INTERIOR-POINT METHODS FOR NONCONVEX NONLINEAR PROGRAMMING: ORDERINGS AND HIGHER-ORDER METHODS

DAVID F. SHANNO AND ROBERT J. VANDERBEI

Statistics and Operations Research
Princeton University
SOR-99-5

ABSTRACT. The paper extends prior work by the authors on LOQO, an interior point algorithm for nonconvex nonlinear programming. The specific topics covered include primal versus dual orderings and higher order methods, which attempt to use each factorization of the Hessian matrix more than once to improve computational efficiency. Results show that unlike linear and convex quadratic programming, higher order corrections to the central trajectory are not useful for nonconvex nonlinear programming, but that a variant of Mehrotra's predictor-corrector algorithm can definitely improve performance.

1. INTRODUCTION

This paper extends previous work of the authors [14] on LOQO, an interior point algorithm for nonconvex nonlinear programming. The purpose of this section is to describe briefly the problem to be solved and the basic algorithm of [14].

For notational simplicity, we consider here only the inequality constrained nonlinear programming problem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h_i(x) \geq 0, \qquad i = 1, 2, \ldots, m, \end{array} \tag{1}$$

where x is a vector of dimension n and $f(x)$ and the $h_i(x)$ are assumed to be twice continuously differentiable. This paradigm is sufficient to describe fully the concepts to be introduced

Date. May 19, 1999.

1991 *Mathematics Subject Classification.* Primary 90C30, Secondary 49M37, 65K05.

Key words and phrases. Nonlinear programming, interior-point methods, nonconvex optimization, predictor-corrector, matrix ordering.

Research of the first author supported by NSF grant DMS-9805495. Research of the second author supported by NSF grant DMS-9870317 and by ONR grant N00014-98-1-0036.

in this paper. The extension of the algorithm to problems with equality constraints, simple bounds, and range constraints is discussed in [14] and [13].

The interior point method of [14] adds slack variables to (1), transforming the problem to

$$\begin{array}{lll} \text{minimize} & f(x) & \\ \text{subject to} & h(x) - w & = 0, \\ & w & \geq 0, \end{array} \tag{2}$$

where $h(x)$ and w represent the vectors $h_i(x)$ and w_i respectively. The nonnegativity constraints of (2) are then replaced with a logarithmic barrier term in the objective function, resulting in the transformed problem

$$\begin{array}{ll} \text{minimize} & f(x) - \mu \sum_{i=1}^{m} \ln w_i \\ \text{subject to} & h(x) - w = 0. \end{array} \tag{3}$$

The Lagrangian for (3) is

$$L_\mu(x, w, y) = f(x) - \mu \sum_{i=1}^{m} \ln(w_i) - y^T(h(x) - w),$$

and the first order conditions for a minimum are

$$\begin{array}{rcccl} \nabla_x L_\mu(x, w, y) & = & \nabla f(x) - \nabla h(x)^T y & = & 0, \\ \nabla_w L_\mu(x, w, y) & = & -\mu W^{-1} e + y & = & 0, \\ \nabla_y L_\mu(x, w, y) & = & h(x) - w & = & 0, \end{array} \tag{4}$$

where W is a diagonal matrix whose diagonal elements are the w_i, e is the m-vector of all ones, and ∇h is the Jacobian matrix of the vector h. Primal-dual methods modify the system (4) by multiplying the second equation by W, resulting in the system

$$\begin{array}{rcl} \nabla f(x) - \nabla h(x)^T y & = & 0, \\ -\mu e + WYe & = & 0, \\ h(x) - w & = & 0, \end{array} \tag{5}$$

where Y is the diagonal matrix with diagonal elements y_i. The basic primal-dual interior point method applies Newton's method to attempt to solve the system (5), leading to the system of equations

$$\begin{bmatrix} H(x, y) & 0 & -A^T(x) \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A^T(x) y \\ \mu e - WYe \\ -h(x) + w \end{bmatrix} \tag{6}$$

for the Newton corrections Δx, Δw, and Δy, where

$$H(x, y) = \nabla^2 f(x) - \sum_{i=1}^{m} y_i \nabla^2 h_i(x)$$

and

$$A(x) = \nabla h(x).$$

This system is not symmetric, but is symmetrized by multiplying the first equation by $-I$ and the second equation by $-W^{-1}$, leading to the system with which we will be concerned throughout the remainder of the paper, which is given by

$$\begin{bmatrix} -H(x,y) & 0 & A^T(x) \\ 0 & -W^{-1}Y & -I \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sigma \\ -\gamma \\ \rho \end{bmatrix}, \tag{7}$$

where

$$\begin{aligned} \sigma &= \nabla f(x) - A^T(x)y, \\ \gamma &= \mu W^{-1}e - y, \\ \rho &= w - h(x). \end{aligned}$$

A small modification of (7) is solved at each iteration of the basic LOQO algorithm. The modification comes from solving the second equation to eliminate Δw, which is quite simple since this equation contains only diagonal matrices. The vector Δw is given by

$$\Delta w = WY^{-1}(\gamma - \Delta y).$$

Substituting it into (7) yields the reduced KKT system

$$\begin{bmatrix} -H(x,y) & A^T(x) \\ A(x) & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sigma \\ \rho + WY^{-1}\gamma \end{bmatrix}. \tag{8}$$

Once the search directions have been determined, the algorithm proceeds iteratively, choosing a step length α at each iteration and determining the new estimates to the optimal solution by

$$\begin{aligned} x^{(k+1)} &= x^{(k)} + \alpha^{(k)}\Delta x^{(k)}, \\ y^{(k+1)} &= y^{(k)} + \alpha^{(k)}\Delta y^{(k)}, \\ w^{(k+1)} &= w^{(k)} + \alpha^{(k)}\Delta w^{(k)}. \end{aligned}$$

The step length α is chosen both to assure that $w \geq 0$ is maintained and that the merit function

$$\Psi_{\beta,\mu}(x,w) = f(x) - \mu \sum_{i=1}^{m} \ln(w_i) + \frac{\beta}{2}\|\rho(x,w)\|^2 \tag{9}$$

is sufficiently reduced. In [14], it is shown that at each iteration, there is a β for which the solution to (8) is a descent direction for the merit function (9) provided the matrix $H(x,y)$ is positive definite. To assure that this is the case, the algorithm modifies $H(x,y)$ with a diagonal perturbation,

$$\widehat{H}(x,y) = H(x,y) + \lambda I, \tag{10}$$

whenever during the factorization of the reduced KKT matrix a diagonal element with the wrong sign occurs. While conceptually the system solved by LOQO is the system (8), even before the perturbation λ is added, the system is modified to

$$\begin{bmatrix} -(H(x,y)+E_n) & A^T(x) \\ A(x) & E_m \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \Phi_1 \\ \Phi_2 \end{bmatrix}, \tag{11}$$

where E_n and E_m are positive definite diagonal matrices which arise in the handling of simple bounds, equality constraints, and the treatment of free variables. While this system is less likely to have an indefinite matrix $H(x,y)+E_n$, it can and often does become indefinite, and so a diagonal perturbation is necessary. The derivation of the matrices E_n and E_m are given in detail in [14] and [13].

While the solution of the system (11) seems straightforward, there are two approaches to solving the system which are mathematically equivalent but lead to dramatically different results in practice. These will be discussed in detail in the next section. Further, when using interior point methods to solve linear programming problems, it has been noted that often the major computational work at each iteration lies in forming and factoring the reduced KKT matrix, and higher order methods are used to attempt to use each factorization more than once (see, for example, [1], [10], [2], [7]). In nonlinear programming, the costs of forming and factoring the reduced KKT matrix may well be much higher, as the matrix contains second derivatives of the objective function and first and second derivatives of the constraint functions. Thus it appears worthwhile to use each factorization to the greatest possible extent. This is studied in a subsequent section. The final sections show that the algorithm can become stuck at a nonoptimal point, which has very real implications for any future convergence analysis, and conclude with a set of computational results that show that a carefully designed higher order method based on Mehrotra's predictor-corrector algorithm can enhance performance of the basic algorithm.

