A GLOBAL OPTIMIZATION ALGORITHM

Ronald W. Becker, G. V. Lago
University of Missouri at Columbia
Columbia, Missouri

ABSTRACT

A global optimization algorithm using a Mode -Seeking Program and Monte Carlo analysis is described. The parameters of a function to be minimized can be varied at random over any desired range, subject to any restrictions. The algorithm is especially adaptable to circuit theory design.

I. INTRODUCTION

The advent of the digital computer has permitted the solutions of many optimization problems whose solutions would have been impractical otherwise. Gradient methods, such as Fletcher and Powell's method [1], often require many derivative calculations and function evaluations in each step of the iterative process. Alternatives to these gradient methods are methods which require only function evaluations. Rosenbrock [2], Hooke and Jeeves [3], and Nelder and Mead [4] (Simplex Method) have presented methods which require only function evaluations. The method presented in this paper is an extension of random search methods as discussed by Brooks [5]. The latter methods are sometimes referred to as direct search methods, whereas the gradient methods are also referred to as descent methods. Descent methods can require many derivative and function evaluations, while direct methods require many more function evaluations but no derivative calculations.

The random search procedures to be discussed will be concerned with minimization of a function E, which, in the case of network design, includes some design criteria. Usually E will be a measure of the error between a desired response F and a randomly generated response F'. E may be a least squares error, a Chebyshev error or a coefficient matching error function. F' will generally be a function of two sets of variables: first, the set of parameters, X_i, i=1,---,k, to be randomly generated in order to minimize E; and, secondly, the set of points, Y_j,j=1,---,w, where F and F' are measured. For network design the set Y_j may correspond to a set of frequencies at which F and F' are measured.

In any random search method trials are made at combinations of parameters selected in some random fashion. The estimate of the optimum parameter combination is simply the combination of parameters which produced the lowest value for the error function. This process can also be classified as a Monte Carlo method [6].

Brooks has reviewed several random search techniques: the simple random method, the stratified random method and the creeping random method [7]. The simple random method basically makes trials at points X_i selected at random throughout the whole experimental region. The trial with the lowest error value is called the optimum solution. In the stratified random sampling method the experimental region is divided up into subregions by a regular grid. All of these rectangular subregions are of equal size and the number of subregions is made to correspond to the number of desired trials. A trial is made at a randomly selected point within each subregion. The optimum solution is that combination of parameters which has the lowest error evaluation. The creeping random method is different from the previous two methods in that it is a sequential random procedure. A guess is made initially and then a set of random trials is made around the initial guess. The random trials may be selected from a normal distribution that is centered on the initial guess. The optimal point of the set of trials is retained as the center of another hypothetical normal distribution, from which another

set of random trials is sampled. The procedure is repeated, and the variance of the normal distribution is diminished each time, thereby converging to a local minimum.

II. RANDOM SEARCH OPTIMIZATION BY MODE-SEEKING

A new algorithm has been developed to improve upon simple random methods by implementing a mode-seeking program. The algorithm is a two or more step procedure depending on the desired degree of minimization.

Step one is essentially a simple random search over the entire experimental parameter region. Instead of retaining only the point with the minimum error as the unaltered simple random method does, the proposed method retains a predetermined number of points which have the lowest error values of all the original trials. If the error function were unimodal over the entire experimental parameter region, then the set of retained points would be centered about the absolute minimum of the unimodal function. The points could be said to form a cluster about the minimum. The compactness of the cluster would be dependent upon the number of original random trials and the number of points retained. If the error function were bimodal and the modes were symmetric with equal absolute minimums, then the retained points would geographically form two clusters. Suppose the error function were bimodal, but one mode had an absolute minimum much lower than the other mode. It is still likely that the retained points would form two clusters, though most of the points would probably be located around the lowest absolute valued mode, particularly if the two modes had similar contours. These ideas can be extended to include functions with many modes, functions with modes within modes, and functions which have spike-shaped modes. Irregular error functions will naturally require more initial trials to obtain a good set of retained points and, therefore, clusters which are representative of modes.

The retained points are grouped into clusters by an algorithm based on the geographical location of the points with respect to each other. Details of the particular algorithm will be discussed later. Since this program is designed to find clusters which correspond to modes, the program will be called a Mode-Seeking Program.

