AAEC/E 85

UNCLASSIFIED

AAEC/E 85

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

AUTOMATIC SOLUTION OF OPTIMUM DESIGN PROBLEMS ON A DIGITAL COMPUTER

by

B. R. LAWRENCE

Issued Sydney, March 1962

UNCLASSIFIED

AUSTRALIAN ATOMIC ENERGY COMMISSION

RESEARCH ESTABLISHMENT

LUCAS HEIGHTS

AUTOMATIC SOLUTION OF OPTIMUM DESIGN PROBLEMS ON A DIGITAL COMPUTER

by

B. R. LAWRENCE

ABSTRACT

A description is given of a method suitable for the automatic solution of certain optimum design problems on a digital computer for cases where the number of constraints imposed on the design is not greater than the number of design variables. The problem is transformed to one requiring the minimisation or maximisation of an unconstrained function, for which a gradient method is used.

CONTENTS

INTRODUCTION

Digital computers are currently being used extensively to relieve the design engineer of much of the tedious computation associated with conventional methods of engineering design. There are however distinct advantages to be gained by using direct methods for the solution of optimum design problems to which some attention has been given in recent years (Dennis, Nease, Saunders 1959; Brown 1959; Dickinson 1958). The method described herein is a direct method, suitable for the automatic solution on a digital computer of certain optimum design problems where the number of constraints imposed on the design is not greater than the number of design variables. It is hoped that the method will find use in reactor core design and reactor system optimisation studies as well as other engineering design problems.

In the type of optimum design problem with which this report is concerned, values of the design variables must be found which make some criterion function a maximum or minimum and simultaneously satisfy constraints imposed by the design specifications or physical limitations of the system. The conventional approach is the trial – and – error process where guesses are made of the values of the design variables, performance parameters are computed, and the performance is compared with the specifications. If the design does not satisfy the specifications the design variables are changed and the calculation repeated until finally an acceptable design, one which meets the specifications, is produced. For an optimum design where the system has to be optimised with respect to some criterion, the above procedure is repeated until a series of acceptable designs is produced and the best of these is selected as the optimum design. The conventional method suffers from the disadvantage that it relies heavily on the insight and experience of the designer in the choice of changes to make to the design variables. It is not readily applicable to the optimum design of new or different systems with which the designer has had little or no experience, and the successful solution of large problems is severely handicapped by the fact that human insight becomes less dependable as the number of variables increases.

A direct method for the solution of the optimum design problem is one which relies on a mathematical technique rather than an intuitive one to find the values of the design variables corresponding to a minimum or maximum of the criterion function in a region where the constraints are satisfied. When a direct method is programmed for a digital computer it becomes automatic and no intervention or decision making is required of the designer once the program has been initiated. There is no difficulty in applying the technique to entirely different systems, and the resulting values of the design variables are known to correspond to the minimum (or maximum) of the criterion function and to satisfy the constraints. In addition, the number of design variables which can be handled is limited only by the capacity of the computer.

THEORY

The method deals with optimum design problems which may be formulated in the following way:

minimise
$$G = g(x_1, x_2, \ldots, x_d) \tag{1}$$

subject to the constraints

$$g_j(x_1, x_2, \ldots, x_d) \leq C_j \quad , \quad j = 1, 2, \ldots, n \tag{2}$$

where $n \leq d$ and the constraints are independent of one another. The functions g and g_j are assumed to be continuous with continuous derivatives and may be non linear. Although the problem is formulated as one of minimisation subject to constraints of the type given by (2), it should be noted that maximising instead of minimising, and replacement of $\leq$ by $\geq$ in the inequalities, present no further difficulties.

$x_1, x_2, \ldots, x_d$ are the physical variables of the system which the designer may vary. The constraints either specify performance or ensure that the final design is physically realisable. The functions g_j are design formulae giving performance parameters in terms of the design variables and the constants C_j are fixed by the specifications. The minimisation (or maximisation) of the criterion function $G = g(x_1, x_2, \ldots, x_d)$ corresponds to stating that the design should be optimum with respect to some criterion e.g. cost, weight, volume, efficiency.

The inequalities are firstly converted to equalities by the introduction of slack variables $z_1, z_2, \ldots, z_n$ which are squared so that they may assume any value, positive or negative, without violating (2).

