2

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (Leave blank)

2. REPORT DATE
December 1989

3. REPORT TYPE AND DATES COVERED
Final Report, 01 Mar 87 to 28 Feb 89

4. TITLE AND SUBTITLE
NEW APPROACHES TO LINEAR AND NONLINEAR PROGRAMMING

5. FUNDING NUMBERS
AFOSR-87-0196
61102F 2304/A8

AUTHOR(S)
Walter Murray and Michael A. Saunders

PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Department of Operations Research
Stanford University
Stanford, CA 94305-4022

8. PERFORMING ORGANIZATION REPORT NUMBER
AFOSR-TR- 90 0662

SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)
AFOSR/NM
Bldg 410
Bolling AFB DC 20332-6448

10. SPONSORING/MONITORING AGENCY REPORT NUMBER
AFOSR-87-0196

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release;
distribution unlimited.

Cb

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report describes technical progress during the final twelve months of AFSOR Contract 87-01962. The project involves study of the theoretical properties and computational performance of techniques that solve linear and nonlinear programs by means of nonlinear transformations.

The group at the Systems Optimization Laboratory (SOL) were the first to recognize the connection between Karmarkar's (1984) projective method and the logarithmic barrier method (see Gill, Murray, Saunders, Tomlin and Wright, 1986). It is now generally recognized that essentially all interior-point methods for linear programming inspired by Karmarkar's method are closely related to application of Newton's method to a sequence of barrier functions (see e.g., Gonzaga, 1987; Renegar, 1988; Anstreicher, 1988). Each barrier function is defined from the objective function and a barrier term that is infinite along the boundary of the feasible region. As the weight on the barrier term is reduced to zero, the solution of the subproblem becomes closer to the solution of the original problem.

14. SUBJECT TERMS

15. NUMBER OF PAGES
7

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
UNCLASSIFIED	UNCLASSIFIED	UNCLASSIFIED	SAR

Final report to the
DEPARTMENT OF THE AIR FORCE
Contract No. AFSOR 87-01962

NEW APPROACHES TO LINEAR AND NONLINEAR PROGRAMMING

March 1, 1988—February 28, 1989

Principal Investigators
Walter MURRAY
Michael A. SAUNDERS

Senior Research Personnel
Philip E. GILL

Department of Operations Research
Stanford University
Stanford, California 94305-4022

December 1989

1. PROJECT DESCRIPTION

This report describes technical progress during the final twelve months of AFSOR Contract 87-01962. The project involves study of the theoretical properties and computational performance of techniques that solve linear and nonlinear programs by means of nonlinear transformations.

The group at the Systems Optimization Laboratory (SOL) were the first to recognize the connection between Karmarkar's (1984) projective method and the logarithmic barrier method (see Gill, Murray, Saunders, Tomlin and Wright, 1986). It is now generally recognized that essentially all interior-point methods for linear programming inspired by Karmarkar's method are closely related to application of Newton's method to a sequence of barrier functions (see e.g., Gonzaga, 1987; Renegar, 1988; Anstreicher, 1988). Each barrier function is defined from the objective function and a barrier term that is infinite along the boundary of the feasible region. As the weight on the barrier term is reduced to zero, the solution of the subproblem becomes closer to the solution of the original problem.

2. REVIEW OF PROGRESS

2.1. Summary

During the last year, research has concentrated on barrier function methods for *linear* programming. Highlights of this year's research include:

- The completion of an implementation that treats linear programs in the form that occurs most often in practice, i.e., the general primal problem

  $$\min c^T x \quad \text{subject to} \quad Ax = b, \quad l \le x \le u.$$

  This work culminated in the first reported results on *all* the problems in the initial NETLIB test set. These results subsequently appeared in the thesis of Aeneas Marxen.

- The development of a sparse least-squares solver based upon a combination of Cholesky factorization, Schur complement and iterative refinement.

- The formulation of a new single-phase dual method that again treats the general primal problem with both upper and lower bounds.

- Completion of the preliminary theoretical analysis of a class of *shifted* barrier methods.

The connections with more traditional areas of nonlinear programming and numerical linear algebra, along with much analysis of path-following methods, have indicated that the new class of interior-point methods are capable of achieving good performance on a significant proportion of real-world problems. In terms of robustness the verdict is still out, since present implementations (within our experience) are highly sensitive to slight changes in strategy. However, there is every hope that useful and acceptably reliable implementations will be developed within the next few years.

For a review of recent developments in interior-point methods for linear programming, with an emphasis on computational results, see Report SOL 88-14.

2.2. Comparative Testing

Recent interest in new methods for linear programming has resulted in the need for serious computational comparisons between new algorithms and the simplex method. The wide availability of the portable optimization code MINOS 5.1 (Murtagh and Saunders, 1987) has given researchers throughout the United States the opportunity to compare new LP algorithms with a state-of-the art simplex code.

In order to facilitate a fair comparison of new methods with the simplex method, the simplex code MINOS 5.1 has been run on the *netlib* standard test set of 53 real-world problems compiled by Gay (1985a). The largest problem in this set is the PILOT model, which has about 1500 rows, 3700 columns and 43000 nonzeros. This problem is only medium-scale by conventional standards, yet "large" in the sense that a cold-start solution with the simplex method takes over 20 hours on a DEC VAXstation II.[1])

The results of the experiments are given in Lustig (1987). Lustig's results illustrate the speed-ups that can be obtained by invoking certain optional procedures in the simplex method, notably problem scaling and partial pricing.

An important feature of Lustig's work has been the production of a pictorial description of the zero/nonzero structure of the constraint matrix of each test problem. The pictures reveal that a large number of problems in the set have staircase structure. Various subsets of these problems have been used to compare the simplex method with interior-point algorithms. The more favorable results reported for the interior-point approach tend to be associated with strong staircase structure (see e.g., Gill, Murray, Saunders, Tomlin and Wright, 1986; Adler, Karmarkar, Resende and Veiga, 1987; Monma and Morton, 1987). This is fortuitous, since staircase problems have long been viewed as unusually difficult for the simplex method. Staircase problems tend to require many simplex iterations to solve and to have rather dense basis factorizations. It is probable that many problems of interest in the "real world" display staircase structure. Continued research on nonlinear methods for LP is therefore easily justified.

2.3. Sparse Least Squares

In *all* interior-point methods, the search direction is obtained from a system of the form

$$AD^2A^Tq = v, \tag{2.1}$$

where D is a diagonal matrix and v depends on the algorithm. If the right-hand side happens to be of the form $v = AD^2r$, this system is a set of "normal equations"

[1] This is a typical run-time for MINOS 5.3 (May 1988). A commercial Mathematical Programming System would take 10 to 20 minutes on an IBM 3090.

equivalent to the linear least-squares problem

$$\min \|D(r - A^Tq)\|_2. \tag{2.2}$$

The matrix AD^2A^T is *large*, it could be quite *dense* compared to A, and in general it is very nearly *singular*.

The principal method developed at SOL uses the *exact* Cholesky factors of AD^2A^T (excluding perhaps a few dense columns of A). The main reason is that the sparsity pattern of the normal-equations matrix does not change as D changes; hence a single "analyze" can be performed on the sparsity pattern of AA^T to obtain an ordering of the rows of A that preserves the sparsity of the Cholesky factor. The same ordering is used for all subsequent factorizations $AD^2A^T = R^TR$.

The analyze, factorize and solve procedures are performed using the off-the-shelf equation solver SPARSPAK (see George and Liu, 1981). If necessary, a partitioning (Schur-complement) scheme with iterative refinement is used to remove dense columns of A before the Cholesky factorization.

2.4. A Primal Barrier Method

Work has now been completed on a primal barrier method for linear programming. The analysis and development of the primal method constitutes the thesis of a graduate student Aeneas Marxen, which will be published (together with accompanying reports) later this year.

The primal barrier method has been the prototype for many of the investigations into the efficiency and reliability of the numerical procedures to solve the least-squares problem. Various techniques have been devised to *regularize* the least-squares problem. For example, the addition of a quadratic term to the barrier subproblem and the introduction of artificial slack variables. Both these modifications significantly improve the condition of the least-squares problem. If such or similar modifications were not made, the least-squares problem could be so ill-conditioned that the algorithm for computing the Cholesky factors of the matrix AD^2A^T would break down.

Until the primal method was completed, repeatable published results had involved only small- to medium-scale problems, with lower bounds (but no upper bounds) on the variables. For the first time, successful results have been obtained on all of the 53 test problems available in the *netlib* collection. These results were presented at the ORSA/TIMS Washington meeting in April (Gill, Marxen, Murray, Saunders and Wright, 1988a). Of particular interest is the solution time for PILOT: about 9 hours on a VAXstation II. This is a speed-up of 2.3 on a real-world model that is unquestionably non-trivial for the simplex method. The periodic structure revealed in Lustig (1987) may be a contributing factor, but in any event, this represents a bright note for the interior-point approach within the scope of current repeatable computational results.

Our preliminary conclusion from this work is that the primal method can be made reasonably reliable. However, the algorithm is highly sensitive to many of its parameters (e.g., the initial approximation to the solution, termination criteria, etc.). Both reliability and efficiency critically depend on choosing appropriate values for these and other parameters. It was found that the parameters needed for satisfactory performance on *all* of the problems gave poorer average performance on a subset of the problem having only lower bounds. This implies that published results concerning problems with only lower bounds may be optimistic.

2.5. A Single-Phase Dual Algorithm

It has already been noted (see Gill, Murray, Saunders and Wright, 1986) that applying the barrier transformation to the dual linear program has certain numerical advantages when solving the barrier subproblem. Report SOL 88-10 describes a new *single-phase* dual algorithm that treats both upper and lower bounds on the variables. At each iteration, estimates of both primal and dual optimal values are maintained. The dual variables are strictly interior to the dual linear program. The primal variables satisfy the constraints $Ax = b$ and approach feasibility with respect to the bounds $l \leq x \leq u$ as the solution is approached.

Consider the *dual linear program*:

$$\begin{array}{ll} \underset{\pi,y,z}{\text{minimize}} & -b^T\pi + u^Ty - l^Tz \\ \text{subject to} & -A^T\pi + y - z = -c, \qquad y,\ z \geq 0. \end{array} \tag{2.3}$$

In Report SOL 88-10 it is shown that the barrier search vector for the primal and dual variables x, π, y and z may be defined in terms of the vector q that satisfies the equations

$$AD^2A^Tq = AD^2r + \mu(b - Ax), \tag{2.4}$$

where D is a diagonal matrix.

An important benefit of the dual barrier formulation is that if x is chosen so that $Ax = b$, then (2.4) are the normal equations for a weighted least-squares problem of the form (2.2). Least-squares problems can be solved more reliably if treated as such. For example, conjugate-gradient methods generally require less iterations to solve (2.2) than they do to solve (2.1), particularly when the matrix DA^T is ill-conditioned (as it invariably is in this context). We note that *not all interior-point methods permit the least-squares formulation.*

A useful property of (2.3) is that for any value of the normal dual variables π, it is possible to choose positive values for y and z that satisfy the constraint $-A^T\pi + y - z = -c$. Thus an initial interior feasible point can be constructed easily, and there is no need for an "artificial column" of the kind that has frequently been introduced in this context.

Many components of y and z are "artificial variables" in the conventional sense. For example, if $u_j = \infty$, we know that y_j should be zero at an optimal solution. Similarly if $l_j = -\infty$. In practice we can change "∞" to a reasonably large number such as 10^6 and retain the components of y and z as long as convenient, even when we know their optimal values.

An experimental implementation to solve (2.3) is currently under development at SOL. In preliminary tests, speed-up factors in the range 1 to 13 relative to

MINOS 5.2 have been obtained for the largest of the 53 problems in *netlib*. For more details, see Report SOL 88-10. These results were presented at the 13th International Symposium on Mathematical Programming in Tokyo, August 1988 (Gill, Marxen, Murray, Saunders and Wright, 1988b).

2.6. The Shifted Barrier Method

Newton's method is based on minimizing a local quadratic model of the barrier function derived from first and second derivatives at the current iterate. Unfortunately, several difficulties can arise because of the nature of barrier functions. The extreme nonlinearity of the barrier term near the boundary means that a quadratic model is accurate only in a very small neighborhood of the current point. For a degenerate linear program, the Hessian of the barrier function becomes increasingly ill-conditioned at the solution when the barrier parameter is very small. Moreover, a strictly interior starting point may be inconvenient or impossible to obtain.

The shifted barrier methods developed at SOL are specifically designed to avoid these difficulties. The shifted barrier function is of the form

$$F(x) = c^T x - \sum_{j=1}^{n} w_j \ln(x_j + s_j),$$

where w and s are given positive vectors of *weights* and *shifts*.

The shifted barrier function enables *any* initial estimate to the solution to be used. Consequently, both an infeasible and/or a good estimate may be used. Neither of these choices is possible in current barrier methods—for example, it has been reported that the combined software/hardware system that is currently being marketed by AT&T (the KORBX^{TM} Linear Programming System) can fail to confirm a solution when the solution is used as the initial estimate.

By introducing shifts on the constraints we can bound the singularity away from the minimizer. Report SOL 88-9 describes methods for generating sequences of weights and shifts to ensure that the minimizer of $F(x)$ converges to the solution of the original linear program. It is shown that there is a considerable degree of freedom in specifying the weights and shifts. By a judicious choice, it can be assured that the Hessian is not only bounded but also reasonably well conditioned. By allowing *any* initial point, we believe that shifted barrier methods will be able to capitalize on good estimates of the solution.

2.7. Cycling in the Simplex Algorithm

The efficiency of the new methods is judged by comparing their results to those obtained by the simplex method. The most commonly used implementation in such comparisons is the SOL code MINOS. A consequence of having a point of comparison with the simplex method is to reveal some of *its* latent deficiencies. Therefore, it is important, if the comparisons are to be valid, to improve the simplex method whenever this is possible.

