Optimization Problems: Solution by an Analogue Computer

By A. W. O. Firth

Optimization problems in general are discussed, including the particular field of linear programming, and the method of solving these problems (under constraints) rapidly and accurately on an analogue computer is explained. The method of solution is explained purely from an analytic point of view. A simple example is given to illustrate the process of optimization of a second order function.

1. Introduction

Optimization problems involve either the minimization or maximization of a given function of the n variables $x_1 \ldots x_n$, the minimization of a cost function or say the maximization of a profit function. This function designated $f(x_1, x_2, \ldots x_n)$ will be subject to certain conditions involving $x_1, x_2, \ldots x_n$. The expression $f(x_1, x_2, \ldots x_n)$ may, of course, contain powers of the x_j. Such problems are familiar in many industries and range from the optimization of the operation of a complete chemical plant and its distribution organization, to the optimization of an advertising campaign, with the corresponding modification to the function to be optimized $f(x_1, x_2 \ldots x_n)$.

Linear Programming is a particular example of an optimization problem in which $f(x_1, x_2, \ldots x_n)$ is a linear function of $x_1 \ldots x_n$. The best illustration of this type of problem is probably the optimization of transport costs; $f(x_1, x_2, \ldots x_n)$ is the cost function to be minimized and is a linear function of the variables $x_1 \ldots x_n$ which themselves represent the quantities of goods transported along the n different possible routes. $f(x_1 \ldots x_n) = c_1x_1 + c_2x_2 + \ldots c_nx_n$, where $c_1 \ldots c_n$ are the costs associated with the n routes. The restrictions on $x_1 \ldots x_n$ arise from the requirement and availability conditions for the transportation of the goods.

Previous papers which have discussed the solution of linear-programming problems by analogue computers have all approached the explanation of the method of solution by drawing analogies from particular dynamic systems (Korn and Korn, 1956), or from geometric n-spaces (Jackson, 1960; Pyne, 1956; Ablow and Brigham, 1955). So far, the explanation of the analogue computer solution has not been considered from an analytic starting point. This paper has been written to fill this gap and gives an analytic explanation of the method of solution of optimization problems on an analogue computer without the need for introducing analogies. It is considered that this approach gives a clearer understanding of the computer solution, and, indeed, permits the easy extension of the method to optimization of non-linear problems.

2. Problem

The general optimization problem (including linear programming and other particular examples) may be stated as follows.

A given function $f(x_1 \ldots x_n)$, single valued and continuous over the domain of interest, is to be optimized subject to the following restrictions:

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n &\geqslant a_{10} \\ a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n &\geqslant a_{20} \\ \ldots\ldots\ldots\ldots\ldots \\ a_{m1}x_1 + a_{m2}x_2 + \ldots + a_{mn}x_n &\geqslant a_{m0} \end{aligned}$$

It is sufficient for the purposes of this paper to consider the problem of minimizing $f(x_1, x_2, \ldots x_n)$, since maximizing is an exactly similar process. The restrictions may be grouped as

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geqslant a_{i0}$$
$$(\text{for } i = 1, 2, \ldots m).$$

Some of these restrictions may of course reduce to the simple condition $x_j \geqslant 0$.

3. Computer Solution

The necessary condition for $f(x_1, x_2 \ldots x_n)$ to assume its absolute minimum value is

$$\frac{\partial f}{\partial x_j} = 0 \quad (j = 1, 2, \ldots n).$$

This is not a sufficient condition, since the same condition defines other stationary points of the function. However, the introduction of sufficiency into the solution is a simple matter and is treated later.

The computer solution is based upon the property of an integrating amplifier (i.e. a high-gain amplifier with capacitance feedback—see Thomason, 1961) that when the input volts decrease to zero the output assumes a constant value. n integrating amplifiers are required, their outputs representing $x_1 \ldots x_n$, respectively.

Fed as inputs to the n integrating amplifiers are $\frac{\partial f}{\partial x_j}$ respectively (as in Fig. 1) so that, when $f(x_1 \ldots x_n)$ is driven to its minimum value, the $\frac{\partial f}{\partial x_j}$ have become zero. Consequently the integrating amplifier inputs $= 0$ and their outputs, now constant, represent the values of $x_1 \ldots x_n$ which minimize $f(x_1, x_2 \ldots x_n)$. This is a closed-loop arrangement as the amplifier outputs x_j are used continuously to compute the partial differentials $\frac{\partial f}{\partial x_j}$ which are fed as inputs to the integrators.

