Applied and
Computational
Mathematics
Division

NISTIR 4556

Computing and Applied Mathematics Laboratory

An Interior-Point Method for Linear and Quadratic Programming Problems

P.T. Boggs, P.D. Domich, J.E. Rogers, and C. Witzgall

April 1991

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Gaithersburg, MD 20899

NISTIR 4556

An Interior-Point Method for Linear and Quadratic Programming Problems

P. T. Boggs
P. D. Domich
J. E. Rogers
C. Witzgall

U.S. DEPARTMENT OF COMMERCE
National Institute of Standards and Technology
Computing and Applied Mathematics Laboratory
Applied and Computational Mathematics Division
Gaithersburg, MD 20899

April 1991

U.S. DEPARTMENT OF COMMERCE
Robert A. Mosbacher, Secretary
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
John W. Lyons, Director

An Interior-Point Method for Linear and Quadratic Programming Problems *

Paul T. Boggs † Paul D. Domich ‡ Janet E. Rogers §
Christoph Witzgall ¶

April 23, 1991

1. Introduction

We have been working on a particular class of interior point methods for solving linear programming problems for several years. (See, e.g., [DBRW91].) Our methods combine several search directions that are readily computed at each iteration. The final step is then calculated by computing the step that solves the original problem restricted to the subspace spanned by these search directions. In this paper we propose an extension of these ideas to the case of convex quadratic programming.

The linear programming (LP) problem that we consider is

$$\begin{aligned} &\min_u c^T u \\ &\text{subject to: } Au \leq b \end{aligned} \tag{1.1}$$

*Contribution of the National Institute of Standards and Technology and not subject to copyright in the United States. This research was supported in part by ONR Contract N-0014-87-F0053.

†Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899. INTERNET: boggs@cam.nist.gov

‡Applied and Computational Mathematics Division, National Institute of Standards and Technology, Boulder, CO 80303-3328. INTERNET: domich@cam.nist.gov

§Applied and Computational Mathematics Division, National Institute of Standards and Technology, Boulder, CO 80303-3328. INTERNET: jrogers@cam.nist.gov

¶Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899. INTERNET: witzgall@cam.nist.gov

and the quadratic programming (QP) is

$$\begin{array}{l} \min\limits_{u} c^{\mathsf{T}} u + \frac{1}{2} u^{\mathsf{T}} Q u \\ \text{subject to: } \; Au \leq b \end{array} \tag{1.2}$$

where $u, c \in \Re^n$, $Q \in \Re^{n \times n}$, $A \in \Re^{m \times n}$, and $b \in \Re^m$. We assume that Q is symmetric and positive semi-definite, i.e., we allow the possibility that some variables in the quadratic program enter only linearly. In addition we make the standard assumptions that the problems have a finite optimal solution and that A has full column rank. We do *not* assume that the problem has a full dimensional interior, nor do we assume that the problem is nondegenerate.

Interior point methods have been extended to the quadratic programming problem by several authors. For example, Shanno, *et al.* [Sha91] have been working on a primal-dual method, and Anstreicher has suggested a log barrier approach [AdHRT90] and a related dual version [Ans90] that is similar to our approach, although his work is primarily theoretical. Our work is based on the method of centers originally proposed by Huard [Hua67]. We believe that this framework provides some advantages over the primal-dual approach. In particular, as will be shown below, the Hessian matrix that is required involves the sum of Q and a matrix of the form $A^{\mathsf{T}}A$ whereas in the primal-dual framework the Hessian involves a matrix of the form $A^{\mathsf{T}}Q^{-1}A$. Clearly, even if Q is sparse, the inverse will generally not be, and thus this latter form is much more likely to be dense. Also, since the method of centers was originally proposed for general convex programs, the directions we compute are natural extensions of those for the linear programming case and can be computed at a low relative cost.

