

Preface

Mathematical optimization is important in the application of computer power to problems such as minimization, maximization, and resource optimization. IBM has long been represented in this area by products such as the IBM Mathematical Programming System Extended/370 (MPSX/370). The newest set of broadly applicable tools is known as the IBM Optimization Subroutine Library (OSL). An initial essay and seven papers following it describe OSL and its capabilities. We are indebted to D. G. Wilson of the IBM Application Solutions Division in Kingston, New York, for his efforts in bringing together this exposition on OSL.

The first OSL contribution is the essay by Wilson and Rudin that places the rest of the OSL papers and OSL itself in perspective, historically and technically. OSL is described as capable of addressing linear, quadratic, mixed-integer, and pure network programming problems across a variety of platforms. The essay conveys the scope of problems that can be solved using these techniques, as well as the philosophy behind OSL and its technical content.

Forrest and Tomlin provide us with two papers on methods used in OSL for solving linear programming problems. In the first, the simplex method as implemented for OSL is described. The authors pay particular attention to the effect on the algorithms of such factors as robustness, reliability, stability, speed, and the need to execute on hardware platforms with substantial architectural differences.

In their second contribution, Forrest and Tomlin describe the interior point (barrier) algorithms for OSL. Most of the paper is concerned with the characteristics of the matrices for these methods and the adaptation of the methods to hardware platforms such as mainframe computers and workstations.

Convex quadratic programming problems are discussed next in a paper by Jensen and King. These

algorithms are particularly useful for portfolio analysis, with the linear terms of the objective function representing expected return and the quadratic terms representing risk. The authors first decompose their quadratic algorithms in order to prepare for the simplex methods of OSL, and then generate an optimal solution from that decomposed solution.

Programming systems and applications that utilize OSL must present and receive data in a form that is recognized by the OSL algorithms, whether or not that form is used elsewhere in the systems or applications. In particular, many systems that could use OSL rely on representations of problems that are expressed symbolically. Minkoff shows an organized way to translate symbolics into computer code suitable for interaction with OSL routines.

Jensen and King provide a second paper on OSL, presenting their pilot graphic user interface for portfolio optimization, known as Frontier. A particular feature of this interface is its new implementation of the piecewise linear-quadratic risk framework, which allows a user to choose and test various portfolios under a user-supplied risk assumption. This work is an advanced application showing the use of OSL capabilities.

An early application of OSL to airline crew scheduling served as both a practical use of the new algorithms and a first look at the usefulness of OSL in a customer situation. Anbil and Tanga of American Airlines Decision Technologies and Johnson of IBM have joined to present the results of that effort. The problem is complex and is tightly constrained by legal, contractual, and cost considerations. The use of OSL provided reductions of 5 to 11 percent in crew costs.

The last contribution related to OSL in this issue is a look at where the technology stands in the area of mathematical programming and optimiza-

tion, and where it may well go in the foreseeable future. Of particular interest to the authors, Johnson and Nemhauser, are the types of problems that tools such as OSL may make workable for the first time and the further algorithmic and methodological improvements that would complement the present exploitation of recent hardware advances.

Hazony and Zeidner present a software development environment that allows for the customized development of complex engineering application systems at low cost and in a short period of time. The environment is based on APL2* and takes advantage of the mathematically expressive power of APL to avoid much of the programming effort generally associated with complex engineering software. The authors show how their environment and related tools can support the specification and development of application structures, algorithms, and systems, including those intended for use with distributed processing.

One of the challenges faced by systems programmers is operating system support for a large and generally increasing number of devices. Often, unique device drivers are created for each device, with the expectable effect on development time and maintenance. Feriozi describes a different approach, which can more efficiently and dependably support the growing set of devices. Operating system considerations are isolated from device considerations. Two separate physical parts are created that can be reused or substantially reused to create new drivers. He shows how this approach was taken to implement the Operating System/2* (OS/2*) Small Computer System Interface (SCSI).

Grossman describes the new architecture, performance characteristics, and linkages that are needed to support direct access storage devices (DASD) in the emerging systems configurations typified by IBM's Enterprise Systems Connection* (ESCON*) environment. Architectures built around parallel channels and channel commands are giving way under demands for lowered physical constraints, dynamic configuration control, and higher availability. The author focuses on the effects these changes are having on DASD storage control, while outlining the ESCON environment.

As the *Journal* begins its 31st year, we would like to acknowledge the support of readers, authors,

and referees that makes such a long history possible. We thank you and encourage you to continue your interest and participation in this publication. It also seems appropriate at such a time to state a few facts that sometimes escape us as we focus on a single paper, theme, or special issue. First, this publication is a quarterly refereed technical journal, which means that the integrity of each paper is ensured by a process that depends upon peer reviews of content, currency, and value by recognized experts within and outside IBM. Second, it is intended for the software and systems professional community worldwide. The papers are written for a technically aware readership, and we welcome submissions by knowledgeable authors around the globe, within and outside IBM. Third, the *Journal* has over 100 000 subscribers worldwide. Of those, approximately two-thirds are customers, technical professionals, and researchers; one-third are IBM employees; two-thirds are within the United States; and one-third are outside the U.S.

One of the specific ways in which the authors have provided support is through their responses to a recent survey of a sample of authors from the last four years. We want to thank the 64 authors who participated (out of 137 sampled) and convey some of the more interesting results of our 21 questions. First, we are pleased to note that the three primary reasons for submitting a paper to the *Journal* were its reputation, broad readership, and the opportunity for professional recognition. Second, authors found the referees, editorial staff, and guidelines for authors (as published in Volume 29, Number 4) to be quite helpful to the process of creating a quality paper, both in terms of content and appearance. Third, many of the authors stated that they and their peers find the *Journal* to be of significant value, use it in their work, and keep it for over two years. We thank the authors for their willingness to help us assess our process and its results, and, of course, for their very positive remarks.

The next issue of the *Journal* will focus on recent advances in software and systems for communications network management.

Gene F. Hoffnagle
Editor

*Trademark or registered trademark of International Business Machines Corporation.