Hence

$$g_j(x_1, x_2, \ldots, x_d) + z_j^2 = C_j \quad . \qquad (3)$$

$$j = 1, 2, \ldots, n$$

There are now d design variables and n slack variables, d + n variables in all. They are, however, not all independent. By means of (3), n of the d + n variables may be expressed in terms of the remaining d independent variables which are always made the n slack variables and d − n design variables. Denoting the n dependent variables by $v_1, v_2, \ldots, v_n$ and the d independent variables by $w_1, w_2, \ldots, w_d$ the problem, in general terms, becomes one of minimising

$$G = f(v_1, \ldots, v_n, w_1, \ldots, w_d) \, , \qquad (4)$$

subject to

$$f_j(v_1, \ldots, v_n, w_1, \ldots, w_d) = C_j \quad , \qquad (5)$$

$$j = 1, 2, \ldots, n$$

where the $v_1, v_2, \ldots, v_n$ are functions of the $w_1, w_2, \ldots, w_d$. Although the $v_1, v_2, \ldots, v_n$ cannot in general be obtained explicitly in terms of the $w_1, w_2, \ldots, w_d$, the partial derivatives $\frac{\partial G}{\partial w_1}, \frac{\partial G}{\partial w_2}, \ldots, \frac{\partial G}{\partial w_d}$ can be evaluated and a gradient technique used to minimise $G = h(w_1, w_2, \ldots, w_d)$.

From (4)

$$\frac{\partial G}{\partial w_i} = \frac{\partial f}{\partial w_i} + \frac{\partial f}{\partial v_j} \frac{\partial v_j}{\partial w_i} \quad . \qquad (6)$$

$$i = 1, 2, \ldots, d$$

$$j = 1, 2, \ldots, n$$

From (5)

$$\frac{\partial f_j}{\partial w_i} = - \frac{\partial f_j}{\partial v_k} \frac{\partial v_k}{\partial w_i} \quad . \qquad (7)$$

$$i = 1, 2, \ldots, d$$

$$j = 1, 2, \ldots, n$$

$$k = 1, 2, \ldots, n$$

In matrix notation (6) and (7) become

$$G_w = V_w F_v + F_w \qquad (8)$$

and

$$\bar{F}_w = - V_w \bar{F}_v \quad , \qquad (9)$$

where G_w is the d x 1 column matrix with elements $\frac{\partial G}{\partial w_i}$

F_w is the d x 1 column matrix with elements $\frac{\partial f}{\partial w_i}$

F_v is the n x 1 column matrix with elements $\frac{\partial f}{\partial v_j}$

V_w is the d x n matrix with elements $\frac{\partial v_j}{\partial w_i}$

$\bar{F}_w$ is the d x n matrix with elements $\frac{\partial f_j}{\partial w_i}$

$\bar{F}_v$ is the n x n matrix with elements $\frac{\partial f_j}{\partial v_j}$

Hence, from (8) and (9),

$$G_w = F_w - \bar{F}_w \quad \bar{F}_v^{-1} \quad F_v \quad . \tag{10}$$

From this expression the components $\frac{\partial G}{\partial w_1}, \frac{\partial G}{\partial w_2}, \ldots, \frac{\partial G}{\partial w_d}$ may be evaluated from a knowledge of the various partial derivatives of f and f_j. It is proposed to focus attention on cases where the partial derivatives of f and f_j are obtainable analytically so that the elements of F_v, $\bar{F}_v$, F_w, $\bar{F}_w$ may be written in terms of $w_1, w_2, \ldots, w_d$ and $v_1, v_2, \ldots, v_n$. Where derivatives are not possible analytically, provision can be made for their numerical evaluation and (10) used to evaluate the required components $\frac{\partial G}{\partial w_1}, \frac{\partial G}{\partial w_2}, \ldots, \frac{\partial G}{\partial w_d}$ as before.

It should be noted that when q of the n constraints (2) are equalities there are $d + n - q$ variables in all, n dependent variables, and $d - q$ independent variables of which $n - q$ are slack.

