Computer Program Descriptions

Program for Investigating Minimax Optimality Conditions

PURPOSE: This program is a package of subprograms which investigates the optimality of a design or a proposed solution to an approximation problem in the minimax sense.

LANGUAGE: Fortran Version 2.3 and Scope Version 3.4 for the CDC 6400 computer; 721 cards, including comments.

AUTHORS: J. W. Bandler and T. V. Srinivasan, Department of Electrical Engineering, McMaster University, Hamilton, Ont., Canada.

AVAILABILITY: ASIS/NAPS Document No. NAPS-02096. Listing also available from J. W. Bandler at $15.00.

DESCRIPTION: The program is designed to test a solution for the necessary conditions for a minimax optimum by two different formulations. One uses a linear programming approach, and the other the solution of a set of linear independent equations.

Conditions for a Minimax Optimum [1]

Let $\hat{y}_l(\phi)$, $l=1, \cdots, n_r$ be n_r discrete maxima in descending magnitude at a given point ϕ in a k-parameter space. If $\hat{y}_l(\phi)$, $l=1, \cdots, k_r$ are taken as equal, then for ϕ to satisfy the necessary conditions for a minimax optimum, there exist $u_l \geq 0$ for $l=1, \cdots, k_r$ such that

$$\sum_{l=1}^{kr} u_l \nabla \hat{y}_l(\phi) = 0 \tag{1}$$

$$\sum_{l=1}^{kr} u_l = 1. \tag{2}$$

When testing the conditions for optimality at a point ϕ, we attempt to solve (1) and (2) for $k_r=1, 2, \cdots$ until for a value of $k_r^*(\leq n_r)$ (1) and (2) are satisfied. If this is not possible then the necessary conditions are not satisfied.

Method 1

Equations (1) and (2) are solved here by minimizing $u_{k_r+1} \geq 0$ such that (2) is satisfied and

$$\left| \sum_{l=1}^{kr} u_l \frac{\partial \hat{y}_l}{\partial \phi_i} \right| \leq u_{kr+1}, \qquad i = 1, 2, \cdots, k. \tag{3}$$

Linear programming ensures that $u_l \geq 0$ for $l=1, 2, \cdots, k_{r+1}$.

Manuscript received November 8, 1972; revised February 9, 1973. This work was supported by the National Research Council of Canada under Grants A7239 and C154.

See NAPS document No. 02096 for 23 pages of supplementary material. Order from ASIS/NAPS c/o Microfiche Publications, 305 E. 46th St., New York, N. Y. 10017. Remit in advance for each NAPS accession number $1.50 for microfiche or $5.00 for photocopies up to 30 pages, 15¢ for each additional page. Make checks payable to Microfiche Publications.

Method 2

Here, we solve a set of linearly independent equations

$$\sum_{l=1}^{k_r} u_l \frac{\partial \hat{y}_l}{\partial \phi_i}(\phi) = 0 \qquad i \in J \tag{4}$$

and (2), where J is a suitable subset of $\{1, 2, \cdots, k\}$.

There is no guarantee, however, that $u_l \geq 0$ for $l=1, \cdots, k_r$. When k_r-1 is greater than the number of elements of J, the system of (2) and (4) has more unknowns than equations, and we use Method 1 to get the u_l.

The program package can be called from the user's main program and either of the two, or both of the methods, can be used to test the optimality conditions. The user can either specify the value of k_r or a tolerance relative to $\hat{y}_1$ within which some of the $\hat{y}_2, \cdots, \hat{y}_{n_r}$ lie. The necessary conditions for optimality are satisfied when the norm $\|r\|$ of the residual vector

$$r = \sum_{l=1}^{m_r} u_l \nabla \hat{y}_l$$

for $m_r=1, \cdots, k_r$ falls within a user-specified value ϵ for $u_l \geq 0$ and (2) is satisfied, in which case a logical variable OPTIM is set·TRUE·. The output list consists of the variables m_r, $u_l (l=1, \cdots, m_r)$, r, $\|r\|$ and OPTIM.

Required Subprograms

The user has to have a subprogram by which the discrete values of n_r functions $\hat{y}_l$ (arranged in descending magnitude) and their derivatives $(\nabla \hat{y}^T)^T$ with respect to the parameters $\phi_1, \phi_k, \cdots, \phi_k$ are explicitly available. The package uses the following subroutines, the listings of which are available as indicated in [2]–[6].

ARRAY converts data array from single to double dimension or vice versa [2] while MINV inverts a matrix and calculates its determinant [3]. MFGR determines the rank and linearly independent rows and columns of a given matrix [4]. SIMPLE is a linear-program solving subroutine [5], [6], and SOLVE solves a set of linear simultaneous equations [7].

Comments

The package has been programmed to handle any number of variable parameters and discrete maxima. This program was run and tested on a CDC 6400 computer. The package requires roughly 40 000 octal units of computer memory for $k=15$ and $\hat{n}_r=15$.

References

[1] J. W. Bandler, "Conditions for a minimax optimum," *IEEE Trans. Circuit Theory* (Corresp.), vol. CT-18, pp. 476–479, July 1971.
[2] Subroutine ARRAY, p. 98, System/360 Scientific Subroutine Package, Version III, in *IBM Programmer's Manual* Number 360A-CM-03x.
[3] Subroutine MINV, p. 118, System/360 Scientific Subroutine Package, Version III, in *IBM Programmer's Manual* Number 360A-CM-03x.
[4] Subroutine MFGR, p. 127, System/360 Scientific Subroutine Package, Version III, in *IBM Programmer's Manual* Number 360A-CM-03x.
[5] Subroutine SIMPLE, Data Processing and Computing Center, Library Information Sheet MILIS 15.02.01, p. 130, McMaster University, Hamilton, Ont., Canada.
[6] J. W. Bandler and T. V. Srinivasan, "The grazor search program for minimax objectives," *IEEE Trans. Microwave Theory Tech.* (Program Description), vol. MTT-20, pp. 784–785, Nov. 1972.
[7] Subroutine SOLVE, Data Processing and Computing Center, Library Information Sheet MILIS 45.04.16, p. 27, McMaster University, Hamilton, Ont., Canada.