At this point the clusters found by the Mode-Seeking Program are graded and rated. The clusters are graded by searching in each cluster for the retained point which has the lowest error value. The clusters are then rated according to the points found in the grading process. The best cluster would be the cluster containing the point with the lowest error value. Consider again the bimodal error function with unequal absolute modal minimums. Assume that a proper number of initial trials are made and that the retained points are classified into two clusters. Then, with a very high probability, the cluster located around the lower valued mode would be rated as the better cluster.

A subregion of the original experimental region is constructed to contain all of the retained points of each cluster. The bimodal error function would have two subregions, one for each cluster. In some instances it might be advantageous to enlarge the subregions somewhat to insure that the absolute minimums are contained within the subregions. The random search method has essentially been extended to create and store information concerning the topology of the error function. Additional steps will use this topological information to locate the minimum of the error function. This concludes the first step of the proposed random search method using a mode-seeking program.

Step two of the proposed random search method treats each subregion found in the first step as an experimental region upon which a new random search is undertaken. For example, assume that two clusters were found for the bimodal error function by the first step, then the second step takes a set of random trials on each subregion formed by the two clusters. The second step proceeds as in the first step: it retains a certain number of

the points with the lowest error values, clusters the points, rates the clusters, and finally creates sub-subregions. It is important to specify that the second step performs a random search first on the best cluster, next on the second best cluster and so on. Theoretically, the absolute minimum of the error function should lie somewhere within the best cluster. Consequently, it would appear that the second step should only perform a random search on the best cluster. In practice, however, it is a good idea to include the second best cluster and possibly the third to allow for probability discrepancies. Much computer time can be saved by limiting the number of subregions searched by the second step.

The pattern of the proposed random search method with a mode-seeking program should be evident. If necessary, the third step and possibly higher steps may be taken in a similar manner to the second step. Each additional step should reduce the subregions to smaller and smaller intervals and in some cases, to the exact absolute minimum. A terminating error bound based on the desired level of solution accuracy may also be provided as the point below which no further improvement is necessary.

Global Optimization and Constraints

The proposed random search method with a mode-seeking program is a global optimization algorithm. The initial experimental region under investigation may be as large as desired and have as many parameters as desired. Recall that the parameters of F were X_i,i=1,---,k, where k is the order of the parameter space. The parameters may also have any constraints placed on them since the parameters can be generated very easily to satisfy the constraints.

Such an optimization procedure has many advantages. Gradient methods usually work well and fast; however, these methods necessarily require that the error function be unimodal over a region. The proposed random search method would work especially well under similar circumstances. If the error function is multi-modal, such as in typical network problems, and gradient optimization methods are applied, the result will probably be a local minimum of disputable merit. Successful gradient techniques rely heavily on the initial point from which the optimization proceeds. A multi-modal error function would be solvable by the proposed random search technique. Such a problem may take many trials and several algorithm steps in the process. The simple random search method and the stratified random search method might eventually find the global minimum, but they would certainly take many more trials than the proposed random search algorithm. The creeping random method, as stated before, would only find a local minimum for a multi-modal error function.

Parameter scaling is often necessary with gradient methods. In particular, the element values of a network must often be normalized to use Fletcher and Powell's method [8]. Parameter scaling, on the other hand, is not necessary with the proposed random search algorithm.

The proposed random search method will also work on problems with nonlinearities and discontinuities. These, of course, must be incorporated within the response functions F and F' and the error function E. If this can be done, then the proposed random search algorithm should find the optimum solution.

The circuit designer may also incorporate another novel type of constraint on the parameter values. Suppose the designer wishes to use only commercially available resistors, capacitors, inductors, transistors, and other components. It would be possible to preload the experimental region with commercial values as parameter points. The end result would be an optimum set of available components. Gradient optimization techniques, on the other hand, usually result in a set of extremely precise parameter values with little likelihood of locating or making the exact element value as specified.

Number of Trials Required

A sufficient number of trials must be taken if the proposed random search method can be expected to be meaningful. The researcher using this method must estimate before running the program the number of trials needed for the initial step and the number of trials for succeeding steps.

Recall the concept of the subregion or subregions as related to the entire experimental region. The first step of the proposed random search algorithm defines a subregion (subregions), in the form of clusters, where the optimum solution exists. Some of the initial random trial points, therefore, must fall within the subregion (subregions) where they will be retained. Let P denote the relative sizé of the subregion or sum of subregions with respect to the entire experimental region. It may be seen that P is also the probability that a randomly selected trial point will fall within the desired subregion or one of the subregions. The distribution of random trials is simply a binomial distribution.