In §2 we present a brief description of the multi-dimensional method that forms the basis both for our linear programming work and for the quadratic extensions. In §3 we discuss the generation of the directions; in §4 we describe the details of our implementation; and in §5 we give a summary of our preliminary numerical results. These results indicate that the proposed algorithm can solve the quadratic programming problems in approximately the same time as that required for the linear problems. Finally, in §6 we note some directions for future research.

2. Multidimensional Methods

Our strategy for solving linear programming problems by interior-point methods is based on the observation that the formation and factorization of the Hessian

matrix consumes the majority of the CPU time. Thus our goal is to use the factorization to compute several independent directions, and to determine the "best" possible combination based on these directions. Viewed in this way, our strategy is similar in spirit to the "predictor-corrector" methods of Mehrotra [Meh89].

Our multidimensional algorithm for LP can be thought of as optimizing over a low-dimensional subspace at each major iteration. We formalize it as follows.

Multidimensional Algorithm

1. Compute a strictly feasible point, u^0; set i=0.
2. At u^i, generate q independent directions, $s^j,\ j = 1, \ldots, q$.
3. Form and solve the subproblem

$$\min_{\zeta} c^T \left(u^i + \sum_{j=1}^{q} \zeta_j s^j \right)$$
$$\text{subject to: } A \left(u^i + \sum_{j=1}^{q} \zeta_j s^j \right) \leq b$$

   to obtain ζ^*.
4. Set $u^{i+1} := u^i + \alpha \sum_{j=1}^{q} \zeta_j^* s^j$ where α is the steplength.
5. If convergence then
   Stop;
   Else, set $i := i + 1$; Go to 2.

Note that the subproblem in Step 3 is equivalent to

$$\min_{\zeta} \sum_{j=1}^{q} \zeta_j c^T s^j$$
$$\text{subject to: } \sum_{j=1}^{q} \zeta_j A s^j \leq b - A u^i.$$

The quantities $c^T s^j$, As^j, and $b - Au^i$ need only be computed once and thus this low-dimensional problem can be solved efficiently. We comment further on its solution in §4. The steplength α in step 4 is fixed at .99, a value in accordance with much of the work in interior-point methods for LP. (See §4 for details.)

The extension to QP of this general procedure is obvious. The subproblem objective function includes the quadratic term, i.e.,

$$\begin{array}{l} \min_{\zeta} \tilde{c}^{\mathsf{T}}\zeta + \frac{1}{2}\zeta^{\mathsf{T}}\tilde{Q}_q\zeta \\ \text{subject to: } Au \leq b \end{array} \tag{2.1}$$

where

$$\tilde{c}_j = c^{\mathsf{T}}s^j + (u^i)^{\mathsf{T}}Qs^j$$

and $\tilde{Q}_q \in \Re^{q\times q}$ is defined analogously.

In our work to date, our algorithm optimizes over a 3-dimensional subspace, and hence we designate the method by *O3D*. The subproblem solution in Step 3 has three variables and m constraints. In the LP case, we can efficiently solve this problem using a specialized, revised dual simplex procedure. In the QP case, this does not appear to be a viable option, and the QP subproblem is solved by a simplified interior-point method, the details of which are in §4.

3. Directions

The efficacy of the multidimensional algorithm depends critically on the directions that generate the subproblem. In this section, we derive our directions from the method of centers. To do this, we first present some notation.

Define the residuals for the constraints to be

$$r_k(u,t) = b_k - A_k u, \qquad k = 1,\ldots,m$$

where A_k is the kth row of A. Define the residual for the objective function to be

$$r_0(u,t) = t - c^{\mathsf{T}}u - \tfrac{1}{2}u^{\mathsf{T}}Qu$$

where t is a scalar whose value is determined as follows. Let u^0 be strictly feasible and let t^0 be the value of the objective function at u^0. Note that $r_k(u^0,t^0) > 0,\ k = 1,\ldots,m$ and that $r_0(u^0,t) > 0$ for $t > t^0$. Thus

$$\prod_{k=0}^{m} r_k(u,t^0)$$

is positive in the interior of the feasible region that corresponds to points where the objective function is less than t^0. The *center* can then be defined as the maximum of this product, or, equivalently,

$$\min_u L(u,t^0) = \min_u \sum_{k=0}^{m} \log r_k(u,t^0) \tag{3.1}$$

which defines $L(u,t)$. The *method of centers* is to solve (3.1) to obtain u^1, set t^1 equal to the objective function value at u^1, and repeat. It can be shown that this yields a sequence of points that converges to an optimal solution.

In [DBRW91] we show that if the constraint on the objective function is moved continuously then a trajectory is formed that approaches an optimal solution. More importantly, by a slight generalization, it can be shown that there exists a trajectory connecting every strictly feasible point to an optimal solution. The vector field defined by these directions is given by

$$s^1 = \nabla_{uu} L(u,t)^{-1} \nabla_{ut} L(u,t).$$

Our first direction, therefore, is s^1, known in the literature as the *dual-affine* direction [ARV89]. After some algebraic simplifications we have that s^1 is

$$s^1 = \beta_1 H^{-1}(c + Qu) \tag{3.2}$$

where

$$H = A^{\mathsf{T}} D^2 A + \frac{Q}{r_0(u,t)},$$

D is a diagonal matrix with kth diagonal entry $1/r_k(u,t)$, and β_1 is a scalar.

It is likewise possible to compute the so-called *recentering* direction, i.e., the Newton direction for solving (3.1) given by

$$\nabla_{uu} L(u,t)^{-1} \nabla_u L(u,t).$$

This direction is a combination of two directions, the first of which is s^1 and the second of which is our second direction, s^2, given by

$$s^2 = \beta_2 H^{-1} A^{\mathsf{T}} De \tag{3.3}$$

where e is the vector of all ones, and β_2 is a scalar.

The third direction is derived from considering the rank one effect on the Hessian matrix, H, of the first constraint, whose index is denoted by $\tilde{k}$, encountered in the s^1 direction. It is easily shown that this change is dominated by the vector $A_{\tilde{k}}$, and that the resulting direction is

$$s^3 = H^{-1} A_{\tilde{k}}^{\mathsf{T}}. \tag{3.4}$$

Finally, using the notation of factorable functions [JM86], we obtain our fourth direction, a third order correction to s^1, namely

$$s^4 = H^{-1} \sum_{k=1}^{m} A_k^{\mathsf{T}} \left[\frac{1}{r_k(u,t)} \right]^3 (A_k s^1)^2. \tag{3.5}$$

The details for all of these directions may be found in [DBRW91]. Our strategy, outlined below, is to choose three of these four directions at each iteration.

The only difference between the formulas here for the quadratic programming problem and those for the linear programming problem is the appearance of the Q term. One can readily observe that the sparsity of H is decreased by the addition of Q, but not catastrophically, as it is in methods such as those mentioned in §1. The work per iteration, therefore, is approximately the same for the quadratic program as for the linear program.

4. Implementation Details

In the results reported here, we use the basic *O3D* algorithm presented in §2 with the three directions, s^1, s^4, and either s^2 or s^3, to specify the subproblem. The selection of the third direction is based on the proximity to the optimal vertex. The implementation uses s^2 in early iterations, and s^3 in the final iterations. The "switch-over" is performed when the duality gap (see below) is less than or equal to 10^{-4} or the residual on the objective function (c.f., §3) is less than or equal to $1/m$. The main procedure continues until it satisfies at least one of three convergence criteria: (a) the relative change in the objective function, (b) the relative difference between the primal and dual objective values (see, e.g., [ARV89]), and (c) the steplength.

Note that computation of the dual variables is theoretically more complicated in the QP case. Our preliminary work, however, simply extends the techniques we used for the LP case, using $y = D^2A(A^\top D^2A + Q/r_0)^{-1}(c + Qu)$ to approximate a dual feasible solution, provided that $y \geq 0$. It can be shown that this is guaranteed to yield a dual feasible point in the limit, and this appears to be working well in practice.

The subproblem is solved by using an interior point approach on the three dimensional subspace. At each iteration of the subproblem, a dual affine direction (3.2) is computed and a step is taken either a fixed percentage of the distance to the boundary, or a distance which minimizes the objective function in that direction, whichever is smaller.

Problem scaling, starting values and the phase 1 procedure are exactly the same as used for our earlier LP subproblem work. In particular, both A and the subproblem constraint matrix defined in (1.2) are scaled. Our algorithm is initialized by setting $u_0 = 0$, where 0 denotes the 0-vector of the appropriate dimension, and then taking a single recentering step using a quadratic model in

the steepest descent direction, $A^T De$. When necessary, an initial feasible solution is obtained using a big-M Phase 1 procedure (see, e.g., [BJ77] and [DBRW91]).

The A matrix is stored in sparse format using the XMP experimental mathematical programming data structures described in [Mar81]. The Hessian matrix $(A^T D^2 A + Q/r_0)$ is stored using the data structures from the Yale Sparse Matrix Package SMPAK [SMP85]. A minimum-local-fill ordering procedure [VS83] is invoked only once at the beginning of the procedure to find a permutation of the rows and columns that reduces fill-in. The Hessian is then factored and solved using the Yale Sparse Matrix Package SMPAK [SMP85]. Constraints that are sufficiently far from the current point u, are explicitly removed from the computations.

The methods reported here are implemented in Fortran 77 and executed in double precision on an IBM RISC System/6000 Model 520 workstation running at 20MHz using the *xlf* compiler with the -O option.

5. Results

We create a set of QP test problems by augmenting 83 of the linear programming problems (publicly available through Netlib [Gay85]) with a quadratic term using $Q = I$. This allows an easy comparison of the work required to solve the QP problems with that required for the LP problems. Also, in the sense that any strictly convex QP can be transformed into one with $Q = I$, this can be regarded as a general formulation.

The convergence conditions and the corresponding total CPU time required for these QP problems are nearly identical to the LP results reported in [DBRW91]. For the 83 problems, 40 terminated due to the convergence criterion based on the relative difference between the primal and dual objective values. Of the remaining 43 problems that terminated due to the objective function convergence criterion, all but 3 found a point reasonably close (but not close enough for duality gap convergence) to the dual objective to indicate that the problem had solved correctly. Because the "true" objective function values are not readily available for these QP problems, however, we did not verify that the correct objective value was found. The total CPU time for the 83 QP problems is 2637 seconds, only 50 seconds more than the time required for our best LP results.

6. Future Directions

Our preliminary study demonstrates the feasibility of extending our multidimensional method, *O3D*, to the QP case. We intend to explore these ideas further by first attempting the problem set with a general positive semi-definite matrix Q. In these tests, we will not transform to $Q = I$ since we believe that transforming the problem to $Q = I$ will destroy too much of the sparsity. Next, although our subproblem solver has performed adequately in these preliminary tests, we think that some improvements are possible. Finally, we will investigate the use of this procedure in a sequential quadratic programming (SQP) algorithm for general large-scale nonlinear programming problems.

References

[AdHRT90] K. M. Anstreicher, D. den Hertog, C. Roos, and T. Terlaky. A long step barrier method for convex quadratic programming. Technical report, Delft University of Technology, Delft, Netherlands, 1990.

[Ans90] Kurt M. Anstreicher. On long step path following and sumt for linear and quadratic programming. Manuscript, Yale School of Organization and Management, August 1990.

[ARV89] Ilan Adler, Mauricio G. C. Resende, and Geraldo Veiga. An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming*, 44(3):297–335, 1989.

[BJ77] Mokhtar S. Bazaraa and John J. Jarvis. *Linear Programming and Network Flows.* John Wiley and Sons, Inc., New York, 1977.

[DBRW91] Paul D. Domich, Paul T. Boggs, Janet E. Rogers, and Christoph Witzgall. Optimizing over three-dimensional subspaces in an interior-point method for linear programming. *Linear Algebra and its Applications*, 152, July 1991.

[Gay85] David M. Gay. Electronic mail distribution of linear programming test problems. Mathematical Programming Society COAL Newsletter 13, December 1985.

[Hua67] Pierre Huard. Resolution of mathematical programming with nonlinear constraints by the method of centres. In J. Abadie, editor, *Nonlinear Programming*, pages 209–219. North Holland, Amsterdam, 1967.

[JM86] Richard H. F. Jackson and Garth P. McCormick. The polyadic structure of factorable function tensors with applications to higher-order minimization techniques. *Journal of Optimization Theory and Applications*, 51(1):63–93, 1986.

[Mar81] Roy E. Marsten. The design of the XMP linear programming library. *ACM Transactions on Mathematical Software*, 7(4):481–497, December 1981.

[Meh89] Sanjay Mehrotra. Implementations of affine scaling methods: Approximate solutions of systems of linear equations using preconditioned conjugate gradient methods. Technical Report 89-04, Northwestern University, August 1989.

[Sha91] David F. Shanno, 1991. Personal communication.

[SMP85] *SMPAK User's Guide Version 1.0*, 1985.

[VS83] Jiri Vlach and Kishore Singhal. *Computer Methods for Circuit Analysis and Design.* Van Nostrand Reinhold Company, New York, 1983.

NIST-114A
(REV. 3-90)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY

BIBLIOGRAPHIC DATA SHEET

1. PUBLICATION OR REPORT NUMBER
NISTIR 4556

2. PERFORMING ORGANIZATION REPORT NUMBER

3. PUBLICATION DATE
APRIL 1991

4. TITLE AND SUBTITLE

An Interior-Point Method for Linear and Quadratic Programming Problems

5. AUTHOR(S)

Paul T. Boggs, Paul D. Domich, Janet E. Rogers, and Christoph Witzgall

6. PERFORMING ORGANIZATION (IF JOINT OR OTHER THAN NIST, SEE INSTRUCTIONS)

U.S. DEPARTMENT OF COMMERCE
NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY
GAITHERSBURG, MD 20899

7. CONTRACT/GRANT NUMBER

8. TYPE OF REPORT AND PERIOD COVERED

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (STREET, CITY, STATE, ZIP)

10. SUPPLEMENTARY NOTES

11. ABSTRACT (A 200-WORD OR LESS FACTUAL SUMMARY OF MOST SIGNIFICANT INFORMATION. IF DOCUMENT INCLUDES A SIGNIFICANT BIBLIOGRAPHY OR LITERATURE SURVEY, MENTION IT HERE.)

We have been working on a particular class of interior point methods for solving linear programming problems for several years. Our methods combine several search directions that are readily computed at each iteration. The final step is then calculated by computing the step that solves the original problem restricted to the subspace spanned by these several directions. In this paper we propose an extension of these ideas to the case of convex quadratic programming.

12. KEY WORDS (6 TO 12 ENTRIES; ALPHABETICAL ORDER; CAPITALIZE ONLY PROPER NAMES; AND SEPARATE KEY WORDS BY SEMICOLONS)

linear programming; quadrative programming; interior-point methods; method of centers

13. AVAILABILITY

[X] UNLIMITED

[ ] FOR OFFICIAL DISTRIBUTION. DO NOT RELEASE TO NATIONAL TECHNICAL INFORMATION SERVICE (NTIS).

[ ] ORDER FROM SUPERINTENDENT OF DOCUMENTS, U.S. GOVERNMENT PRINTING OFFICE, WASHINGTON, DC 20402.

[X] ORDER FROM NATIONAL TECHNICAL INFORMATION SERVICE (NTIS), SPRINGFIELD, VA 22161.

14. NUMBER OF PRINTED PAGES
12

15. PRICE
A02

ELECTRONIC FORM