References

AITKEN, A. C. (1950). "Studies in Practical Mathematics. On the Iterative Solution of a System of Linear Equations," *Proc. Roy. Soc. Edinburgh*, A.63, pp. 652–60.

FORSYTHE, G. E., and WASOW, W. R. (1960). *Finite Difference Methods for Solving Partial Differential Equations*, pp. 220–35, London: Wiley.

HABETLER, G. J., and WACHSPRESS, E. L. (1961). "Symmetric Successive Over-Relaxation in Solving Diffusion Difference Equations," *Math. Comput.*, Vol. 15, pp. 356–63.

MARTIN, D. W., and TEE, G. J. (1961). "Iterative Methods for Linear Equations with Symmetric Positive Definite Matrix," *The Computer Journal*, Vol. 4, No. 3, pp. 242–53.

SHELDON, J. W. (1955). "On the Numerical Solution of Elliptic Difference Equations," *Math. Tables. Aids. Comput.*, Vol. 9, p. 101.

Book review: Dynamic programming

Applied Dynamic Programming, by RICHARD E. BELLMAN and STUART E. DREYFUS, 1963; 363 pages. (London: *Oxford University Press*, £2 15s. 6d.)

The type of problem giving rise to the technique known as Dynamic Programming can be formulated in a few words. Supposing that I want to invest £1000 in 10 projects ("activities") $A_1 \ldots A_{10}$ and that the profit of A_i in terms of the invested capital x_i is given by a function $g_i(x_i)$. How should I subdivide my £1000 into 10 portions $x_1 \ldots x_{10}$ so as to obtain the maximum, called $f_{10}(1000)$, of the total profit $g_1(x_1) + \ldots + g_{10}(x_{10})$?

The answer given by Dynamic Programming is this. Assume that we already know the solution to the problem for the case of 9 activities. Allocating £x_{10} to activity A_{10} leaves £$(1000 - x_{10})$ to be invested in $A_1 \ldots A_9$. Naturally, $x_1 \ldots x_9$ will be chosen so as to maximize the return for $A_1 \ldots A_9$, and this maximum has the known value $f_9(1000 - x_{10})$.

The return for all 10 projects is then

$$g_{10}(x_{10}) + f_9(1000 - x_{10}),$$

and by varying x_{10} the maximum value $f_{10}(1000)$ of this expression can be computed.

In general, the problem of finding the maximum $f_N(x)$ of the sum $\sum_{i=1}^{N} g_i(x_i)$ subject to $\sum_{i=1}^{N} x_i = x$ is solved by the recurrence relation

$$f_n(x) = \max_{x_n} [g_n(x_n) + f_{n-1}(x - x_n)].$$

This method, and its modifications for solving generalized versions of the basic problem, has already been extensively discussed in Richard Bellman's first book published about six years ago. In spite of this, reports of its application up to now have been few and far between and, in contrast to Linear Programming which, in some industries, is used almost as widely as PAYE calculations, its "image" is still that of an ingenious mathematical technique of doubtful practical value.

The new book should do much to alter this situation. While the theoretical connections between Dynamic Programming and classical methods of analysis (such as Lagrange Multipliers and the calculus of variations) are by no means neglected, its emphasis is on practical applications and on the achievement of actual numerical results. The remarkable flexibility of the method is shown by the wide range of its possible uses; among those mentioned in the book are the loading of a ship of given capacity so as to maximize the total value of the cargo; transportation problems with non-linear cost functions; the problem of finding a defective coin by the least number of weighings; studies of the reliability of multi-component devices; minimizing the cost due to shortages of replacement parts; optimum advertising campaigns; smoothing and scheduling problems; computation of optimal trajectories; and the minimization of the number of steps when searching for the zero of a function.*

The solution of these and other problems is illustrated by detailed flow diagrams of the programs developed for the various modifications of the method; and the tabulation of data and results, together with statements of actual running times on existing computers, leave the reader in no doubt that problems of practical importance can be and have been solved by this method. The book will be required reading for anyone concerned with problems of optimization that cannot be solved by more familiar algorithms.

Paper, printing and proof-reading are good, but one slip (on page 19), though not affecting the argument, should be corrected in future editions: the statement that 10^7 seconds (actually about four months) "is something of the order of 10^5 hours, and thus of the order of magnitude of ten years."

D. G. PRINZ

* Another interesting application, using Dynamic Programming as a subroutine in Linear Programming for minimizing the scrap in cutting stock, was published too recently to be included (P. C. Gilmore and R. E. Gomory, "A linear-programming approach to the cutting-stock problem," *Jour. Op. Res. Soc. Am.*, Vol. 9, p. 849, Nov.–Dec. 1961)