2. PRIMAL AND DUAL ORDERINGS

The purpose of this section is to examine two mathematically equivalent methods for solving the system (8), which have radically different performances in practice. For simplicity, we will rewrite (8) as

$$\begin{bmatrix} -H & A^T \\ A & WY^{-1} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} \sigma \\ \rho + WY^{-1}\gamma \end{bmatrix}. \tag{12}$$

There are two mathematically equivalent but conceptually different ways to solve the system (12). The first uses the first equation to eliminate Δx, yielding

$$\Delta x = -H^{-1}(\sigma - A^T \Delta y). \tag{13}$$

Substituting (13) into (12) gives the system

$$(AH^{-1}A^T + WY^{-1})\Delta y = \rho + WY^{-1}\gamma + AH^{-1}\sigma. \tag{14}$$

Solving (14) gives

$$\Delta y = (AH^{-1}A^T + WY^{-1})^{-1}(\rho + WY^{-1}\gamma + AH^{-1}\sigma). \tag{15}$$

Substituting (15) into (13) gives

$$\Delta x = -H^{-1}\sigma + H^{-1}A^T(AH^{-1}A^T + WY^{-1})^{-1}(\rho + WY^{-1}\gamma + AH^{-1}\sigma).$$

We refer to this method as the *primal method*, as it solves first for the primal corrections.

The alternative way of solving (12) uses the second equation to eliminate Δy, giving

$$\Delta y = W^{-1}Y\rho + \gamma - W^{-1}Y\Delta x. \tag{16}$$

Substituting (16) in the first equation gives

$$(H + A^T W^{-1}YA)\Delta x = A^T(W^{-1}Y\rho + \gamma) - \sigma. \tag{17}$$

As above, this gives

$$\Delta x = (H + A^T W^{-1}YA)^{-1}(A^T(W^{-1}Y\rho + \gamma) - \sigma)$$

and

$$\Delta y = W^{-1}Y\rho + \gamma - W^{-1}Y(H + A^T W^{-1}YA)^{-1}(A^T(W^{-1}Y\rho + \gamma) - \sigma).$$

We refer to this method as the *dual method*. (The equivalence of the expressions given in the primal and the dual method follows trivially from the Sherman-Morrison-Woodbury formula.)

The first significant difference between these two methods is in the sparsity of the matrices factored by the two methods. For large sparse problems, it is desirable to keep the factorization as sparse as possible. For interior point methods for linear programming, where H is diagonal, either method may be preferable for a given problem. When H is not diagonal, however, even when H is very sparse, H^{-1} is often very dense, and hence the resulting coefficient matrix of (14) is very dense, with a dense factorization. If both H and A^TA have reasonable sparsity patterns, however, the resulting factorization of the coefficient matrix in (17) can be quite sparse. Unless otherwise directed, LOQO chooses between ordering the primal or dual coefficient matrices to maximize sparsity in the factorization based on a heuristic which estimates which of the two methods are likely to produce the greatest sparsity. For nonlinear programming, it virtually always chooses the dual ordering.

While sparsity is an important consideration, especially on large problems, another issue enters in when solving nonconvex nonlinear programming problems and it dictates that the dual method should always be used on such problems. The issue arises from the way diagonal perturbations are added to the matrix H whenever a zero or negative diagonal element is encountered. The primal method first factors H, and hence will always encounter zero or negative diagonal elements when H is indefinite. The dual method, however, only factors the matrix $H + A^T W^{-1}YA$. As $W^{-1}Y$ is a positive definite diagonal matrix, this matrix can well be positive definite even when H is indefinite, which leads to much more stable and efficient algorithms.

