Software Section

One of the most important questions regarding optimization software is what is the *best* available code. As is often the case, the best complexity algorithm for a specific problem does not correspond always to the best software. There are two reasons for this:

- The worst case complexity of an algorithm may not reflect the average case behavior of the algorithm. A classic paradigm is the ellipsoid algorithm and simplex type algorithms for linear programming. Although the ellipsoid algorithm is a polynomial time algorithm and the simplex is an exponential time algorithm, the average case behavior of the ellipsoid algorithm is inferior to the simplex algorithm.
- An efficient implementation of an algorithm can make the code run much faster than a poor implementation of a better complexity algorithm. This is true especially in the modern era of object oriented implementations, vector processors, and massively parallel hardware. Any implementation that exploits the vector concurrency will run faster. Take the case of solution of linear systems. If one uses the iterative method of *Conjugate Gradients* which vectorizes naturally, the code is likely to run faster than the traditional *Gaussian Elimination*. As another example, in C++, one can "in-line" many functions and make the code more efficient.

It is impossible to find a universal criterion to measure the performance of a given code. Test problems provide a standard platform for comparing robustness, accuracy, speed of execution, and other characteristics of algorithm efficiency. In the past years many collections of test problems appeared in the literature or have been available by electronic mail [1, 2]

It is the purpose of this column to report a summary of recent available mathematical software related to global optimization and interesting test problems. In addition, any comments on existing software or *improved* solutions to known test problems will be published in this column. Please send any relevant information or submit related material to

K.G. Ramakrishnan
Room 2C-126
AT&T Bell Laboratories
600, Mountain Ave.

Journal of Global Optimization **4:** 111–112, 1994.

Murray Hill, N.J. 07974
e-mail: kgr@research.att.com
(908) 582-6722

References

1. J. J. Dongarra and E. Grosse (1987), Distribution of Mathematical Software Via Electronic Mail, *Communications of the ACM* **30**, 403–407.
2. D. S. Johnson and C. C. McGeoch, eds. (1993), *Network Flows and Matching: First DIMACS Implementation Challenge,* DIMACS Series Vol. 12, AMS.