A feature of many problems is that degenerate vertices are common. Degeneracy is often regarded as a discomforting but otherwise tolerable hindrance to the simplex method, and to other active-set algorithms for solving optimization problems involving linear constraints. Sequences of non-improving steps are known to occur (perhaps many times during a run), but such sequences are rarely observed to be infinite. The phenomenon of "stalling" is therefore recognized and accepted, but "cycling" is deemed very unlikely to occur.

In spite of such folklore, a rigorous anti-cycling procedure can provide welcome peace of mind to users and implementors alike, particularly if the cost is small. Such a procedure was given by Wolfe (1963), and the possible benefits have been demonstrated recently by Ryan and Osborne (1986). We have devised a new anti-cycling procedure and incorporated it into MINOS (see report SOL 88-4). An objective of the new procedure is to preserved well-conditioned bases, and to guarantee termination on degenerate problems. Reliable performance has been achieved on all of the *netlib* problems. Several advantages exist over other anti-cycling methods; for example, there is no need to judge whether or not degeneracy is present.

References

I. Adler. N. Karmarkar, M. G. C. Resende and G. Veiga (1987). Data structures and programming techniques for the implementation of Karmarkar's algorithm, Manuscript (December 1987), Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA.

I. Adler, M. G. C. Resende and G. Veiga (1986). An implementation of Karmarkar's algorithm for linear programming, Report ORC 86-8, Department of Industrial Engineering and Operations Research, University of California, Berkeley, CA.

K. M. Anstreicher (1988). Linear programming and Newton barrier flow, *Mathematical Programming* 41, 367–373.

D. M. Gay (1985a). Electronic mail distribution of linear programming test problems, *Mathematical Programming Society COAL Newsletter* 13, 10–12.

D. M. Gay (1985b). Solving sparse least-squares problems, Seminar, Department of Operations Research, Stanford University, Stanford, CA.

J. A. George and J. W. H. Liu (1981). *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ.

P. E. Gill, A. Marxen, W. Murray, M. A. Saunders and M. H. Wright (1988a). Newton barrier methods for linear programming: some implementation aspects, presented at TIMS/ORSA Joint National Meeting, Washington, DC.

P. E. Gill, A. Marxen, W. Murray, M. A. Saunders and M. H. Wright (1988b). Barrier function methods for linear programming, presented at the 13th International Symposium on Mathematical Programming, Tokyo, Japan.

P. E. Gill, W. Murray and M. A. Saunders (1988). A single-phase dual barrier method for linear programming, Report SOL 88-10, Department of Operations Research, Stanford University, Stanford, CA.

P. E. Gill, W. Murray and M. A. Saunders (1988). Interior-point methods for linear programming: a challenge to the simplex method, Report SOL 88-14, Department of Operations Research, Stanford University, Stanford, CA.

P. E. Gill, W. Murray, M. A. Saunders, J. A. Tomlin and M. H. Wright (1986). On projected Newton barrier methods for linear programming and an equivalence to Karmarkar's projective method, *Mathematical Programming* 36, 183–209.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1986). A note on nonlinear approaches to linear programming, Report SOL 86-7, Department of Operations Research, Stanford University, Stanford, CA.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1988a). A practical anti-cycling procedure for linear and nonlinear programming, Report SOL 88-4, Department of Operations Research, Stanford University, Stanford, CA.

P. E. Gill, W. Murray, M. A. Saunders and M. H. Wright (1988b). Shifted barrier methods for linear programming, Report SOL 88-9, Department of Operations Research, Stanford University, Stanford, CA.

C. C. Gonzaga (1987). Search directions for interior linear programming methods, Memorandum UCB/ERL M87/44, Electronics Research Laboratory, College of Engineering, University of California, Berkeley, CA.

I. J. Lustig (1987). An analysis of an available set of linear programming test problems, Report SOL 87-11, Department of Operations Research, Stanford University, Stanford, CA.

N. Karmarkar (1984). A new polynomial-time algorithm for linear programming, *Combinatorica* 4, 373–395.

N. Karmarkar (1987). Recent developments in new approaches to linear programming, presented at the SIAM Conference on Optimization, Houston, TX, May 1987.

A. Marxen (1989) Primal Barrier methods for linear programming, Ph.D. Thesis, Department of Operations Research, Stanford University.

C. L. Monma and A. J. Morton (1987). Computational experience with a dual affine variant of Karmarkar's method for linear programming, *Operations Research Letters* 6, 261–267.

B. A. Murtagh and M. A. Saunders (1987). MINOS 5.1 user's guide, Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA.