational labor can be achieved for a particular application. We also propose a modification of the Broyden update which incorporates a knowledge, if available, of the structure of the problem (e.g., one that has a sparse Jacobian).

Our presentation is organized into two parts. In Part I, the theoretical aspects of the algorithm are described and its performance is demonstrated by some test problems. In Part II, which is our companion paper [7], the practical impact of the new algorithm is illustrated through three important implementations with applications to robust FET modelling, worst-case tolerance design of a microwave amplifier and efficient multiplexer optimization. The new algorithm is shown to be efficient and effective in handling a large variety of problems.

II. ESTIMATING THE GRADIENT BY PERTURBATIONS

The first-order derivative of $f_j(\mathbf{x})$ with respect to x_i can be approximated by

$$\frac{\partial f_j(\mathbf{x})}{\partial \mathbf{x}_i} \simeq \frac{f_j(\mathbf{x} + h\mathbf{e}_i) - f_j(\mathbf{x})}{h} , \tag{1}$$

where $\mathbf{e}_i$ is a vector with zero entries except that the ith component is 1. The accuracy of approximations of this type can be improved by decreasing the magnitude of h and/or by averaging the results of a two-sided approximation, i.e., using both positive and negative perturbations. This method is very reliable but the labor involved grows in proportion to the dimension of the problem. In actual implementation, we use perturbations to obtain an initial approximation at the starting point unless the user can provide such an initial approximation. During the iteration of optimization, with prescribed regularity, perturbations may also be used to correct the approximate derivatives if so desired.

III. THE BROYDEN UPDATE

It has been shown [2] that for quadratic functions the computational effort is smaller when the Jacobians are generated by the Broyden rank-one updating formula rather than approximated at each step by perturbations. Although such an advantage can not be proved for a general problem, the Broyden formula still provides an efficient alternative for approximating derivatives. Having an approximate Jacobian $\mathbf{G}_k$ at a point $\mathbf{x}_k$ and the function values at $\mathbf{x}_k$ and $(\mathbf{x}_k + \mathbf{h}_k)$, we obtain

$$\mathbf{G}_{k+1} = \mathbf{G}_k + \frac{\mathbf{f}(\mathbf{x}_k + \mathbf{h}_k) - \mathbf{f}(\mathbf{x}_k) - \mathbf{G}_k\mathbf{h}_k}{\mathbf{h}_k^T\mathbf{h}_k} \mathbf{h}_k^T . \tag{2}$$

The new approximation $\mathbf{G}_{k+1}$ satisfies the following equation

$$\mathbf{f}(\mathbf{x}_k + \mathbf{h}_k) - \mathbf{f}(\mathbf{x}_k) = \mathbf{G}_{k+1}\mathbf{h}_k . \tag{3}$$

In other words, G_{k+1} provides a perfect linear model between two points $\mathbf{x}_k$ and $(\mathbf{x}_k+\mathbf{h}_k)$. Notice that if $\mathbf{x}_k$ and $(\mathbf{x}_k+\mathbf{h}_k)$ are iterates of the optimization the Broyden formula does not require additional function evaluations. Some difficulties in the application of the Broyden formula, however, have been reported (see, for example, [4],[5] and [6]).

(1) If some functions are linear in some variables and if the corresponding components of $\mathbf{h}_k$ are nonzero, then the approximation of constant derivatives are updated by nonzero values. Take a simple example. Let $f_j = x_1^2+2x_3$, $\mathbf{x} = [x_1\ x_2\ x_3]^T$ and the gradient $\mathbf{f}_j'(\mathbf{x}) = [g_1\ g_2\ g_3]^T$. Two components of the gradient, namely $g_2=0$ and $g_3=2$, are constants and can be determined accurately by perturbations. g_1 is the only component that needs to be updated. Suppose that $\mathbf{x}_k = [1\ 1\ 1]^T$, $\mathbf{h}_k = [0.5\ 0.5\ 0.5]^T$ and a perfect estimation of $\mathbf{f}_j'(\mathbf{x}_k)$ is given by $[2\ 0\ 2]^T$. The approximation of $\mathbf{f}_j'(\mathbf{x}_k+\mathbf{h}_k)$, as updated by the Broyden formula, would be $[2.167\ 0.167\ \ 2.167]^T$ (the true value is $[3\ 0\ 2]^T$).

(2) Along directions orthogonal to $\mathbf{h}_k$ the Jacobian is not updated:

$$\mathbf{G}_{k+1}\mathbf{p} = \mathbf{G}_k\mathbf{p}, \quad \text{for } \mathbf{p}^T\mathbf{h}_k = 0\,. \tag{4}$$

To overcome these difficulties, we devise a "weighted" update and adapt the special iterations of Powell [4].

IV. WEIGHTED BROYDEN UPDATE

In this method, we update the Jacobian matrix on a row-by-row basis. The jth row vector of the approximate Jacobian, denoted by $(\mathbf{g}_j)_k$, is an approximation to $\mathbf{f}_j'(\mathbf{x}_k)$, the gradient of f_j. Suppose that we know the Hessian of f_j and denote it by $\mathbf{H}_j$, then

$$\mathbf{f}'_j(\mathbf{x}_k + \mathbf{h}_k) = \mathbf{f}'_j(\mathbf{x}_k) + \mathbf{H}_j(\mathbf{x}_k)\,\mathbf{h}_k\,. \tag{5}$$

Analogously to (5), we can devise an update formula to obtain an approximation to $\mathbf{f}_j'(\mathbf{x}_k+\mathbf{h}_k)$ by

$$(\mathbf{g}_j)_{k+1} = (\mathbf{g}_j)_k + \alpha\,\mathbf{H}_j(\mathbf{x}_k)\,\mathbf{h}_k\,. \tag{6}$$

If we choose the coefficient α as

$$\alpha = \frac{f_j(\mathbf{x}_k + \mathbf{h}_k) - f_j(\mathbf{x}_k) - (\mathbf{g}_j)_k^T \mathbf{h}_k}{\mathbf{h}_k^T \mathbf{H}_{jk} \mathbf{h}_k} \tag{7}$$

then the linear model as given by (3) will be preserved, namely

$$f_j(\mathbf{x}_k + \mathbf{h}_k) - f_j(\mathbf{x}_k) = (\mathbf{g}_j)_{k+1}^T \mathbf{h}_k\,. \tag{8}$$

