Reduction in time between specification and final proving run may be as much as 75%.

Documentation has been helped enormously and standardized data descriptions, etc., have also contributed to economies achieved under the new system.

3. Unsatisfactory features

(i) *Expansion of files*

An extra five characters are added to every record. In the case of small records this could be important, and extra steps may be needed to create block-records in these cases.

(ii) *Incompatibility of files*

Intercode files cannot be read directly by CLEO programs.

(iii) *Compiling timc*

In general this may be equal to the normal translation time which is an integral part of the compilation process. In other words, the total translation time is nearly doubled. For average programs this time increases from 8 to 16 minutes. Savings are achieved, however, by substantial reduction in the number of re-translations and amendments required.

(iv) *Inefficiency*

Program expansion seems to be of the order of 10%, but efforts are being made to reduce this. The main inefficiency is related to the file expansion referred to above.

(v) *Training*

There is still no Teach-Yourself-CLEO manual.

(vi) *General*

We are in no way critical of the compiler's editing and reporting action. This is even better than we had expected. Our chief criticism is of the compiler's late arrival.

4. Conclusions

All our new programs will be written in CLEO, and all existing files will be converted to CLEO format. We have found that this system helps to balance programming and computing capacity, and furnishes a basis for training systems-analysts.

Reference

Thompson, T. R. (1962). "Fundamental principles of expressing a procedure for a computer application," *The Computer Journal*, Vol. 5, p. 164.

Book review: Mathematical programming

Recent Advances in Mathematical Programming, Edited by Robert L. Graves and Philip Wolfe, 1963; 347 pages. (Maidenhead: *McGraw-Hill Publishing Company Ltd.*, 92s.)

It is dangerous to include words like "recent" into the title of such a collection as this, which contains contributions to a Symposium held in 1962 at the University of Chicago. What was new then, is now at best a basis on which to build even more recent developments. Such thoughts have some topical significance in 1964, since in July of this year the sequel to the Chicago meeting will be held in London, sponsored by the *British Computer Society* in common with other associations.

The 43 titles in the book under review refer to 23 full papers and 20 abstracts, concerned with theory, applications, and computational aspects. We may assume that it is the last-mentioned category about which readers will expect to find some comments in this *Journal*.

Most papers on computational methods refer to the Simplex Method of Linear Programming, or to its adaptations to the non-linear case. M. Balinski describes a method for solving pairs of dual programs, while P. Wolfe gives a survey of non-linear methods, with a bibliography. Clair E. Miller describes a method for separable programming (the objective function consists of additive terms each of one single variable), which has proved to be applicable to many problems which arise in practice. G. B. Dantzig describes a "compact basis triangularization," and E. M. L. Beale the use of pseudo-basic variables, a method which is, in a sense, the dual to the Dantzig-Wolfe decomposition principle. Other algorithms inspired by the latter are given by J. M. Abadie and A. C. Williams, and by J. B. Rosen.

P. Wolfe and Leola Cutler report on tentative conclusions from SCEMP (Standardized Computational Experiments in Mathematical Programming) and D. M. Smith and W. Orchard-Hays on experiments with product form codes. Glenn T. Martin describes an accelerated algorithm for integer programming.

It will, of course, be appreciated that the practical man can also derive useful ideas from theoretical papers, and even from abstracts. It is therefore recommended that he should at least glance at the whole collection.

In any case, whoever the reader, the book is a useful reference to one stage in the development of mathematical programming, though perhaps not a very exciting one.

S. Vajda.