Lecture Notes
in Economics and
Mathematical Systems
Managing Editors: M. Beckmann and H. P. Künzi

183

Klaus Schittkowski

Nonlinear Programming Codes

Information, Tests, Performance

Springer-Verlag
Berlin Heidelberg New York 1980

Author

Klaus Schittkowski
Institut für Angewandte
Mathematik und Statistik
Universität Würzburg
Am Hubland
8700 Würzburg
Federal Republic of Germany

AMS Subject Classifications (1980): 90-04, 90C30

ISBN-13: 978-3-540-10247-2 e-ISBN-13: 978-3-642-46424-9

DOI: 10.1007/978-3-642-46424-9

2141/3140-543210

Preface

The increasing importance of mathematical programming for the solution of complex nonlinear systems arising in practical situations requires the development of qualified optimization software. In recent years, a lot of effort has been made to implement efficient and reliable optimization programs and we can observe a wide distribution of these programs both for research and industrial applications. In spite of their practical importance only a few attempts have been made in the past to come to comparative conclusions and to give a designer the possibility to decide which optimization program could solve his individual problems in the most desirable way.

Box [BO 1966], Huang, Levy [HL 1970], Himmelblau [HI 1971], Dumitru [DU 1974], and Moré, Garbow, Hillstrom [MG 1978] for example compared algorithms for unrestricted optimization problems. Bard [BD 1970], McKeown [MK 1975], and Ramsin, Wedin [RW 1977] studied codes for nonlinear least squares problems. Codes for the linear case are compared by Bartels [BA 1975] and Schittkowski, Stoer [SS 1979]. Extensive tests for geometric programming algorithms are found in Dembo [DE 1976b], Rijckaert [RI 1977], and Rijckaert, Martens [RM 1978]. For the general case, i.e. minimization of an arbitrary objective function under constraints, we have to mention first the Colville [CO 1968] report. Other relevant papers are Stocker [ST 1969], Holzman [HO 1969], Tabak [TA 1969], Beltrami [BE 1969], Miele, Tietze, Levy [MT 1972], Eason, Fenton [EF 1972], Asaadi [AS 1973], Staha [SH 1973], and Sandgren [SA 1977].

Comparative studies of the latter kind are either designed to test special modifications of a mathematical algorithm or to perform general purpose software tests. The number of optimization programs varies from 2 to 30 executed in up to 35 different versions. Most programs have been realizations of penalty methods and are therefore antiquated from the current point of view. In particular, none of the comparative studies tested so called quadratic approximation or recursive quadratic programming methods which will find growing interest both for future research efforts and applications because of their outstanding efficiency. In this study, we will introduce 20 different optimization codes in 26 versions. Additional versions are used to test the effect of numerical differentiation. Among these programs, we find realizations of quadratic approximation, generalized reduced gradient, multiplier, and penalty methods. Most of them are currently used to solve practical nonlinear programming problems in various kinds of applications. It is one of the intensions of this report to give technical information about the programs such as source, language, length, provision of problem functions, etc.. The programs will be tested extensively from different points of view to give a user the possibility to choose the most appropriate one for solving his individual optimization problems.

The exploitation of any experimental tests of nonlinear programming software depends on the quality of the used test problems since all conclusions can be confirmed only by numerical experiments. The test examples of earlier comparative studies are composed of small, artificial or, more often, of so called 'real life' problems which are believed to reflect typical structures of practical nonlinear programming problems, see Himmelblau [HI 1972], Cornwell, Hutchison, Minkoff, Schultz [CH 1978], or Hock, Schittkowski [HS 1980], but these test examples have some severe disadvantages especially since the precise solution is not known a priori. Therefore it is not possible to evaluate the achieved accuracy of an optimization program and to relate the efficiency, i.e. calculation time, number of function and gradient evaluations, to the accuracy. Furthermore, one has in general too little information on the mathematical structure of the test problems so that in the past, the efficiency of an optimization program was determined mainly in terms of calculation time or of the number of function and gradient evaluations. These drawbacks gave the impulse to develop a test problem generator which is capable to compute test problems randomly with predetermined (at least) local solutions. We are able now to determine not only the efficiency of a program and to relate it to the achieved accuracy, but also to develop quantitative measures for reliability and global convergence. Furthermore it is possible to generate convex, ill-conditioned, degenerate, and indefinite problems which are used for special purpose evaluations and to check the performance of optimization programs in extreme situations. To satisfy all these conditions, we generated 185 test problems randomly with predetermined solutions. Since most of them are executed with different starting points, each optimization code under consideration has to pass 370 test runs in contrast to at most 30 test runs performed in earlier studies.

A reader who is interested in selecting a program for the numerical solution of his optimization problems, could use the following guiding rules: He should start with Chapter I where the problem is formulated and then proceed to Chapter III. The first section will give him a survey of technical details such as length, provision of problem functions, embedded numerical differentiation, etc., leading to a special subset of programs which satisfy certain technical assumptions and which could be implemented to solve his problems. More detailed information and a rough sketch of their performance are contained in the second section where all programs are described individually. Subsequently, the decision maker should read Chapter V where the following nine performance criteria are evaluated:

Efficiency.
Reliability.
Global convergence.
Performance for solving degenerate problems.
Performance for solving ill-conditioned problems.
Performance for solving indefinite problems.
Sensitivity to slight variations of the problem.
Sensitivity to the position of the starting point.
Ease of use.

All these criteria are evaluated in a quantitative manner and, in addition, it is outlined how a final score may be obtained for optimization programs according to the individual significance of the criteria for the decision maker. More detailed numerical results and information about the performance evaluation are contained in Appendices C and D allowing a thorough investigation for special problem types.

In addition, the mathematical background of the algorithms is described in Chapter II and Chapter IV shows how test problems with predetermined solutions are generated randomly. Numerical data for their construction in the scope of this comparative study and a sensitivity analysis are given in Appendices A and B. Final conclusions, recommendations for the design of optimization programs, and some technical remarks are gathered in Chapter VI.

I would like to express my sincere gratitude to all authors for submitting their optimization programs and especially to the Rechenzentrum of the University of Würzburg for the support making it possible to perform the extensive numerical tests. The performance evaluation was influenced by many fruitful discussions with other COAL members (Committee on Algorithms of the Mathematical Programming Society). Furthermore I would like to thank J. Stoer and J. Abadie for helpful comments and suggestions.

Contents