In practice we are very unlikely to have access to the Hessian of f_j. Even so, two basic facts are obvious: the Hessian of a quadratic function is constant and if f_j is linear in x_i then the ith row as well as the ith column of the Hessian contain only zeros. Hence, we propose the use of a constant diagonal matrix

$$\mathbf{W}_j = \mathrm{diag}\,[w_{1j} \dots w_{nj}], \quad w_{ij} > 0. \tag{9}$$

This results in a weighted Broyden update as follows.

$$(\mathbf{g}_j)_{k+1} = (\mathbf{g}_j)_k + \frac{f_j(\mathbf{x}_k + \mathbf{h}_k) - f_j(\mathbf{x}_k) - (\mathbf{g}_j)_k^T \mathbf{h}_k}{\mathbf{q}_{jk}^T \mathbf{h}_k} \mathbf{q}_{jk}\,, \tag{10}$$

where

$$\mathbf{q}_{jk} = \mathbf{W}_j \mathbf{h}_k = [w_{1j} h_{1k} \dots w_{nj} h_{nk}]^T\,. \tag{11}$$

The weights w_{ij} provide a measure of the linearity of f_j. If f_j is linear in x_i, we set $w_{ij}=0$. If f_j is nearly linear in x_i, we assign a small value to w_{ij}. It is clear from (10) that only the relative magnitude of the weights is important, not their absolute values. Consider the example we have used in the previous section, namely, $f_j = x_1^2 + 2x_3$. We set $w_{2j}=w_{3j}=0$ and $w_{1j}=1$. The approximate gradient given by (10) is $[2.5\ 0\ 2]^T$, compared to the result given by (2), namely $[2.167\ 0.167\ 2.167]^T$, and the true value $[3\ 0\ 2]^T$. The assignment of weights requires some knowledge of the functional relationship of $f_j(\mathbf{x})$ to determine $\mathbf{W}_j$. Such a knowledge may come from experience or may be found by performing a few perturbations. For instance, in circuit design optimization, it may be known that some parameters have little influence on the performance function over a particular frequency or time interval. An adaptive method for finding $\mathbf{W}_j$ is open for investigation.

V. POWELL'S SPECIAL ITERATIONS

The Broyden update is a rank-one method. This means that the approximate Jacobian is not updated along directions orthogonal to $\mathbf{h}_k$, as shown in (4). If some consecutive steps of optimization are collinear, the updating procedure may not converge. Powell [4] suggested a method that ensures the steps taken are "strictly linearly independent". This is accomplished by some special iterations which intervene between the ordinary iterations of optimization. The increment vector of such a special iteration is not calculated to minimize the error functions, instead it is set to a value that is intended to improve the accuracy of the approximation of first-order derivatives. The algorithm for computing such an increment vector, as derived by Powell [4], can be found in the Appendix.

VI. A HYBRID APPROXIMATION ALGORITHM

Given a starting point $\mathbf{x}^0$ for optimization, an initial approximate Jacobian is either provided by the user or computed using perturbations. The perturbation method may also be used with prescribed regularity, say, at every pth iteration of optimization. If this is the case, the Broyden update with or without weights, depending on whether the necessary knowledge of $\mathbf{f}(\mathbf{x})$ is available, is employed between perturbations. If perturbation (during optimization) is not desired the Broyden formula will be used throughout the optimization. In both cases, special iterations are introduced in conjunction with the Broyden update. In our actual implementation, a special iteration is skipped provided that the changes in the functions agree fairly well with the linear prediction by the approximate derivatives. This is considered to be true if

$$\| \mathbf{f}_j(\mathbf{x}_k + \mathbf{h}_k) - \mathbf{f}_j(\mathbf{x}_k) - \mathbf{G}_k\mathbf{h}_k \| < 0.1 \, \| \mathbf{f}_j(\mathbf{x}_k + \mathbf{h}_k) - \mathbf{f}_j(\mathbf{x}_k) \| \, . \tag{12}$$

Obviously, the purpose of this provision is to avoid unnecessary computations. This hybrid method, with its flexibility, will enable us to accommodate a large variety of problems whereby a better compromise between computational labor and accuracy can be achieved.

We have implemented our algorithm in a subroutine which calls a user-supplied routine for function values (e.g., a circuit simulation routine), carries out the approximation and returns the approximate gradient to an optimizer. It is independent of and transparent to the optimizer and the simulator. The optimizer sees only a routine that provides both function values and gradient. The simulator, on the other hand, sees an optimizer not requiring a gradient.

Some sophisticated optimizers employ distinct stages of optimization, for example, the 2-stage minimax and ℓ_1 optimizers in [1] and [2]. Usually a particular strategy is devised to achieve fast convergence near the solution where an approximate gradient of better accuracy may be desired. Our implementation allows different schemes of approximation to be used at different phases of optimization.

VII. NUMERICAL RESULTS

A large variety of problems have been solved using our new algorithm. The results have clearly demonstrated the effectiveness and efficiency of the algorithm. Its impact on practical microwave circuit optimization is best illustrated through applications to robust FET modelling, worst-case analysis and multiplexer design. Due to the size and complexity of these implementations they are presented in detail in Part II of our paper. In this section we will discuss some test problems. In the following, examples MM1 to MM7 are minimax problems in which we have to

$$\underset{\mathbf{x}}{\text{minimize}} \; \underset{j}{\max} \; \{f_j(\mathbf{x})\} , \tag{13}$$

where $\mathbf{x}$ is the vector of variables and f_j, $j = 1,\ldots,m$, are a set of nonlinear functions. Examples $\ell_1 1$ to $\ell_1 6$ are ℓ_1 problems in which we have to

$$\underset{\mathbf{x}}{\text{minimize}} \sum_{j=1}^{m} |f_j(\mathbf{x})| . \tag{14}$$

Comparisons of the computational effort for these examples using gradient approximation versus estimating derivatives entirely by perturbations are given in Tables I, II and III.

Problem MM1 We consider a well-established problem of a two-section transmission-line transformer as shown in Fig. 1. The reflection coefficient of the transformer is sampled at 11 normalized frequencies w.r.t. 1 GHz, namely, {0.5, 0.6, ..., 1.4, 1.5}. Madsen and Schjaer-Jacobsen [8] have shown that this is a singular problem when the characteristic impedences Z_1 and Z_2 are taken as variables while lengths ℓ_1 and ℓ_2 are kept constant at their optimal values ℓ_q, which is the quarter wavelength at the center frequency. Fig. 2 shows the minimax contours and illustrates the solution reported in [2], which is obtained using exact derivatives. If the derivatives were to be estimated by perturbations, 24 function evaluations would have to be performed. Using our gradient approximation, we obtained the solution, as shown in Fig. 3, after 18 function evaluations.

Problem MM2 It has also been reported in [2] that a regular minimax problem can be defined by choosing Z_1 and ℓ_1 as variables. The solution using exact derivatives is shown in Fig. 4. Fig. 5 shows the solution obtained using the approximate gradient algorithm.

Problem $\ell_1$1 We formulate an ℓ_1 problem in which the reflection coefficient of the two-section transformer at the optimal point was taken as a measurement from which we attempt to identify the values of Z_1 and Z_2. The solutions obtained with the gradient estimated entirely by perturbations and by our new method are illustrated in Figs. 6 and 7, where 42 and 27 function evaluations were required, respectively.

Problems MM3 and MM4 Two examples of multi-cavity filter design [9] are considered. The reflection coefficient in the passband is minimized and the transducer loss over the stopband is maximized. The couplings and transformer ratios of the filters are optimized. Example MM3 is a 4th order filter having 4 variables and MM4 a 6th order filter having 6 variables.

Problems MM5, MM6 and MM7 This is a problem proposed by Brent [10] for which the Newton-Raphson method is not globally convergent. It is to solve the system of 2 nonlinear equations

$$4(x_1 + x_2) = 0\,, \tag{15}$$

$$(x_1 - x_2)(x_1 - 2)^2 + x_2^2) + 3x_1 + 5x_2 = 0\,.$$

Three starting points were used, namely,

$$\begin{bmatrix} 2.0 \\ 2.0 \end{bmatrix}, \begin{bmatrix} 2.0 \\ 0.0 \end{bmatrix}, \begin{bmatrix} 2.0 \\ 1.0 \end{bmatrix}.$$

Problem $\ell_1 2$ This is a data-fitting problem [5] in which e^y is approximated by a third-order rational function over the interval $-1 \le y \le 1$. The error functions are

$$f_j(x) = \frac{x_1 + x_2 y_j}{1 + x_3 y_j + x_4 y_j^2 + x_5^3} - e^{y_j}\,, \tag{16}$$

$$y_j = -1 + 0.1\,(j - 1)\,,\ j = 1, \ldots, 21.$$

Problem $\ell_1 3$ This is a nonlinear ℓ_1 modelling problem due to El-Attar et al. [11] of finding a third-order model for a seventh-order system involving 6 variables and 51 functions.

$$\begin{aligned} f_j(x) &= x_1 e^{-x_2 t_j} \cos(x_3 t_j + x_4) + x_5 e^{-x_6 t_j} - y_j\,, \\ y_j &= \frac{1}{2} e^{-t_j} - e^{-2t_j} + \frac{1}{2} e^{-3t_j} + \frac{3}{2} e^{-1.5t_j} \sin(7t_j) + e^{-2.5t_j} \sin(5t_j)\,, \\ t_j &= 0.1\,(j - 1)\,, j = 1, \ldots, 51. \end{aligned} \tag{17}$$

The solution of this problem using exact derivatives has been reported in [1].

Problem $\ell_1 4$ This example, due to El-Attar et al. [11], involves finding an ℓ_1 solution to the set of nonlinear equations

$$
\begin{aligned}
f_1(\mathbf{x}) &= x_1^2 + x_2^2 + x_3^2 - 1\,,\\
f_2(\mathbf{x}) &= x_1^2 + x_2^2 + (x_3 - 2)^2\,,\\
f_3(\mathbf{x}) &= x_1 + x_2 + x_3 - 1\,,\\
f_4(\mathbf{x}) &= x_1 + x_2 - x_3 - 1\,,\\
f_5(\mathbf{x}) &= 2x_1^3 + 6x_2^2 + 2(5x_3 - x_1 + 1)^2\,,\\
f_6(\mathbf{x}) &= x_1^2 - 9x_3\,.
\end{aligned}
\tag{18}
$$

The solution, as reported in [1], is singular.

Problem $\ell_1 5$ The problem of fault location is considered for a mesh network consisting of 20 elements. The two faulty elements deviate from their nominal values by 50 percent and tolerances of 5 percent are associated with the other elements. Using measurements on the circuit with a single excitation, the actual faults are to be identified. More detail can be found in [1].

Problem $\ell_1 6$ In this paper, we have proposed a weighted Broyden update in order to exploit possible special structure of the Jacobian of a system such as sparsity. A class of equations has been constructed by Broyden [3], namely

$$
\begin{aligned}
f_1(\mathbf{x}) &= (3 - 0.5x_1)x_1 + 2x_2 - 1\,,\\
f_j(\mathbf{x}) &= x_{j-1} - (3 - 0.5\,x_j)\,x_j + 2x_{j+1} - 1\,,\quad j = 2, 3, \ldots, n-1\,,\\
f_n(\mathbf{x}) &= x_{n-1} - (3 - 0.5\,x_n)\,x_n - 1\,.
\end{aligned}
\tag{19}
$$

In this tridiagonal system the function f_j is linear in x_k for all $k \neq j$. According to the discussion in Section IV, we can define a set of weights, in which $w_{jj} = 1$ and $w_{kj} = 0$ for $k \neq j$, for the update formula (10). We have solved this problem for $n = 5$, 10 and 20. From the results shown in Table III, it is clear that the weighted update is more efficient than the original Broyden formula and the improvement becomes more significant as the size of the system increases. In Part II the use of weights is further illustrated through practical applications which are far more complicated.

VIII. CONCLUSIONS

A new algorithm for gradient approximation has been presented. Methods of integrating such an algorithm with powerful gradient-based optimization techniques have been proposed. The effectiveness and flexibility of the new methods have been illustrated by solving a large variety of problems. It has also demonstrated significant improvement in computational efficiency compared to the more conventional method of perturbations. The use of this method facilitates application of advanced optimization tools to a broader range of practical problems where analytical evaluation of partial derivatives is either unavailable or very complicated and tedious. Implementations of significant interest to microwave circuit engineers are described in Part II [7].

REFERENCES

[1] J.W. Bandler, W.K. Kellermann and K. Madsen, "An nonlinear ℓ_1 optimization algorithm for design, modelling and diagnosis of networks", IEEE Trans. Circuits and Systems, vol. CAS-34, 1987.

[2] J.W. Bandler, W.K. Kellermann and K. Madsen, "A superlinearly convergent minimax algorithm for microwave circuit design", IEEE Trans Microwave Theory Tech., vol. MTT-33, 1985, pp.1519-1530.

[3] C.G. Broyden,"A class of methods for solving nonlinear simultaneous equations", Mathematics of Computation, 19,1965, pp. 577-593.

[4] M.J.D. Powell, "A Fortran subroutine for unconstrained minimization, requiring first derivatives of the objective functions", Atomic Energy Research Establishment, Harwell, Berkshire, England, Report AERE-R.6469, 1970, pp. 20-27.

[5] K. Madsen, "Minimax solution of nonlinear equations without calculating derivatives", Mathematical Programming Study, 3,1975, pp. 110-126.

[6] W.M. Zuberek, "Numerical approximation of gradients for circuit optimization", Proc. 27th Midwest Symp. Circuits and Systems (Morgantown, WV, 1984), pp. 200-203.

[7] J.W. Bandler, S.H. Chen, S. Daijavad and K. Madsen, "Efficient optimization with integrated gradient approximations, Part II: Implementation", submitted as a companion paper.

[8] K. Madsen and H. Schjaer-Jacobsen, "Singularities in minimax optimization of networks", IEEE Trans. Circuits and Systems, vol. CAS-23, 1976, pp.456-460.

[9] J.W. Bandler, S.H. Chen and S. Daijavad, "Exact sensitivity analysis for optimization of multi-coupled cavity filters", Int. J. Circuit Theory and Applic., vol. 14, 1986, pp.63-77.

[10] R.P. Brent, "On the Davidenko-Branin method for solving simultaneous nonlinear equations", IBM J. Research and Development, vol. 16, 1972, pp. 434-436.

[11] R.A. El-Attar, M. Vidyasagar and S.R.K. Dutta, "An algorithm for ℓ_1-norm minimization with application to nonlinear ℓ_1-approximation", SIAM J. Numer. Anal., vol.16, 1979, pp.70-86.

APPENDIX

We have used Powell's special iterations [4] to improve the accuracy of the gradient approximation. This is achieved by applying the Broyden update along a sequence of directions that satisfies a "strict linear independence condition" [4]. The algorithm for computing the increment vector for a special iteration, as derived by Powell [4], is shown as follows.

We denote the increment vector of the kth iteration by $\mathbf{h}_k$. A sequence of $n \times n$ (n being the number of optimization variables) orthogonal matrices $\mathbf{D}_k$ is constructed as follows. $\mathbf{D}_1$ is set to an identity matrix. At each iteration $\mathbf{D}_k$ is revised to yield $\mathbf{D}_{k+1}$. Following the notation of Powell, we use $\boldsymbol{\eta}_1^T, \boldsymbol{\eta}_2^T, \ldots, \boldsymbol{\eta}_n^T$ for the rows of $\mathbf{D}_k$. At a special iteration, the increment vector $\mathbf{h}_k$ is set to a multiple of the first row vector of $\mathbf{D}_k$:

$$\mathbf{h}_k = \Lambda_k \boldsymbol{\eta}_1 , \tag{A1}$$

where $\Lambda_k > 0$ is a parameter controlling the step size of $\mathbf{h}_k$. Usually Λ_k is set to the step size used for the latest ordinary iteration.

It remains to be described as how to revise the matrix $\mathbf{D}_k$. We use $\boldsymbol{\zeta}_1^T, \boldsymbol{\zeta}_2^T, \ldots, \boldsymbol{\zeta}_n^T$ for the rows of $\mathbf{D}_{k+1}$.

For a special iteration, we simply let

$$\boldsymbol{\zeta}_i = \boldsymbol{\eta}_{i+1} , \qquad i=1, \ldots, n-1 , \tag{A2}$$

$$\boldsymbol{\zeta}_n = \boldsymbol{\eta}_1 .$$

For an ordinary iteration, the following steps take place.

1. Compute $\sigma_i = \boldsymbol{\eta}_i^T \mathbf{h}_k$, $i=1, \ldots, n$. (A3)

2. Find t which is the greatest integer such that $\sigma_t \neq 0$.

3. Let $\alpha_t = 0$ and $\boldsymbol{\xi}_t = \mathbf{0}$. For $i = t-1, t-2, \ldots, 1$, compute

$$\boldsymbol{\xi}_i = \boldsymbol{\xi}_{i+1} + \sigma_{i+1}\boldsymbol{\eta}_{i+1} ,$$

$$\alpha_i = \alpha_{i+1} + \sigma^2_{i+1} , \qquad \text{(A4)}$$

$$\boldsymbol{\zeta}_i = \frac{\alpha_i \boldsymbol{\eta}_i - \sigma_i \boldsymbol{\xi}_i}{\sqrt{\alpha_i(\alpha_i + \sigma_i^2)}}$$

4. Let $\boldsymbol{\zeta}_i = \boldsymbol{\eta}_{i+1}$, $i = t, t+1, \ldots, n-1$. (A5)

5. Let

$$\boldsymbol{\zeta}_n = \frac{\mathbf{h}_k}{\sqrt{\mathbf{h}_k^T \mathbf{h}_k}} \qquad \text{(A6)}$$

TABLE I

COMPARISON OF COMPUTATIONAL EFFORT FOR MINIMAX EXAMPLES

Problem	Number of Function Evaluations	
	Numerical Differentiation	Gradient Approximation
MM1	24 (8)	18 (10)
MM2	24 (8)	18 (12)
MM3	59 (11)	30 (18)
MM4	84 (11)	66 (41)
MM5	9 (3)	5 (3)
MM6	32 (10)	19 (14)
MM7	29 (9)	14 (11)

Note: the entries in parentheses are numbers of optimization iterations.
The solutions obtained using gradient approximations agree with those obtained using numerical differentiation to 5 significant figures.

TABLE II

COMPARISON OF COMPUTATIONAL EFFORT FOR ℓ_1 EXAMPLES

Problem	Number of Function Evaluations	
	Numerical Differentiation	Gradient Approximation
$\ell_1 1$	42 (14)	27 (19)
$\ell_1 2$	54 (9)	32 (19)
$\ell_1 3$	105 (15)	63 (40)
$\ell_1 4$	71 (17)	65 (48)
$\ell_1 5$	147 (7)	34 (10)

Note: the entries in parentheses are numbers of optimization iterations.
The solutions obtained using gradient approximations agree with those obtained using numerical differentiation to 5 significant figures.