Since the method accommodates a maximum of only d constraints it is impossible to include all the d conditions that all design variables be positive:

$$x_i \geq 0 \quad , \quad i = 1, 2, \ldots, d \tag{11}$$

as well as other types of constraints. It is advisable however to include as many of these conditions as possible to make the total number of constraints in any problem the maximum of d. A preliminary computer search of the unconstrained minimum of $g(x_1, x_2, \ldots, x_d)$ may indicate the design variables most likely to violate (11).

It is essential to be able to evaluate G when required, both because it is necessary to know the actual minimum cost, maximum efficiency etc. of the resulting design, and because it is the only way of distinguishing the absolute minimum from relative minima. Also, values of G are required at each step in the proposed minimisation procedure. Since both the evaluation of G and the components $\frac{\partial G}{\partial w_1}, \frac{\partial G}{\partial w_2}, \ldots, \frac{\partial G}{\partial w_d}$ require the values of $v_1, v_2, \ldots, v_n$ corresponding to those of $w_1, w_2, \ldots, w_d$, a suitable method for the simultaneous solution of the constraint equations (5) is required. The Newton-Raphson method

$$V^{(p+1)} = V^{(p)} - (\bar{F}_v^{-1})^{T(p)} \, Y^{(p)} \quad , \tag{12}$$

where V is the $n \times 1$ column matrix with elements v_j

and Y is the $n \times 1$ column matrix with elements $f_j - C_j$,

has the advantage that it requires no more information than is already available. If an acceptable design does not exist, (12) will not yield a solution and provision must be made for the iterative process to be halted after a sufficiently large number of cycles.

During minimisation where the values of $w_1, w_2, \ldots, w_d$ are changing, an estimate of the corresponding changes in $v_1, v_2, \ldots, v_n$ may be obtained from

$$\frac{\partial f_j}{\partial w_i} \delta w_i \approx - \frac{\partial f_j}{\partial v_k} \delta v_k \tag{13}$$

Hence

$$\Delta V \approx - (\bar{F}_v^{-1})^T \, \bar{F}_w^T \, \Delta W \quad , \tag{14}$$

where ΔV is the $n \times 1$ column matrix with elements δv_j

and ΔW is the $d \times 1$ column matrix with elements δw_i .

By making the change ΔV given by (14) to V before applying (12), only a few Newton-Raphson iterations will be required to obtain the V corresponding to the new W, W being the $d \times 1$ column matrix with elements w_i .

MINIMISATION OF G

A gradient method is used to minimise G. Starting with an initial guess $w_1^{(o)}, w_2^{(o)}, \ldots, w_d^{(o)}$ at an optimum design, a sequence of values of $w_1, w_2, \ldots, w_d$ is constructed using

$$W^{(p+1)} = W^{(p)} - \lambda G_w^{(p)} \tag{15}$$

until a minimum is reached. λ is chosen such that the step length

$$s = \lambda \left[G_w^{(p)T} \, G_w^{(p)} \right]^{1/2}$$

is halved if the step is large enough to cause an increase in G rather than a decrease (for the minimisation case). The procedure corresponds to taking steps in directions perpendicular to contour lines of constant $G = h(w_1, w_2, \ldots, w_d)$. It is clear that the work required to find a minimum is greatly reduced by a good initial guess at the optimum point, although this is in no way essential for a correct solution.

A minimum is reached when

$$\left| \frac{\partial G}{\partial w_i} \right| < \epsilon \quad \text{for all } i, \text{ where } \epsilon \text{ is a small positive constant.}$$

The case where the function $h(w_1, w_2 \ldots w_d)$ has several minima may arise. It is essential therefore that several starting points be chosen and the problem re-run in each case to check for the same result. If there are several minima the absolute minimum representing the solution to the problem is found from a knowledge of values of the function at each of the individual minima. It is unlikely that the gradient method will come to rest at a saddle point, if there is one. If it does, however, the results of other runs will yield a lower value of G. The physical aspects of the problem will often suggest the behaviour of the criterion function.