If the values x_j corresponding to the minimum solution are required, these may be read off directly as amplifier outputs to a digital printer; if the minimum value of $f(x_1, x_2, \ldots x_n)$ is required this is computed by other computer units from the x_j values and can also be recorded.

The sufficiency for the computer solution to be the absolute minimum of $f(x_1 \ldots x_n)$ is ensured by repeating the solution say 3 times, each time taking a different starting point for the x_j and verifying the minimum solution. Since, for a medium-sized problem, one solution takes only 2 seconds, the sufficiency check takes only 6 seconds, and so the complete solution is still obtained in a very short time. With the best analogue computers, which can resolve the fourth or fifth decimal digit at each component operation, the overall computing accuracy for medium-sized problems will be better than $\frac{1}{2}\%$. The sufficiency check on certain analogue computers may be programmed automatically, so that the correct minimum solution is selected by the computer. Solution time may be speeded up by feeding $\frac{k\partial f}{\partial x_j}$ to the input of the integrator shown in Fig. 1, where $k \geqslant 1$.

4. Constraints

The introduction of constraints into the minimization process is readily achieved by the use of discontinuous driving voltages $\gamma_i (i = 1 \ldots m)$ such that γ_i is a large positive voltage if the ith constraint

$$(a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n - a_{i0} \geqslant 0)$$

is not satisfied, and is zero if the ith constraint is satisfied. The driving voltage γ_i is applied to the appropriate x_j in a positive sense to increase the value of x_j and so readjust the minimization process to satisfy the ith constraint.

Diode feedback around the constraint amplifiers (high-gain type) provides either $\gamma_i = 0$ or $\gamma_i =$ very large positive voltage (see Fig. 2) since, in the "constraint not satisfied" condition, the amplifier is an open loop. From a condition which is not satisfied, the driving voltage γ_i must be applied to drive back each of the parameters $x_1 \ldots x_n$ which occur in the equation for the ith constraint so that the ith constraint is again satisfied. For each one of the parameters x_j, the restoring voltage γ_i must be factored before feeding to the appropriate x_j amplifier, according to the effect that the particular x_j has in generating the ith constraint; i.e. for x_k, γ_i is factored by a_{ik}.

It is more effective to feed the γ_i components into the x_j integrators (as $-\gamma_i$) rather than to x_j itself (see Fig. 3). This is desirable since the system has a quicker response by this method, and the process of minimization is not affected.

The simple condition $x_j \geqslant 0$ is ensured simply by diode feedback round the x_j integrator as shown in Fig. 3.

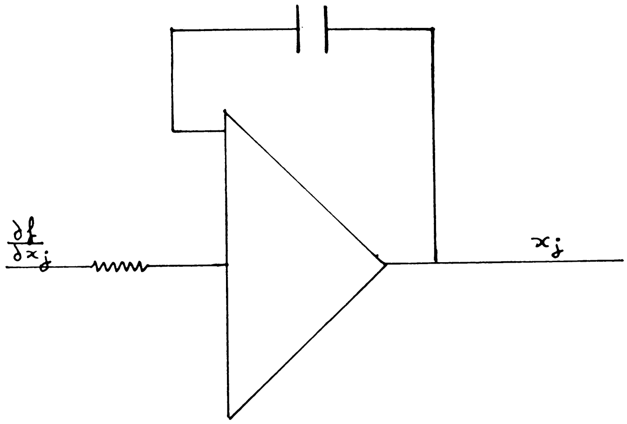

Fig. 1.—x_j integrator without constraints

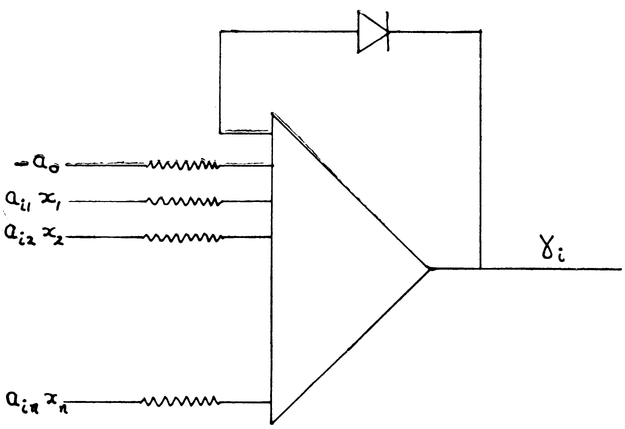

Fig. 2.—Driving voltage γ_i

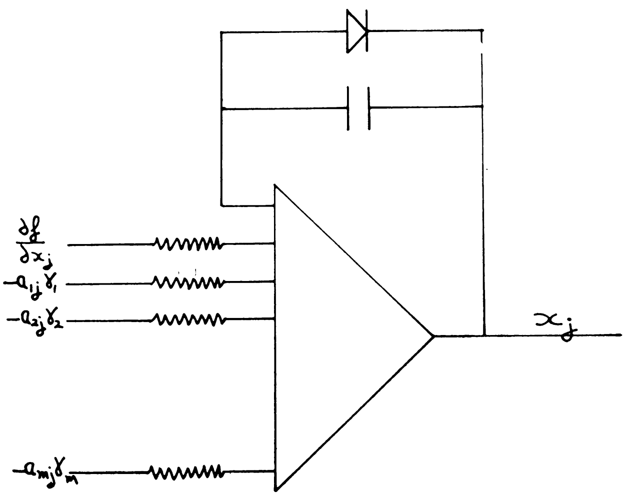

Fig. 3.—x_j integrator with constraints

5. Possible Types of Solution

(*a*) The most straightforward optimization solution occurs when the minimum or maximum value of $f(x_1, x_2 \ldots x_n)$ lies within the region specified by the constraints on $x_1 \ldots x_n$. In this case the constraints do not play any great part in the solution. This type of solution is only possible for a non-linear "objective" function $f(x_1 \ldots x_n)$, as the very nature of this type of solution implies the existence of a minimum or maximum value in $f(x_1 \ldots x_n)$. A linear function $f(x_1 \ldots x_n)$ does not possess stationary points, and so linear-programming problems are only soluble under constraints and are therefore all of type (*b*).

For the solution of a problem with a non-linear objective function the conditions $\frac{\partial f}{\partial x_j} = 0$ are satisfied, and thus integration ceases precisely when the x_j give the minimum value of $f(x_1 \ldots x_n)$. The best accuracy is found in this type of solution, and this should lie well within the 0·5%.

(*b*) When the absolute minimum value of the function $f(x_1, x_2 \ldots x_n)$ lies outside the restricted region for $x_1 \ldots x_n$ the solution is entirely determined by the constraints

$$a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n \geqslant a_{i0} \quad (i = 1 \ldots m).$$

The x_j will be driven in the direction of the minimum (or maximum) value of $f(x_1 \ldots x_n)$ until a restrictive boundary is encountered. The $x_1 \ldots x_n$ will then follow the boundary until $f(x_1 \ldots x_n)$ is optimized under the restrictions applied. An example of this type is given later in the text.

As $\frac{\partial f}{\partial x_j}$ can never be made zero for this case (the absolute minimum or maximum is never reached) the solution to the problem is obtained with a slight error, because the inputs to the x_j integrators can only be zero if there exists a $a_{ij}\gamma_i$ voltage contribution to cancel $\frac{\partial f}{\partial x_j}$, i.e. if one or more of the i restrictions are minutely violated.

Due to the characteristics of the diode feedback circuits providing the large γ_i, this violation is very small indeed, so the error is minute and the overall solution error should not exceed 0·5% for a medium-sized problem.

(*c*) Linear-programming problems, by the very nature of $f(x_1, x_2 \ldots x_n)$ which is a linear function of $x_1 \ldots x_n$, cannot have a stationary value no matter how wide the restricted domain. In every case the optimum solution is dictated by the constraints of the problem, and so the remarks of solutions type (*b*) apply.

In the majority of linear-programming problems $f(x_1 \ldots x_n)$ is the cost function to be minimized or the profit function to be maximized. For the transportation problem $f(x_1 \ldots x_n) = c_1x_1 + c_2x_2 \ldots + c_nx_n$, where the c_i are the transportation costs associated with the n different routes. In this case $\frac{\partial f}{\partial x_j} = c_j$ is the cost function for the jth route and so is a constant for the simplest of problems.

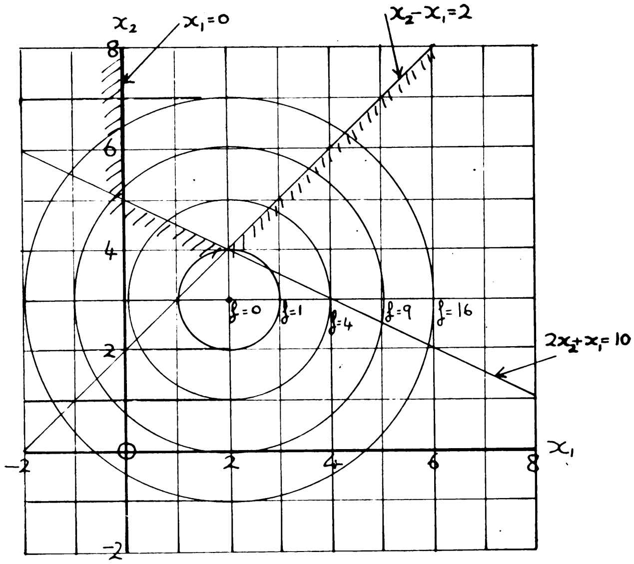

Fig. 4.—Example

6. Illustrative Example

To illustrate the application of the optimization procedure described in this paper to an actual problem, a very straightforward example is now considered.

It is supposed that the actual problem investigation has been carried out by a research team and the mathematical model of the problem constructed. The problem is thus reduced to the minimization of a function of two variables $f(x_1, x_2)$ where

$$f(x_1, x_2) = x_1^2 - 4x_1 + x_2^2 - 6x_2 + 13 \qquad (1)$$

subject to the restrictions

$$x_1 \geqslant 0 \qquad (2)$$

$$x_2 \geqslant 0 \qquad (3)$$

$$x_1 + 2x_2 \geqslant 10 \qquad (4)$$

$$x_2 - x_1 \geqslant 2. \qquad (5)$$

The solution to this simple problem may be conveniently shown graphically as demonstrated in Fig. 4.

First the function $f(x_1, x_2)$ (1) is constructed for various constant values of f. The given function $f(x_1, x_2)$ of course represents a family of circles centre (2, 3). If no restrictions were imposed in the problem then the absolute minimum solution would be attained giving $f = 0$ at $x_1 = 2, x_2 = 3$. However, restrictions (2) . . . (5) exist and these are drawn on to the graph as the lines

$$x_1 = 0 \qquad (6)$$

$$x_2 = 0 \qquad (7)$$

$$x_1 + 2x_2 = 10 \qquad (8)$$

$$x_2 - x_1 = 2. \qquad (9)$$

The restricted domain lies above the shaded boundary. It is noted that the condition $x_2 \geqslant 0$ plays no part in the solution, as this condition is already contained in the other three restrictions.

The value of f is gradually minimized as $x_1 \rightarrow 2$ and $x_2 \rightarrow 3$, and the point of closest approach to $x_1 = 2$, $x_2 = 3$ within the restricted domain is the point $x_1 = 2$, $x_2 = 4$ at which point $f = 1$. This is the required minimum solution.

The graphical method of determining the minimum value of f is useful only for two-variable problems, but for the present purpose it gives a good picture of optimization solution.

For the analogue computer both $\frac{\partial f}{\partial x_1}$ and $\frac{\partial f}{\partial x_2}$ are required:

$$\frac{\partial f}{\partial x_1} = 2x_1 - 4$$

$$\frac{\partial f}{\partial x_2} = 2x_2 - 6.$$

The driving voltages $\gamma_i (i = 1, 2)$ are set up from equations (4) and (5) as shown in Figs. 5 and 6.

The signs of γ_1 and γ_2 are then changed before feeding to the x_1 and x_2 integrators. The complete circuit for this simple quadratic example is shown in Fig. 7.