TABLE III

COMPARISON OF COMPUTATIONAL EFFORT FOR EXAMPLE $\ell_1 6$

Size of the system	Number of Function Evaluations		
	Case 1	Case 2	Case 3
n = 5	36 (6)	17 (9)	13 (7)
n = 10	66 (6)	25 (10)	19 (7)
n = 20	126 (6)	39 (13)	29 (7)

Case 1: Using numerical differentiation

Case 2: Using the original Broyden update

Case 3: Using the weighted Broyden update

The entries in parentheses are numbers of optimization iterations
The solutions obtained using gradient approximations agree with those obtained using numerical differentiation to 5 significant figures.

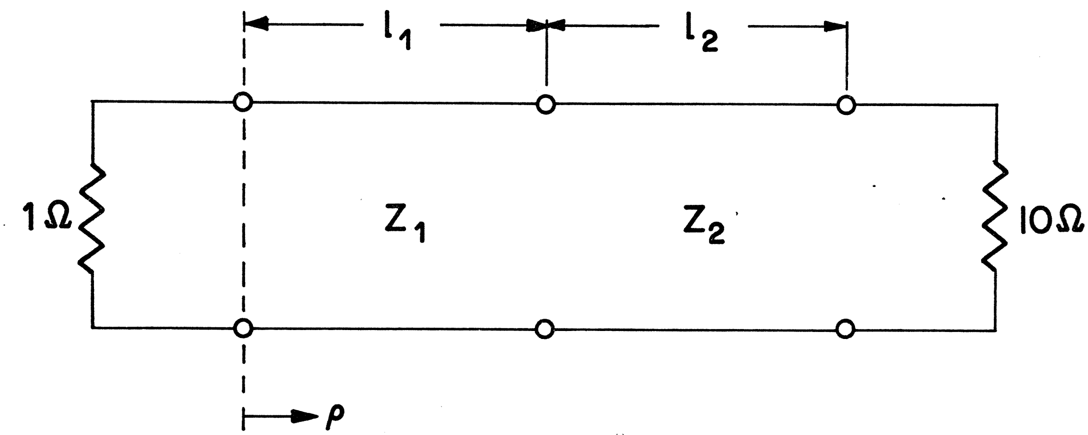

Fig. 1 Two-section, 10:1 transmission-line transformer.

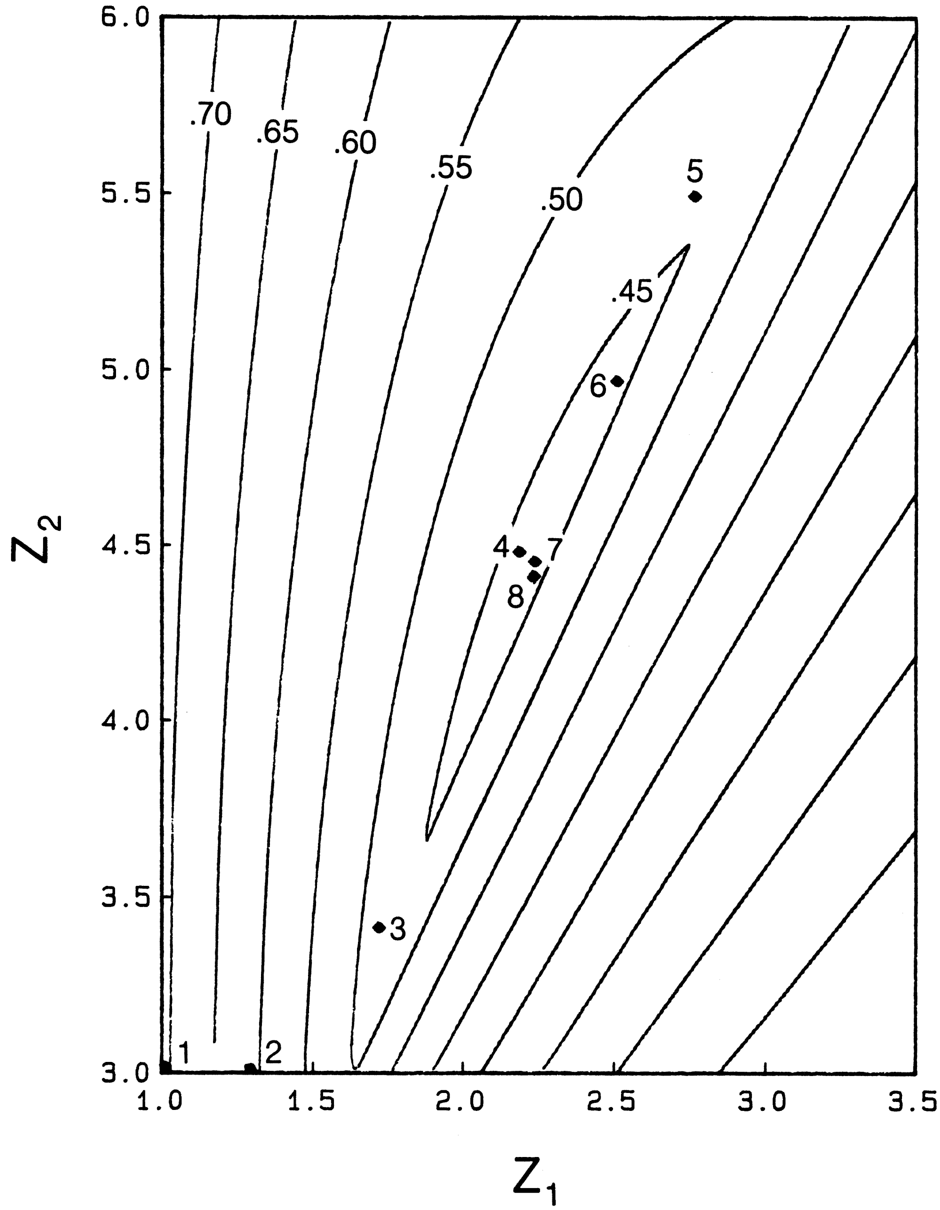

Fig. 2 Contours of (13) for the two-dimensional singular minimax problem MM1 arising from optimization of the two-section transmission-line transformer. Eight iterations using exact derivatives are illustrated.

