ORSA Journal on Computing

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

Rejoinder—The Last Word on Interior Point Methods for Linear Programming—For Now

Irvin J. Lustig, Roy E. Marsten, David F. Shanno,

ORSA Journal on Computing
Vol. 6, No. 1, Winter 1994

0899-1499/94/0601-0035 $01.25

REJOINDER

The Last Word on Interior Point Methods for Linear Programming—For Now

IRVIN J. LUSTIG / *CPLEX Optimization, Inc., Princeton, NJ 08544; Email: irv@dizzy.cplex.com*

ROY E. MARSTEN / *School of Engineering and Operations Research, Georgia Institute of Technology, Atlanta, GA 90332*

DAVID F. SHANNO / *Rutgers Center for Operations Research, Rutgers University, New Brunswick, NJ 08903, Email: Shanno@dantzig.rutgers.edu*

The authors first wish to thank the editors for choosing the commentators, and to thank the commentators for their very careful and thoughtful commentaries. We find little to disagree with in the commentaries, but believe that two major points deserve some further discussion and clarification.

In response to Bob Bixby's computational results, we wish to make several comments. First, our article stresses the rapid improvement in both simplex and barrier method solvers over the last few years. In order to have this article ready for the refereeing process by December of 1992, we had to do our reported computer runs in September and October of 1992. It is our understanding that during this period, many of the improvements in the CPLEX simplex solver were still under development. Certainly the version of CPLEX used in Bixby's commentary was not available to us at that time. The salient point of the disparity between his results and our results is that the rejuvenation of the simplex method is still very much underway. The scope of the improvements in a very short period of time is well presented in Bixby's commentary, and strike us as every bit as interesting and important as the progress with barrier codes.

A second comment concerns the problem of solving a large model for the first time. While the simplex method can now solve many very large models efficiently, both the OSL runs and Bixby's runs demonstrate clearly that this efficiency often depends on which variant of the simplex method is chosen. A great deal of computational effort can be expended in determining which, if any, variant of the simplex method is the correct one for a given model. In contrast, with the exceptions of occasionally removing dense columns or very rarely altering the starting point, the barrier method requires only one run to determine the solution. Thus for most new models, we feel that the barrier method is still likely to find the solution at the lowest cost. This is especially true given the very rapid times for basis recovery given by Bixby, again reporting on work accomplished after our article was written.

As a last comment on comparing the simplex method to a barrier method, we again note that in most cases where the barrier performs poorly, it is because either AA^T has too many nonzeros, or, when AA^T is acceptably sparse, the Cholesky factor L has too many nonzeros. Here we wish only to note that both of these cases are determined before the first factorization is computed, so it is unnecessary to expend large amounts of computational work to uncover those models where the simplex method is most likely to prove superior.

Beyond comparison with the simplex method, the remainder of the material in the commentaries that we feel merits some response deals with the continuing evolution of interior point methods. Both Mike Saunders and Bob Vanderbei correctly point out the limitations of the Cholesky factorization as model size continues to increase. Certainly, solving the KKT system of equations does handle dense columns more efficiently. However, for standard models we have yet to see convincing evidence that algorithms for solving the KKT system can be made either as fast or as stable as our implemented Cholesky. When development of these algorithms is such that we are convinced that they represent a better alternative overall, then we will happily include them as state of the art. We are not yet at that point, but we would be the last to say that barrier methods cannot be improved. We hope that improvements will continue at a rate at least comparable to the simplex method.

Indeed, we are looking well beyond the KKT system for the linear algebra to solve very large problems. One possibility, explored slightly in our article, is parallelization. Today, good dense parallel linear algebra subroutines exist, but sparsity is a problem. To our knowledge, a straight Cholesky appears to be the best approach. Use of parallel techniques can also alleviate some of the requirements for the large memory space that is needed to solve some problems. Rather than have one processor with a large amount of memory, the required memory can be distributed over several processors. This is a topic for future research.

Another approach, largely touted but little documented by AT & T, involves solving the linear least squares system by preconditioned conjugate gradient methods. This has proved very successful in some partial differential equation problems, where knowledge of the problem allows for a good choice of preconditioner. We have experimented slightly with this approach, using an incomplete Cholesky factorization as a preconditioner, but our results are not encouraging. Mehrotra[1] has experimented more fully with such approaches, but again the results are not promising. Until we have seen fully described algorithms, with computational results that can be duplicated rather than patented, we must remain skeptical that this is a viable approach for solving general linear programs, unless one can exploit special structure in the model.

Another approach, which we have not attempted, is computing a sparse QR factorization on the matrix A. We have avoided this because it is two to three times slower than the Cholesky. However, it may prove useful for some large models where AA^T is acceptably sparse, but L is not, for the algorithm may produce an ordering with a sparser L. This remains for further study.

As a final note on the linear algebra, we would like to answer some of Mike Saunders' specific questions. The CRAY that was used to solve the large problem had a gigaword of memory. We needed all of it. We did have total control of all processors of the machine as it sat for a weekend on a loading dock waiting to be shipped to the customer. The termination on this problem was normal. We simply gave it 10^{-6} as the convergence tolerance. We did this because we noticed that convergence did not accelerate as the optimum was approached on the smaller model, so two more digits of accuracy were going to require an amount of extra computational effort that we could not justify. In fact, the problem required an integer solution, which startlingly, at least to us, was easily recovered from the six digit linear programming solution.

In general, we do monitor the accuracy with which the linear equations are solved, and when this is unacceptable, we use iterative refinement and small perturbations of the diagonal elements to assure specified accuracy. All problems reported in our article were solved to required accuracy. No dense columns were removed in our results, but Bixby's commentary does include a problem where 38 dense columns were removed successfully. This is still very necessary with our approach to the linear algebra. The excessive fill-in on the huge models of the paper was caused by coupling constraints, not dense columns. This was also responsible for the fill-in noted for problem *initial*.

Another approach to very large problems that we find interesting is decomposition. Mulvey and Ruszczynski[2] report very encouraging results on large stochastic programming problems. While the application of decomposition techniques to barrier methods is just beginning, we feel that it has great potential for at least some classes of problems.

Finally, in response to Mike Todd's commentary, we are as encouraged as he is by the coming together of theory and practice. We have never been comfortable developing algorithms that fly in the face of theory, but initially we had no choice if we wanted efficient algorithms. We cannot overemphasize how pleasant it has become to code algorithms such as the global convergence algorithm of Kojima et al. that is mentioned by Todd and have them improve performance in some cases, and almost never hurt. We look forward to much closer collaboration with the theorists in the future.

References

1. S. Mehrotra, 1992. Implementation of Affine Scaling Methods: Approximate Solutions of Systems of Linear Equations Using Preconditioned Conjugate Gradient Methods, *ORSA Journal on Computing* 4, 103–118.
2. J. M. Mulvey and A. Ruszczynski, 1990 A Diagonal Quadratic Approximation Method for Large Scale Linear Programs, Technical Report SOR 90-08, School of Engineering and Applied Science, Department of Civil Engineering and Operations Research, Princeton University, Princeton, NJ.