In Fig. 7 amplifiers 11 and 12 are the integrators to generate the minimizing variables x_1 and x_2. Both have diode feedbacks to ensure that $x_1 \geqslant 0$, $x_2 \geqslant 0$. Amplifiers 21 and 22 are restriction amplifiers which generate the driving voltages γ_1 and γ_2 respectively. γ_1 and γ_2 are sign changed in amplifiers 31 and 32 respectively. Amplifier 41 inverts the sign of x_1.

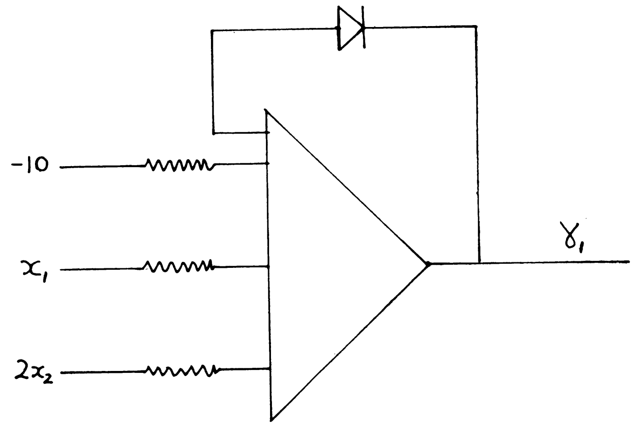

Fig. 5.—Generation of γ_1

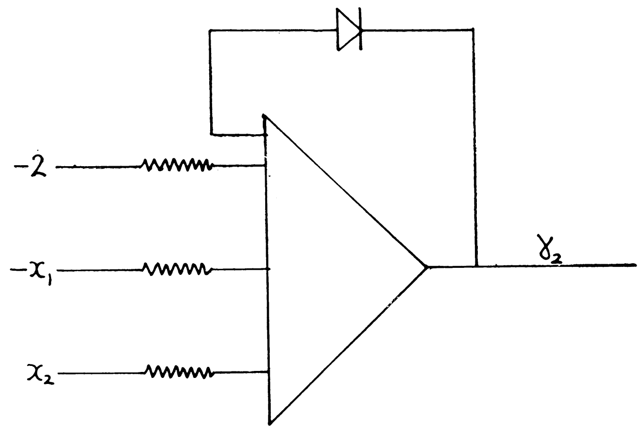

Fig. 6.—Generation of γ_2

7. Conclusions

It is noted that the solution to this non-linear optimization problem could be re-arranged to require only 6 amplifiers from a general-purpose analogue computer. It is thought that this number could possibly be reduced by a further 2 by making use of passive condenser-resistance networks to replace the integrating amplifiers used to generate x_1 and x_2.

However, confining our attention to conventional analogue computer units, the rule for calculation of the number of amplifiers necessary to solve linear or quadratic programming problems is:

(*a*) one amplifier per parameter x_j,
(*b*) two amplifiers per restriction.

Hence a problem in 20 parameters x_j under 15 constraint conditions would require 50 computer amplifiers. Similarly a problem in 200 parameters x_j under 150 constraints would require 500 amplifiers. 50-amplifier computers are today common in many Research Departments, while 500-amplifier computers have already been installed in several centres in this country.

Without increase of amplifier capacity, non-linear effects (such as step changes in cost coefficients) can be appropriately set into the problem, as demanded.

The solutions for a large-scale problem (with automatic sufficiency checks) might take 20 seconds to converge, but this is very favourable compared with solution times on a digital computer. The effects of small changes in critical coefficients may be examined very readily with the analogue, each new solution taking only approximately 20 seconds to converge.

Of course accuracy in the solutions is subject to limitations, but for a large scale problem would be better than 1%; it must be borne in mind here that a large number of industries employing optimization techniques are not able to provide problem data to the computer to an accuracy better than 1%.

Specially designed analogue computers for linear programming problems are being considered, with automatic setting up of the problem from digital input equipment, and with the facility for digital recording of results. Such computers could be usefully employed in an on-line sense in a closed loop system where high solution speeds are necessary.

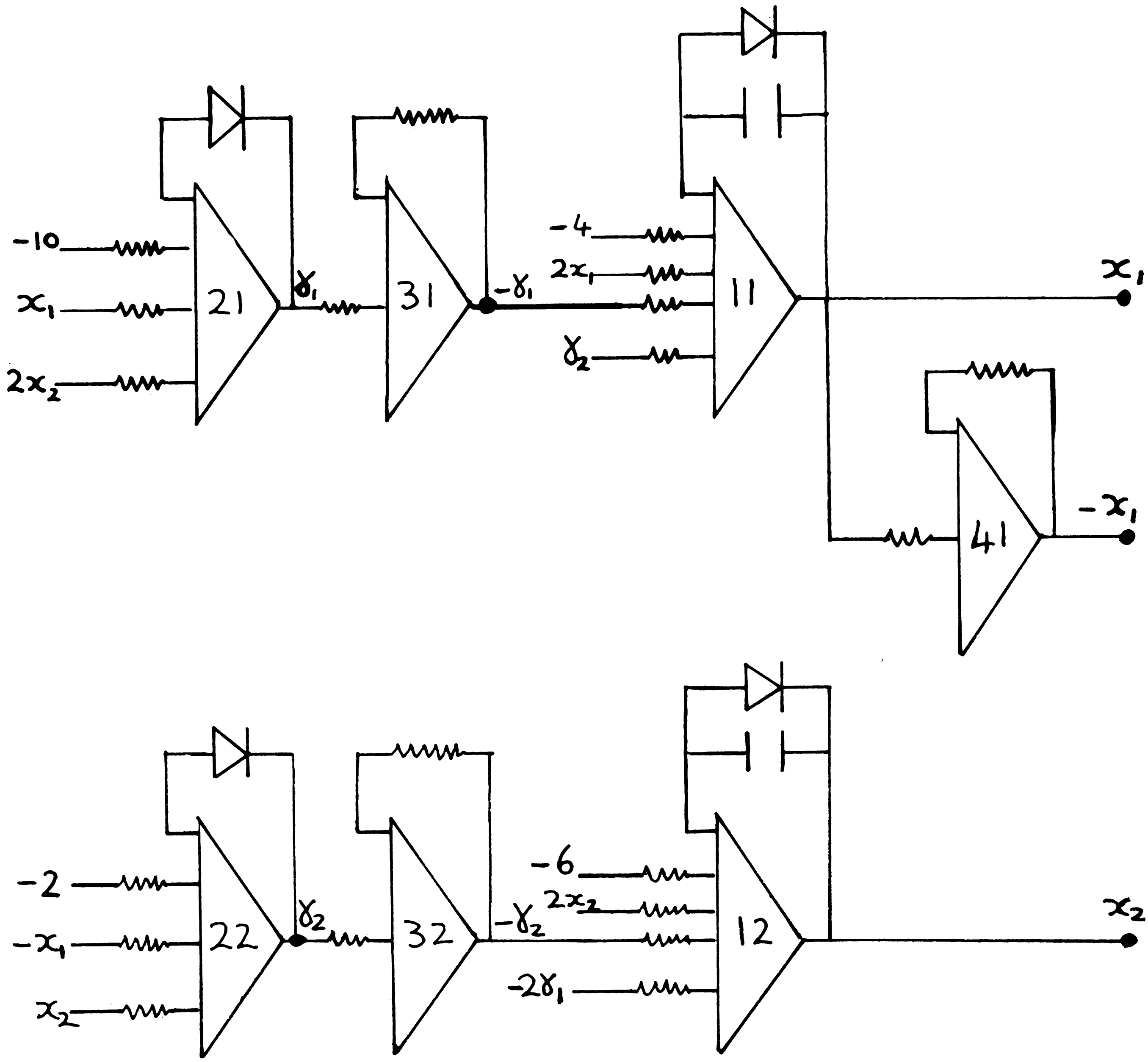

Fig. 7.—Analogue computer set-up

References

Ablow, C. M., and Brigham, G. (1955). "An Analog Solution of Programming Problems," *J. Operations Research Society of America*, Vol. 3, p. 388.

Jackson, A. S. (1960). *Analogue Computation*, McGraw-Hill.

Korn, G. A., and Korn, T. M. (1956). *Electronic Analog Computers*, McGraw-Hill.

Pyne, I. B. (1956). "Linear Programming on an Analog Computer," *Trans. A.I.E.E.*, Part 1, p. 139.

Thomason, J. G. (1961). "An Introduction to Analogue Computer Methods," *The Computer Journal*, Vol. 3, p. 211.