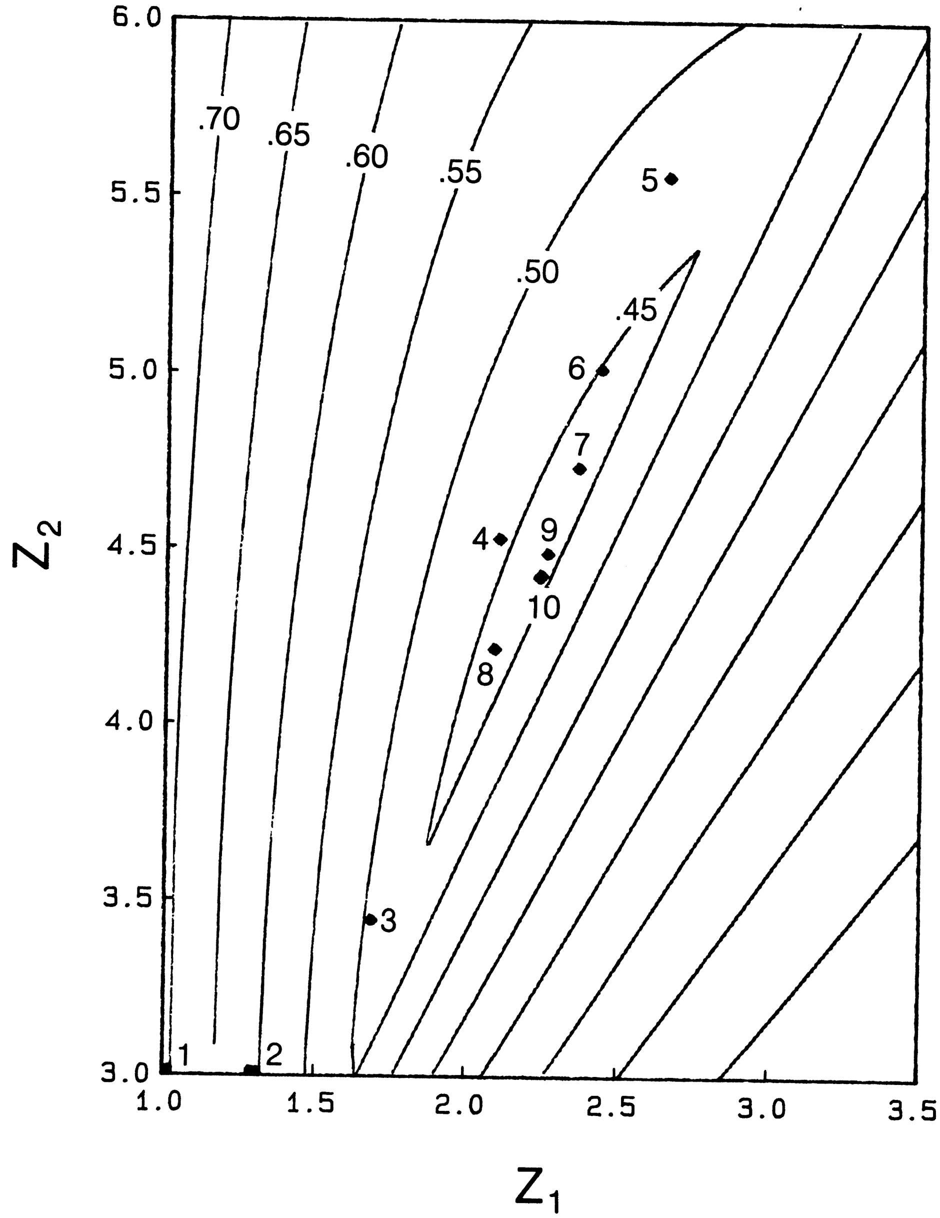

Fig. 3 The same problem as shown in Fig. 2 is solved using 10 iterations of our algorithm for gradient approximations.

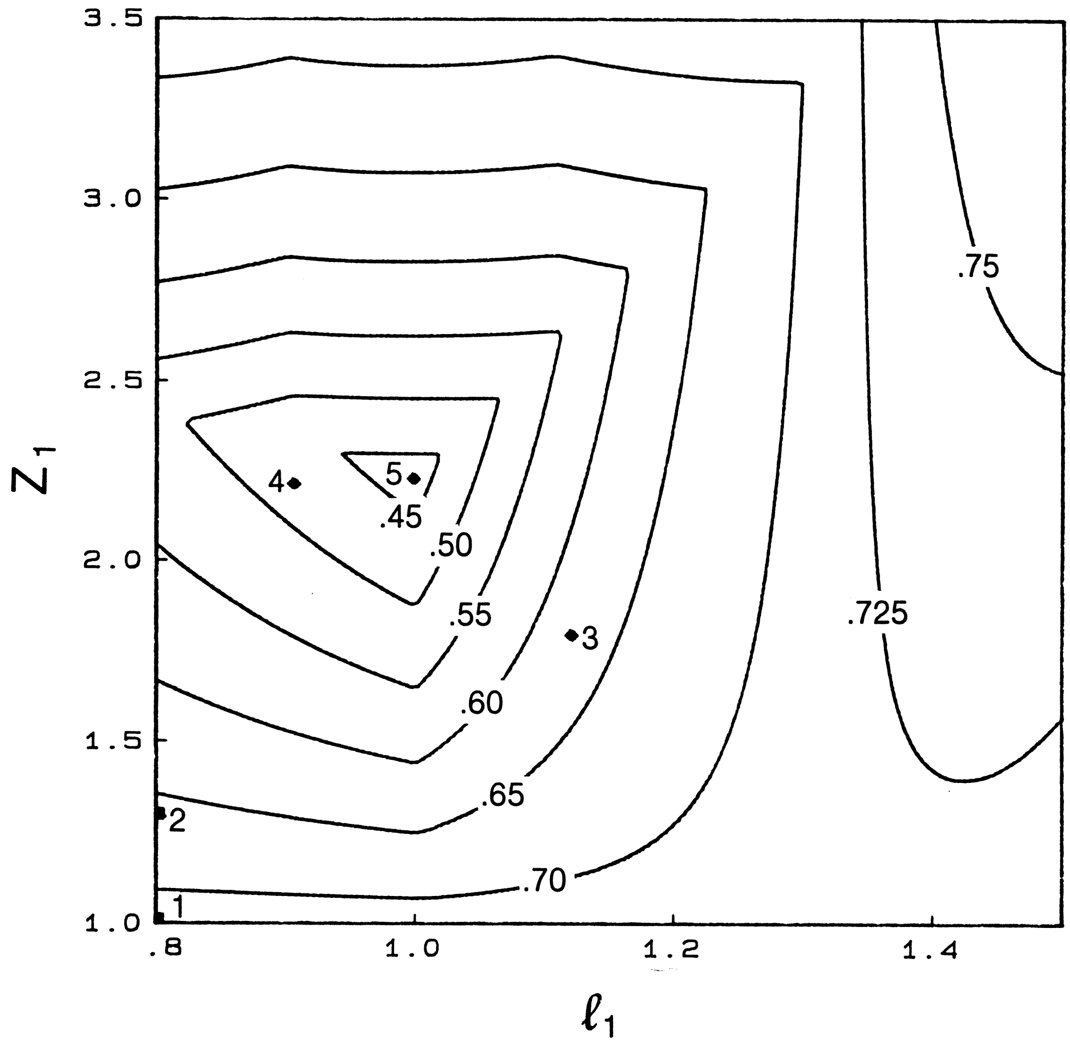

Fig. 4 Contours of (13) for the two-dimensional regular minimax problem MM2 arising from optimization of the two-section transmission-line transformer with variables ℓ_1/ℓ_q and Z_1. The derivatives are estimated entirely by perturbations. A total of 8 iterations and 24 function evaluations are required to reach to the solution. The first 5 iterations are shown.

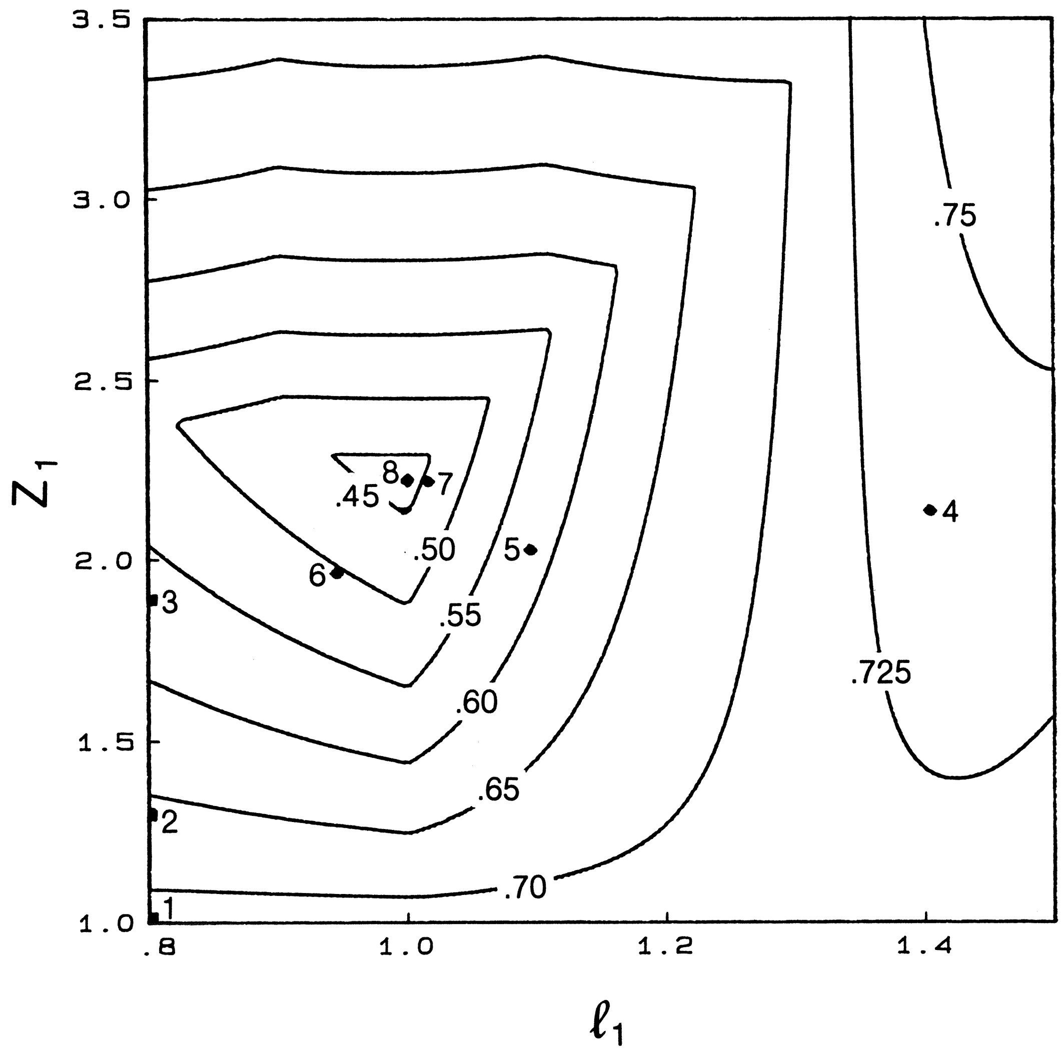

Fig. 5 The same problem as shown in Fig. 4 is solved using our algorithm for gradient approximations. The solution requires 12 iterations and 18 function evaluations. The first 8 iterations are shown.