As an example, consider the nonconvex problem

$$\begin{array}{lrl} \text{minimize} & -x_1x_2 & \\ \text{subject to} & x_1 + 2x_2 & \leq 4 \\ & x_1, x_2 & \geq 0. \end{array}$$

This problem has a minimum at $x_1 = 2$, $x_2 = 1$. Let w_1 and w_2 be the slack variables associated with the simple bounds, and let w_3 be the slack associated with the linear constraint. Let y_1, y_2, and y_3 the corresponding dual variables. For this problem, the diagonal matrices E_n and E_m appearing in (11) are given by

$$E_n = \text{Diag}\begin{pmatrix} \dfrac{y_1}{w_1} & \dfrac{y_2}{w_2} \end{pmatrix} \qquad \text{and} \qquad E_m = \begin{bmatrix} \dfrac{w_3}{y_3} \end{bmatrix}.$$

Hence, the coefficient matrix corresponding to (11) is

$$\begin{bmatrix} -\dfrac{y_1}{w_1} & -1 & -1 \\ -1 & -\dfrac{y_2}{w_2} & -2 \\ -1 & -2 & \dfrac{w_3}{y_3} \end{bmatrix},$$

where clearly

$$H + E_n = \begin{bmatrix} \dfrac{y_1}{w_1} & 1 \\ 1 & \dfrac{y_2}{w_2} \end{bmatrix}.$$

For details of handling bounded variables see [13]. For the primal method, $H + E_n$ is factored first. Therefore, diagonal perturbations will be added whenever the ratio of the y_i's to the w_i's becomes less than one. As neither bound is active at the optimum, these ratios quickly approach zero. The dual algorithm, however, factors the matrix

$$-\begin{bmatrix} \dfrac{y_1}{w_1} + \dfrac{y_3}{w_3} & 1 + 2\dfrac{y_3}{w_3} \\ 1 + 2\dfrac{y_3}{w_3} & \dfrac{y_2}{w_2} + 4\dfrac{y_3}{w_3} \end{bmatrix},$$

which will remain negative definite as y_3/w_3 goes to infinity as the optimum is approached, as at the optimum, $w_3 = 0$ and y_3 is nonzero. For this problem, the dual ordering algorithm solves to optimality in 10 iterations, and never perturbs the diagonal. In contrast, the primal algorithm solves to optimality in 180 iterations. It first perturbs the diagonal on the third iteration and then on every subsequent iteration. Thus for nonconvex problems, dual orderings appear to be essential, and are the default in LOQO. As a final note of caution, however, on convex problems it is possible that a primal ordering can produce far greater sparsity, depending on H and A. On the problem **`structure_socp`** ([12]), LOQO solved with the primal ordering

in 2 minutes 20 seconds, and in 96 minutes with the dual ordering. Thus the ordering should be chosen with care, but if the problem is nonconvex, the dual always seems preferable.

3. HIGHER ORDER METHODS

Almost from the inception of serious study of interior point methods for linear programming, the concept of using one factorization of the reduced KKT system several times in order to improve efficiency has been a subject of interest [1]. The first method to gain wide popular acceptance was Mehrotra's predictor-corrector method [10]. While the method was developed initially for linear programming, it proved easily adaptable to convex quadratic programming [2], [13], and it is within this context that we will briefly describe the method here. The convex quadratic programming problem we consider is

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T Qx + x^T c \\ \text{subject to} & Ax \geq b. \end{array}$$

Adding slacks as before, the problem becomes

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^T Qx + x^T c \\ \text{subject to} & Ax - w \geq b, \\ & w \geq 0. \end{array}$$

The KKT conditions corresponding to (5) are

$$Qx + c - A^T y = 0, \tag{18}$$

$$-\mu e + WYe = 0, \tag{19}$$

$$Ax - b - w = 0. \tag{20}$$

The only nonlinearity in this system occurs in (19), in the product WYe. Instead of applying Newton's method to the system (18)-(20), we can derive the predictor-corrector by substituting Δx, