Computer Program Descriptions

The Razor Search Program

PURPOSE: The razor search program is a package of subroutines which locates the minimum of a function of several variables by the razor search method of Bandler and Macdonald.

LANGUAGE: FORTRAN IV, 215 lines.

AUTHORS: J. W. Bandler, Department of Electrical Engineering, McMaster University, Hamilton, Ont., Canada, and P. A. Macdonald, formerly with Numerical Applications Group, Department of Electrical Engineering, University of Manitoba, Winnipeg, Man., Canada.

AVAILABILITY: ASIS–NAPS Document No. NAPS-01389. Listing also available from J. W. Bandler at $5.00

DESCRIPTION: The razor search method of Bandler and Macdonald [1] is a direct search method for locating the minimum of a function of several variables. It was developed to optimize network or system responses automatically in a minimax sense. Full details of the method, including descriptive and mathematical flow diagrams and a discussion of computational experience, have already been published [1]. As far as possible the variables in the program correspond to those used in that paper.

The user may call the package from his own program as follows. CALL RAZOR (PHO, UPHO, EPSMIN, RHO, ETA, KAPPA, U, FINISH, CONV, PRINT, UNIT). The variables in the argument list are given in the succeeding section.

Input Variables

PHO — An array specifying the initial approximation to the location of the optimum, i.e., the initial parameter values.

EPSMIN — The minimum permissible exploratory increment.

RHO — A scale factor for randomization.

ETA — A scale factor for the minimum exploratory increment.

KAPPA — The maximum number of random moves to be permitted.

U — The name of a user-supplied function subprogram which controls the evaluation of the objective function. This function name must appear in an external statement in the calling program. (See under Required Subprograms below.)

FINISH — The name of a user-supplied logical function subprogram which tests a finish criterion for terminating optimization. This function name must appear in an external statement in the calling program. (See under Required Subprograms below.)

PRINT — A logical variable: when .TRUE. instructs the computer to print out intermediate and final results, including a breakdown of the number of function evaluations at critical stages during optimization; when .FALSE. instructs the computer not to print out any results.

UNIT — An integer specifying the data set reference number of the output unit.

Output Variables

PHO — An array specifying the final approximation to the location of the optimum, i.e., the final parameter values.

UPHO — Value of the objective function at PHO.

CONV — A logical variable, initially .TRUE., becoming .FALSE. unless the finish criterion FINISH is satisfied.

It was found convenient to place the following variables in COMMON/RZR/.

K — An integer specifying the number of variable parameters.

DELTA — The next exploratory increment.

ALPHA — A scale factor for the reduction of DELTA.

LIMIT — The maximum number of function evaluations to be permitted.

Required Subprograms

U(·) — This user-supplied function subprogram for evaluating the objective function for given values of the parameters will, for network optimization, usually call an analysis program, the results of which will yield the objective function value. In [1], U was part of the ripple search package of programs which was used to determine efficiently the maximum deviation of the response from the desired response. The argument of U is an array of parameters.

FINISH (·) — This user-supplied logical function subprogram can be used to terminate optimization. Possible criteria, apart from number of function evaluations which are handled by LIMIT, may include a test to determine whether the maximum deviation has fallen below a specified value, or to determine whether any significant improvement has occurred since the last one or more times the finish criterion was tested, and so on. The argument of FINISH is an array of parameters.

R(1) — A random number generator (values between —1 and 1) for the random moves.

Comments

As it stands the package will handle up to 15 variable parameters. It was designed to handle ill-conditioned problems, particularly problems in which the objective function has discontinuous derivatives. Because of the difficulty inherent in trying to optimize such functions, the choice of values for the input parameters and scale factors may be critical to efficiency. Their effect upon the optimization procedure should be well understood before attempting to use this program.

The program was run and tested on an IBM 360/65.

Discussion

Parameter and other constraints can be handled in a variety of ways as explained by Bandler [2], [3]. A particularly involved class of constrained optimization problems, the design of inhomogeneous waveguide transformers [3], has been solved by the razor search method.

Acknowledgment

The cooperation of Dr. A. Wexler, Department of Electrical Engineering, University of Manitoba, Winnipeg, Man., Canada, where this work was carried out, is gratefully acknowledged.

References

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

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

[3] ——, "Computer optimization of inhomogeneous waveguide transformers," *IEEE Trans. Microwave Theory Tech.*, vol. MTT-17, Aug. 1969, pp. 563–571.

Manuscript received February 3, 1971; revised March 17, 1971. This work was supported by the Faculty of Graduate Studies of the University of Manitoba and by the National Research Council of Canada.

For program listing, order document NAPS-01389 from ASIS National Auxiliary Publications Service, c/o CCM Information Corporation, 909 Third Avenue, New York, N. Y. 10022; remitting $2.00 per microfiche or $5.00 per photocopy.