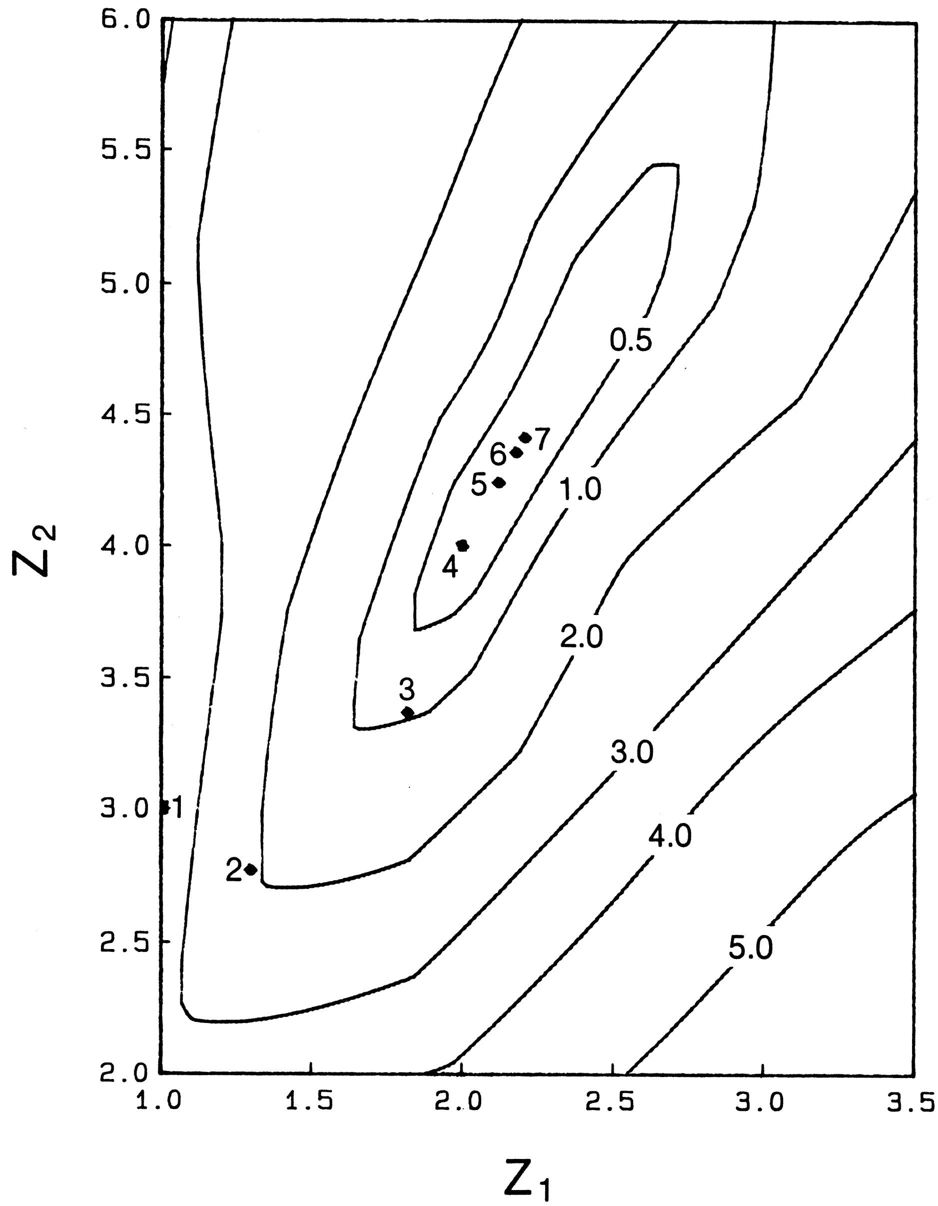

Fig. 6 Contours of (14) for the two-dimensional ℓ_1 problem $\ell_1 1$ arising from parameter identification of the two-section transmission-line transformer. The solution obtained with the derivatives estimated entirely by perturbations requires 14 iterations and 42 function evalutions. The first 7 iterations are illustrated.

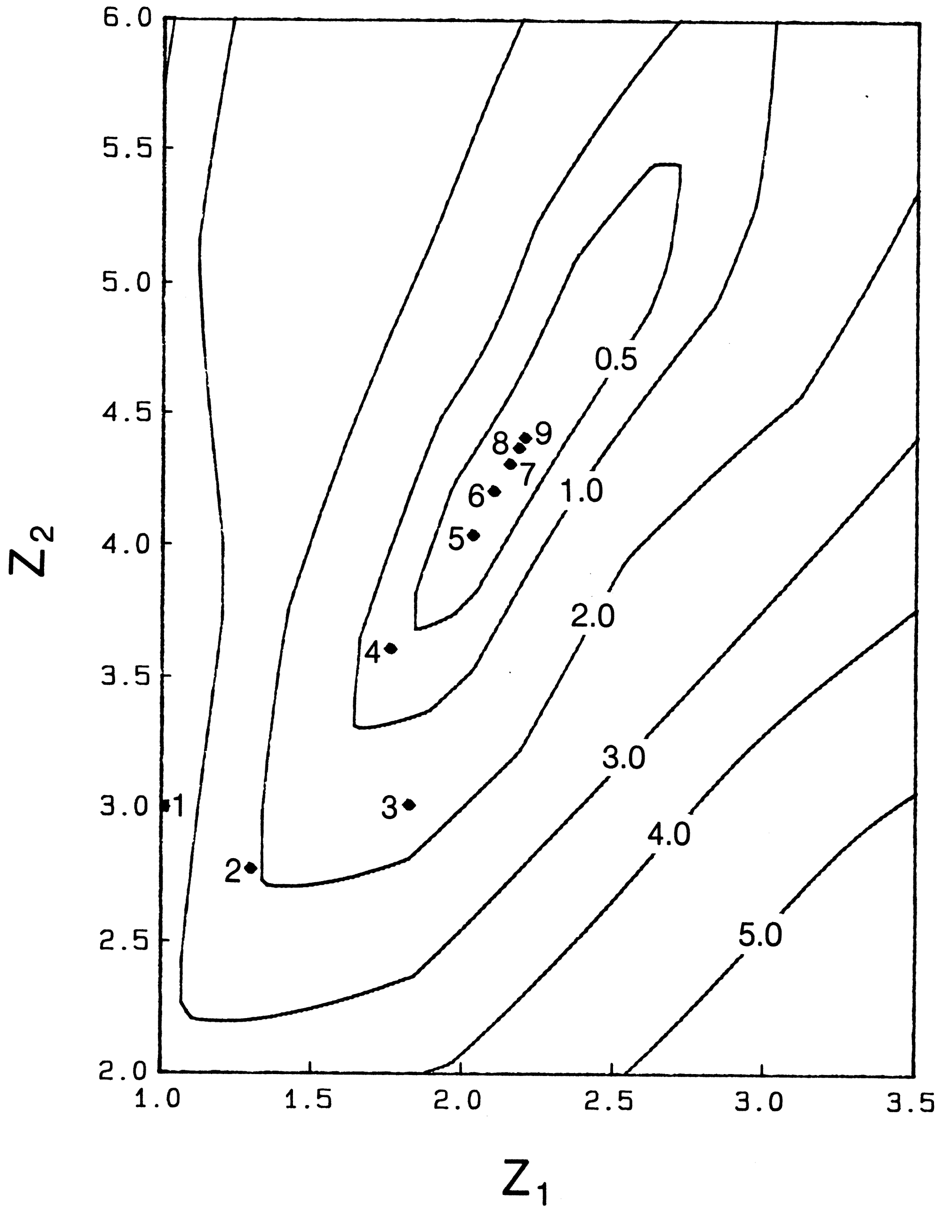

Fig. 7 The same problem as shown in Fig. 6 is solved using our algorithm for gradient approximations. The solution requires 19 iterations and 27 function evaluations. The first 9 iterations are illustrated.