When searching for maxima instead of minima, (15) becomes

$$W^{(p+1)} = W^{(p)} + \lambda G_w^{(p)} .$$

COMPUTER LOGIC DIAGRAM

A logic diagram for a computer program based on the method is shown in Figure 1. Starting with an initial guess $W^{(o)}$ and $V^{(o)}$ of W and V and a step length s, the correct V corresponding to the initial W is found by the Newton – Raphson method. At each iteration N is advanced by unity to keep count of the number of iterations required for convergence to the specified accuracy ϵ_1. If an acceptable design does not exist N will become equal to some limiting value. In this case an identification can be punched and the program terminated. When the correct V corresponding to $W^{(o)}$ has been found, G is computed and G, V, and W are stored in G_o, V_o, and W_o respectively. The components G_w of the gradient at that point are computed and punched together with V, W, and G. If the absolute magnitudes of the components of the gradient are all less than some prescribed quantity ϵ_2 the point is a minimum and another starting point is chosen.

If the point is not a minimum, a change ΔW to W is computed such that the distance between the new point with coordinates $W - \Delta W$ and the old with coordinates W is s. The approximate value of ΔV required to adjust V is computed and the correct value of V corresponding to the new W is obtained from one or more Newton – Raphson iterations. The value G of the function at the new point is computed and compared with that at the old, G_o. If the step has been such that $G < G_o$ a new station is established and the procedure repeated. If not, the step length is halved, a different ΔW computed, and the step length halved again in preparation for the next step. This procedure is repeated until a new station is established. Finally a minimum is reached.

Parameters L and N_x keep a record of the number of times s has to be halved to establish a particular station and the total number of stations established respectively. Should the initial s be too large the program will automatically reduce it rapidly. Should it be too small a minimum will be reached but at the expense of some computer time. In selecting an initial s therefore the tendency should be to overestimate. Provision is made for stopping the program if N_x becomes excessively large.

EXAMPLE

To demonstrate the method simply without referring to any specific design problem it will be assumed that it is necessary to minimise

$$G = 12 - 6x_1 - 4x_2 + x_1^2 + 2x_2^2 \quad ,$$

subject to

$$x_1 \leq 2.5$$

$$x_1 x_2 + 2x_2 \leq 10 \quad .$$

The function $G = g(x_1, x_2)$ is shown in contour in Figure 2 and has a minimum value of unity at the point (3, 1). The region in which the solution must lie is shaded.

The constraints are written

$$x_1 + z_1^2 = 2.5$$

$$x_1 x_2 + 2x_2 + z_2^2 = 10 \quad .$$

Choose

$$v_1 = x_1$$

$$v_2 = x_2$$

$$w_1 = z_1$$

$$w_2 = z_2 \quad .$$

Then

$$\bar{F}_v = \begin{bmatrix} 1 & v_2 \\ o & 2 + v_1 \end{bmatrix}, \quad \bar{F}_w = \begin{bmatrix} 2w_1 & o \\ o & 2w_2 \end{bmatrix} ,$$

$$F_v = \begin{bmatrix} 2v_1 - 6 \\ 4v_2 - 4 \end{bmatrix} , \quad F_w = \begin{bmatrix} o \\ o \end{bmatrix} , \quad Y = \begin{bmatrix} v_1 + w_1^2 - 2.5 \\ v_1 v_2 + 2v_2 + w_2^2 - 10 \end{bmatrix} .$$

This problem was run on an IBM 650 computer for one set of initial values:

$$w_1^{(o)} = \sqrt{2} \qquad v_1^{(o)} = o$$

$$w_2^{(o)} = \sqrt{2.5} \qquad v_2^{(o)} = o \quad ,$$

and the path followed during solution is shown in Figure 2. A step length s = 2 was used and five station points are shown. Although it is not strictly correct to join the station points by straight lines when plotted on the x plane, this is done for convenience. The analytic solution to the problem is the point (2.5, 1) with G = 1.25. It took 2 Newton – Raphson cycles to find v_1 and v_2 corresponding to $w_1^{(o)} = \sqrt{2}$, $w_2^{(o)} = \sqrt{2.5}$ and thereafter a minimum of 1 and a maximum of 2 Newton – Raphson cycles to correct V after the adjustment (14) had been applied. The values chosen for the allowable errors ϵ_1 and ϵ_2 were

$$\epsilon_1 = \epsilon_2 = 0.002 \quad .$$

REFERENCES

Brown, R.R., (1959).- A generalised computer procedure for the design of optimum systems - Parts I and II. Communications and Electronics, No. 43: 285-293.

Dennis, J.B., Nease, R.F., Saunders, R.M., (1959).- System synthesis with the aid of Digital computers. Communications and Electronics, No. 45: 512-515.

Dickinson, J.R., (1958).- A computer program for system optimisation. Trans. Engng. Inst. Canada, 2(4): 157-161.

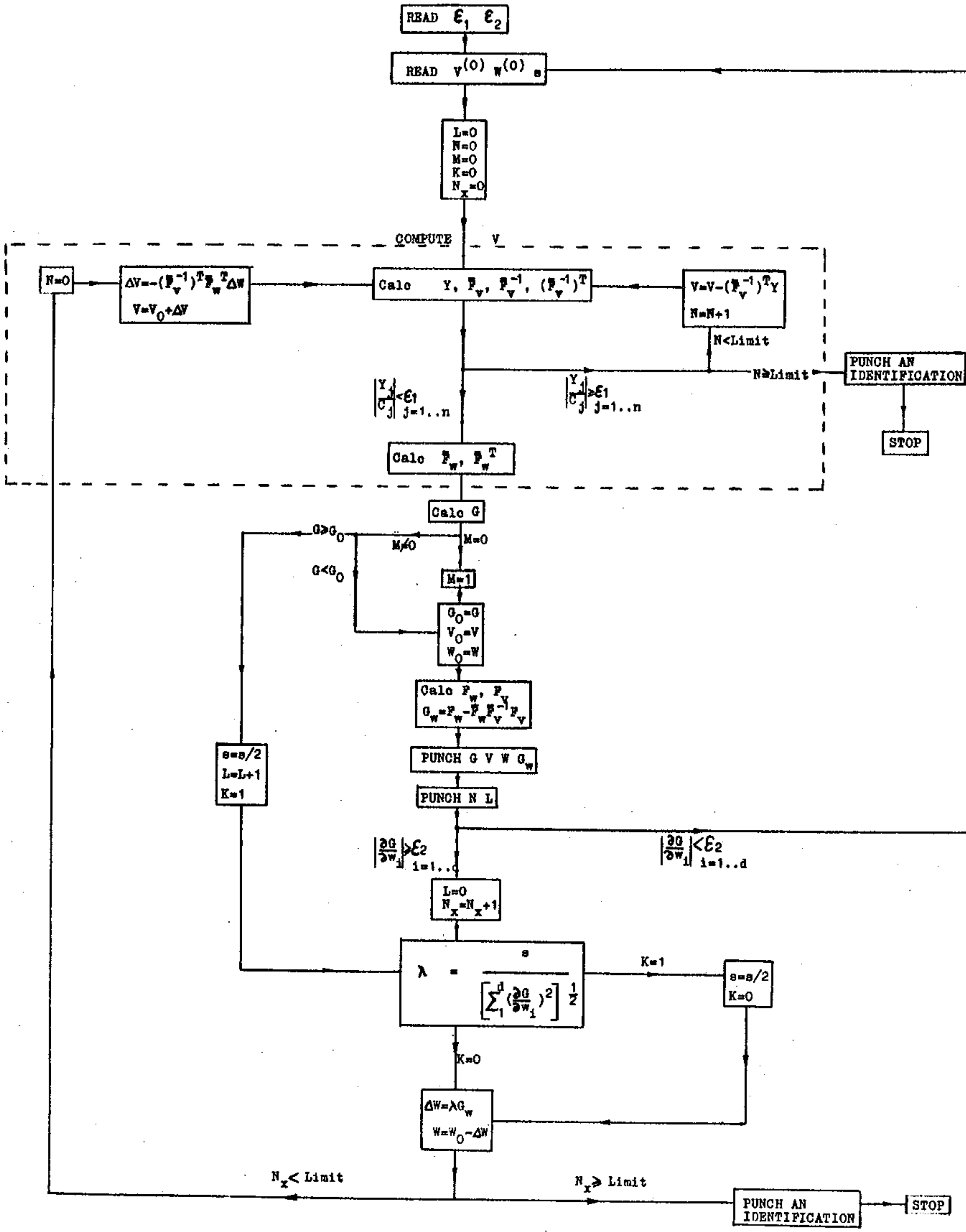

FIGURE 1. COMPUTER LOGIC DIAGRAM

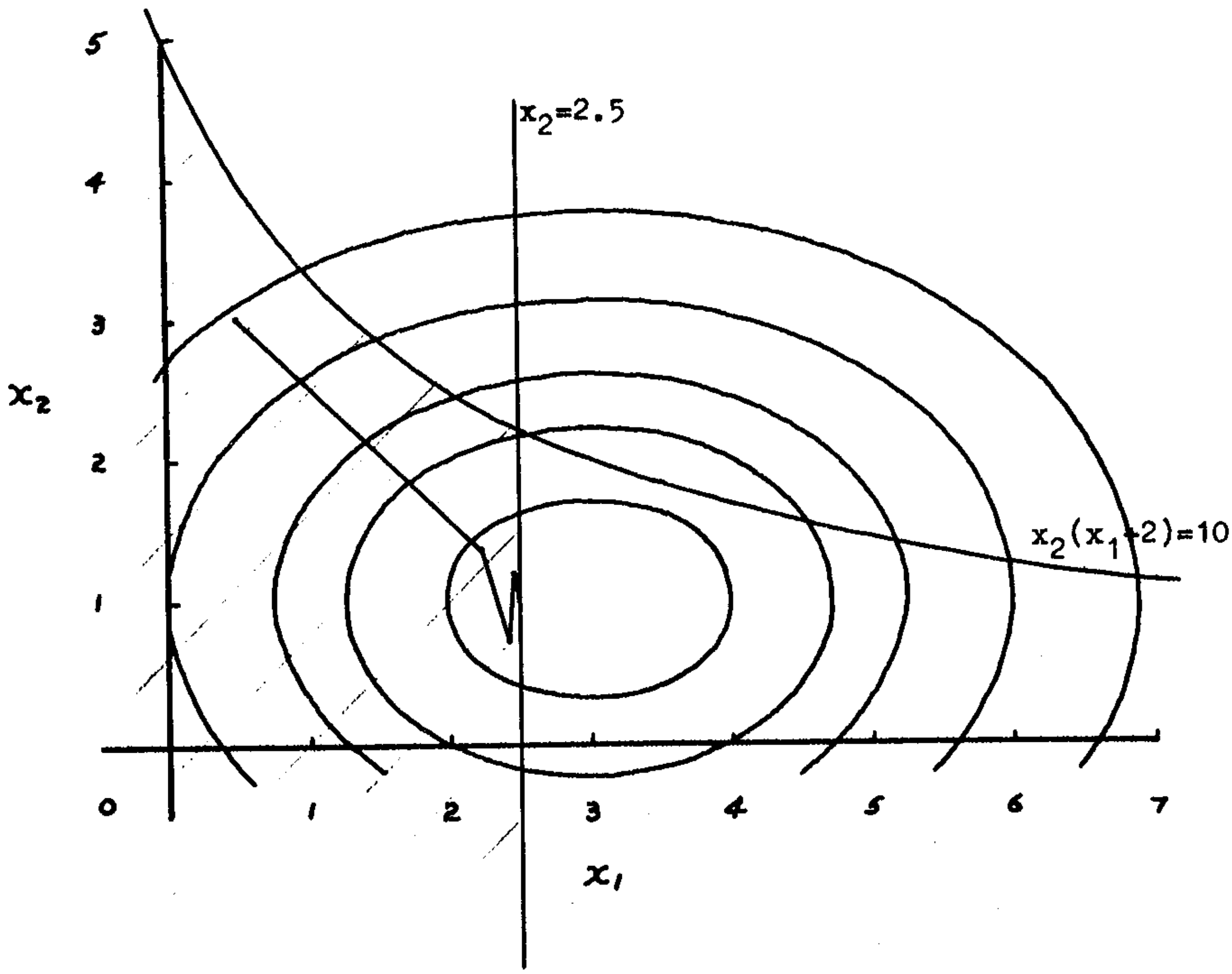

FIGURE 2. SOLUTION OF EXAMPLE PROBLEM