Let m denote a number of retained points and n the number of random trials. Then from the binomial distribution, the probability, S', of m successes in n trials is

$$S' = \frac{n!}{m!(n-m)!}P^m(1-P)^{n-m}$$

It is possible to specify S', P, and m and then calculate n. However, due to the factorials, finding n can be very difficult. Since any extra trials may only reduce the size of the subregion (subregions) more than expected, an easier estimate of the number of trials was developed. Let S denote the probability of at least one trial falling in the subregion out of n trials. Then

$$S = 1 - (1-P)^n ,$$

which is one minus the probability of no points falling in the subregion. Solving for n,

$$n = \log(1-S)/\log(1-P) .$$

If n trials have probability S of at least one successful trial, then the product mn trials will have a probability greater than S of at least m successful trials. The product mn is an over-estimate but any excessive trials are also useful.

The proposed random search algorithm has been programmed so that the researcher must provide the error function E, the parameter generation scheme, the number of trials for each step, and the number of retained points for each step before executing the program. The method could be improved, however, by programming the algorithm into a conversational mode on any capable computer. A user could monitor the progress of the method at suitable points and, if necessary, the user could direct more trials to be taken and perhaps fewer or more points to be retained. One could also eliminate searching through doubtful subregions and perhaps intensify the search in other subregions. A researcher experienced in optimization would certainly appreciate the conversational mode version of the proposed algorithm.

Random Parameter Generation

Any number of random parameter generating schemes can be implemented within the proposed algorithm. There is no one best scheme suited for all situations. Experience has shown that special-purpose generating schemes can save considerable computer time. The only limit on the number and type of generating schemes is the ingenuity of the program user.

Basically all generating schemes on a digital computer use a multiplicative congruential method based on the relation (involving non-negative integers k) $Z_{i+1} = aZ_i \pmod k$ $(0 \leq Z < k)$, which means that the expression aZ_i is to be divided by k^1[9]. The constant a is chosen to provide speed, a long period, and good statistical behavior. Since only the remainder is retained on division by k, the period cannot be greater than k. Therefore, k

is usually chosen to be one more than the largest computer integer. It is then unnecessary to divide by k, since the computer cannot register any product larger than k-1. This automatically provides the remainder. The set of Z's are the random integers which are then converted by additional user-supplied schemes into various sets of random parameters. There are numerous tests available, such as the frequency, serial, gap and poker tests, which test the randomness of a set of numbers [10]. For the purposes of the proposed random search method, a uniform distribution of the random parameters is usually best.

As an illustration, suppose random integers are desired between 0 and 10. Suppose also that a program exists which generated the Z's. Then a Fortran IV statement function might be defined as

```
INTGR(Z,J) = Z(mod J).
```

The integers, 0<I<10, could be produced by

```
I = INTGR(Z,9) + 1.
```

Say for example, the following set of integers need to be produced randomly: 1,2,3,---,10,20,30,---,100,200,300,---,900. A solution would be

```
I = (INTGR(Z,9) + 1)*(10.0**INTGR(Z,3)).
```

Finally, suppose the following set of integers need to be produced randomly: -900,-800,---,-100,-90,-80,---,-10,-9,-8,----1,1,2,---9,10,20,---,90,100, 200,---,900. A solution would be:

```
I = (-1)**INTGR(Z,99)*(INTGR(Z,9) + 1)*(10.0**INTGR(Z,3)).
```

Mode-Seeking Program

The proposed random search method will work well only if the mode-seeking program works well. The mode-seeking program may also be referred to as a clustering program. Basically, the program must input a set of points in K space and output the points in clusters or groups. The points are grouped into clusters according to their geographical location with respect to one another. Unfortunately, the mode-seeking program has no knowledge concerning the number of clusters to expect, so it must be programmed to decide internally how many clusters best fit all of the input points.

The proposed random search method uses a mode-seeking program developed by Bryan [11]. This program performs an estimate of the true positive density function by using potential functions, followed by a hill climbing scheme which finds the modes. The program uses a Mahalonabis distance transformation to allow very large magnitudes between parameters. This may occur for example, when one parameter may be in the micro-farad range and another may be in the kilo-ohm range. Tests have shown that Bryan's mode-seeking program works very well; however, it is too detailed to present in this paper. There are probably additional clustering algorithms available which would work satisfactorily in the proposed random search method.

III. EXAMPLES

Example #1--Four Minimums

A simple but very illustrative example was optimized by the proposed random search method. The example was devised to have four minimums, the solutions of

$$E = [|x_1| - 5]^2 + [|x_2| - 5]^2.$$

It was assumed that the minimums were in the intervals $-10^7 < x_1 < +10^7$ and $-10^7 < x_2 < +10^7$, large enough to illustrate the global capacity of the proposed algorithm. A random number generation scheme, similar to the last example in Section II, was written.

The following notation has been developed to simplify example descriptions:

R = set of all points in the initial experimental region,

t_i = set of mn random trials for step i,

r_i = set of m retained points for step i,

and R_a = set of all points in subregion determined by cluster 'a'.

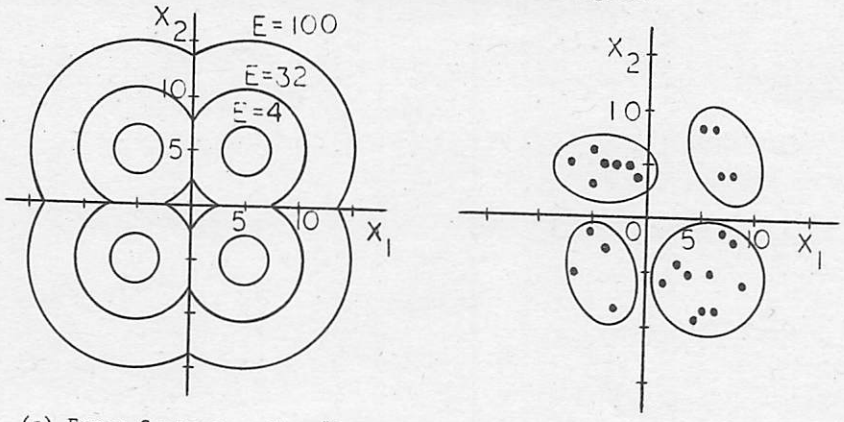
For this example, $R = 126^2$ or 15,876 points since x_1 and x_2 are each defined for X_{ij}; i = 1,---,126, j = 1,2.

Figure 1.a is a plot of some theoretical E (Error) function contours. The solid lines are constant values of E as indicated.

Suppose a subregion reduction factor of 25 is sought with 95% probability of at least one point falling in this subregion. Then

$$n = \log (1-.95)/\log (1-.04)$$
$$n = 73.$$

Suppose also that 25 points are to be retained. Then mn = 1835, and for purposes of rounding, t_1 is set at 2000 trials. Two thousand trials are then generated at random, E is evaluated at each trial, and the 25 points in r_1 are clustered by the mode-seeking program into four clusters as shown in Figure 1.b. Some of the clusters are centered around the minimums; (-5, -5), (-5,5), (5,-5), and (5,5), while some are not. However, each cluster does contain the proper minimum within its own subregion.



(a) Error Contours - Ex. #1 (b) Clusters - Ex. #1

Figure 1

Step two of the proposed algorithm was run on each of the four subregions with t_2 = 100 points and r_2 = 5 points. In every subregion examined by step two, the exact minimum, E = 0, was found. Step two was the final step for this example.

This example points out several important aspects of the proposed optimization method: its global capability, the calculation of mn, the power and accuracy of the mode-seeking program, and the relatively small number of trial points needed to find the four absolute minimums.

Example #2--Parameter Constraints

This example was suggested first by Beale [12] and is recognized as a standard test problem for optimization methods. The nonlinear function to be minimized is

$$E = 9-8x_1-6x_2-4x_3+2x_1^2+2x_2^2+x_3^2+2x_1x_2+2x_1x_3$$

subject to the constraints

$$x_1 \geq 0,\ x_2 \geq 0,\ x_3 \geq 0,$$

and

$$x_1 + x_2 + 2x_3 \leq 3.$$

The last constraint is active. Beale states that the minimum is E = 1/9 at $x_1 = 4/3$, $x_2 = 7/9$, and $x_3 = 4/9$.

From the statement of the problem, the ranges on the x's are:

$$0 < x_1 < 3$$
$$0 < x_2 < 3$$
$$0 < x_3 < 1.5$$

These values define the initial experimental region. The x's can only be randomly generated to a definite number of decimal places and, in this case, two decimal digits were chosen. The fortran statements to generate the x's are:

```
10 x1 = INTGR(Z,299)*1.0E-2 + .01
   x2 = INTGR(Z,299)*1.0E-2 + .01
   x3 = INTGR(Z,149)*1.0E-2 + .01
   IF ((x1 + x2 + x3).GE. 3.0) GO TO 10
```

where the Z's are randomly generated integers and "INTGR" is a modular arithmetic statement function as previously defined. The active constraint is easily handled by the conditional "GO TO". Obviously, the exact solution cannot be found since 4/3, 7/9, and 4/9 cannot be explicitly expressed in two-place decimal numbers. For this problem, S and P were selected to be .85 and .01 respectively. This should reduce the experimental region by a factor of 100 with 85% probability. If $r_1 = 10$, then t_1 is approximately 2000 trials (R = 13,320,749).

Step one produced one cluster referred to as A, the best point having E = 0.122. The corresponding subregion defined by the cluster is

$$1.15 \leq x_1 \leq 1.52$$
$$0.62 \leq x_2 \leq 0.87$$
$$0.32 \leq x_3 \leq 0.52$$

which has $R_A = 23{,}166$ points. So the first step actually reduced the initial experimental region by a factor of 500 or more.

Step two was run with $t_2 = 1000$ trial points and $r_2 = 5$ retained points. This should also have a 100 to 1 reduction factor. E was reduced to 0.112 and one cluster, AA, was formed. The reduced subregion is

$$1.31 \leq x_1 \leq 1.37$$
$$0.76 \leq x_2 \leq 0.79$$
$$0.42 \leq x_3 \leq 0.46$$

which has $R_{AA} = 240$ points, a reduction of almost exactly 100.

The third step found the best solution possible with two decimal place numbers,

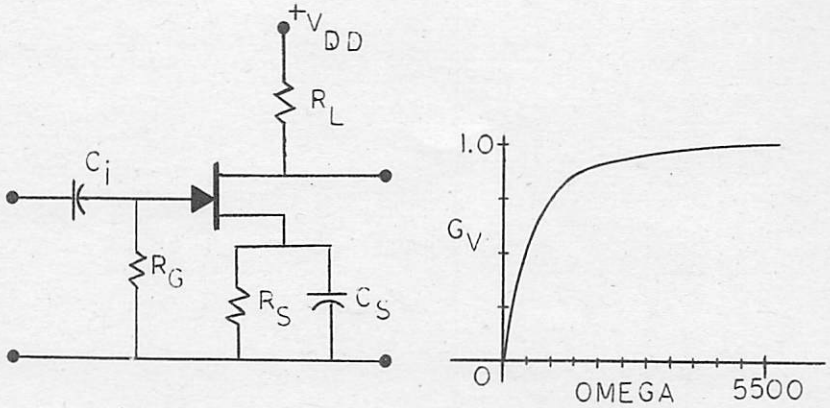
$$x_1 = 1.34$$
$$x_2 = 0.78$$
$$x_3 = 0.44 \quad .$$

E was reduced to .111199 as compared to Beale's 1/9 = .111111---. Example #2 illustrates the method of applying constraints on the parameters and the rapid convergence to an absolute minimum in 3,500 trials. The proposed random search method with a mode-seeking program found the minimum using only .025% of the total number of possible points in R.

Example #3--Field Effect Transistor

This last example was chosen to illustrate the usefulness of the proposed optimization method to a network designer. Consider matching the voltage gain of a Field Effect Transistor amplifier with a desired set of voltage gain points measured at a set of frequencies. Such a problem could be handled with a least squares minimization error function. Figure 2.a shows the circuit of the Field Effect Transistor (FET) and Figure 2.5 shows the desired voltage gain and the twelve sampling radian frequencies. The four parameters to be optimized are: C_i, R_g, R_s, and C_s. Two elements g_{fs} and R_l were assumed to be constant with values of .001 mho and .1Meg ohm, respectively.

The proposed random search method has a network analysis program incorporated as a sub-program. NASAP was chosen as the network analysis program since it solves for a network's transfer function symbolically. The transfer function may be one of many, for instance a voltage gain, driving point admittance or a transfer impedance. Once NASAP finds the transfer function it is available for further use and repeated calculations without destroying the function.



(a) FET Amplifier (b) FET Voltage Gain

Figure 2

Referring to the previously defined notation, F denotes the set of twelve desired voltage gains at the sampled radian frequencies. F' is the voltage gain function from NASAP calculated at $s=jw_i, i=1,---,12$. E is then the least squares error

$$E = \sum_{i=1}^{12} (F_i - F'(X_j, W_i))^2 \quad j=1,2,3,4.$$

where

$$F' = \frac{(R_g R_s R_1 g_{fs} C_i C_s)s^2 + (R_g R_1 g_{fs} C_i)s}{(R_g R_s C_i C_s)s^2 + (R_g C_i + R_s C_s + R_g R_s g_{fs} C_i)s + R_s g_{fs} + 1}$$

The random parameter generating statements were written to produce random integers, 9 to a decade, for the following ranges:

$$10^{-9} \leq C_i < 10^{-6}$$
$$10^{6} \leq R_g < 10^{9}$$
$$10^{3} \leq R_s < 10^{6}$$
$$10^{-6} \leq C_s < 10^{-3}$$

Each dimension in four-space has 30 points and as a result R = 810,000.

The sets of retained points r_1, r_2, and r_3 were set at 25, 15, and 15 points respectively. The number of trials t_1 was set at 2500 for the first step which should have a reduction factor of at least 30 or 40. The first step resulted in three clusters, denoted by A, B, and C. Most of the points in r_1 fell in the best cluster, A. Each of the 25 retained points in r_1 had C_s equal to $1. \times 10^{-6}$, so in all further steps C_s was considered to be optimized at that value. The sub-regions $R_A + R_B + R_C$ totaled 7,000 points, a reduction factor of over 100.

Step two examined $t_2 = 500$ trials for each of the two best clusters A and B produced by step one. The best cluster of step one produced very low error points while the second best cluster has errors five orders of magnitude higher than the best cluster. Therefore, only the A cluster was considered significant. The second step produced two clusters, AA and AB, from the A cluster of step one. Furthermore, R_s was equal to $1.\times10^4$ for all of the points in r_2; so once again, this is considered to be the optimum value.

The third step was preprogrammed to examine only the best cluster produced by the second step, in this case AA. This step found two clusters, AAA and AAB, both of which had very low error values. The AAA cluster had $E=.18\times10^{-7}$, $C_i=1.\times10^{-8}$, and $R_g=1.\times10^7$. The AAB cluster had $E=.21\times10^{-7}$, $C_i=5.\times10^{-9}$, and $R_g=2.\times10^7$. Both sets of points are certainly valid

optimum solutions and the resulting plot of voltage gain matched the desired curve in Figure 2.b.

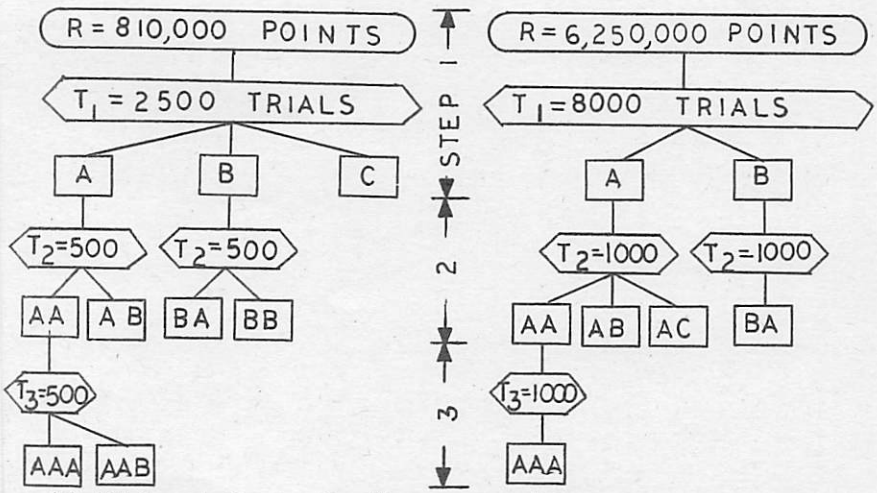
This problem is typical of multi-modal problems in general. The error surface is very irregular and modes often exist within larger modes. Essentially, one mode may appear from a large perspective; however, several modes may appear within the larger mode from a smaller, restricted viewpoint. Figure 3.a summarizes this example with a search chart showing the steps, clusters, and results. Note that the minimums were found using less than .4% of the total number of possible points in R.

This same problem was rerun using a larger initial experimental region to search for minimums other than the two previously found. The expanded ranges on the parameters are:

$$10^{-11} \leq C_i < 10^{-6}$$
$$10^{6} \leq R_g^i < 10^{11}$$
$$10^{3} \leq R_s^g < 10^{8}$$
$$10^{-8} \leq C_s^s < 10^{-3}$$

which have a total of R=6,250,000 possible points. Figure 3.b shows the details of steps one, two and three. Both the original run and the rerun are very similar up to the output of the third step.

The third step produced one cluster which had four different minimum points: $(R_g,C_i)=(.5x10^{7},.2x10^{-7}),(.5x10^{8},.2x10^{-8}),(.1x10^{8},.1x10^{-7})$, and $(.2x10^{9},.5x10^{-9})$. Each minimum has a global minimum in its own right; however, upon closer examination the implication is clear. The R_gC_i product is .1 for all minimums and therefore the global minimum occurs whenever R_g and C_i satisfy this condition. The optimization program provided more information about the circuit than expected.



(a) FET Search Chart - Run #1 (b) FET Search Chart - Run #2

Figure 3

IV. SUMMARY

The three previous examples have shown some of the advantages of the proposed random search optimization method using a mode-seeking program. First and foremost is its ability to locate global minimums. A global minimum is never guaranteed with any random method; however, the proposed random method increases the probability of finding the global minimum. In any case the minimum found by this method is usually better than local minimums

typically found by gradient methods. Secondly, the proposed method can handle parameter constraints very easily as Example #2 illustrated. In network problems, the optimum solution can be forced to be commercial on-the-shelf components. Thirdly, only error functions need to be calculated, as opposed to error and derivative functions for gradient methods. The proposed method requires many function evaluations, to be sure, but the amount of computer time and success of a solution is the real measure of a program. Fourthly, the proposed method can be programmed to handle non-linear networks, and networks in the time domain, although the examples chosen here do not illustrate this phase. Finally, the proposed method could find a small interval containing the minimum, then a gradient method could find the exact minimum.

The proposed optimization method, as with any optimization method, works better with practical experience. Someone who has a feel for the problem under attack can estimate the number of clusters to expect, number of random trials, and number of steps and, consequently, efficiently utilize the method. The optimization method would ideally be suited for a computer with a conversational mode.

REFERENCES

1. Fletcher, R. and Powell, M.J.D., "A Rapidly Convergent Descent Method for Minimization", The Computer Journal, 6, 163-168, July 1963.
2. Rosenbrock, H.H., "An Automatic Method for Finding the Greatest or Least Value of a Function", The Computer Journal, 3, 175-184, October 1960.
3. Hooke, R. and Jeeves, T.A., "Direct Search Solution of Numerical and Statistical Problems", Journal of the Association for Computing Machinery, 8, 212-229, April 1969.
4. Nelder, J.A. and Mead, R., "A Simplex Method for Function Minimization", The Computer Journal, 7, 308-313, January 1965.
5. Brooks, S.H., "A Comparison of Maximum Seeking Methods", Operations Research, 7, 430-457, July-August 1959.
6. Halton, J.H., "A Retrospective and Prospective Survey of the Monte Carlo Method," SIAM Review, 12, 1-63, January 1970.
7. Brooks, S.H., "A Discussion of Random Methods for Seeking Maxima", Operations Research, 6, 244-251, March 1958.
8. Calahan, D.A. and Temes, G.C., "Computer-Aided Network Optimization, the State of the Art", Proceedings of the IEEE, 55, 1832-1863, November 1967.
9. Chambers, R.P., "Random-Number Generation on Digital Computers", IEEE Spectrum, 4, 48-56, February 1967.
10. Gruenberger, F., "The Test-Passing Method of Random Digit Selection", Software Age, 4, 15-16, June 1970.
11. Bryan, J.K., Dwyer, S.J.III, & Lago, G.V., "Non-Parametric Decision Schemes for EEG Classification, Sixth Annual Rocky Mountain BioEngineering Symposium Conference Record, 56-61, University of Wyoming, Laramie, Wyoming, May 1969.
12. Beale, E.M.L., "Numerical Methods", Nonlinear Programming, ed. J. Abadie North Holland Publishing Company, Amsterdam, 1967, p. 150.

This work is part of a Ph.D. Thesis to be submitted by R. W. Becker.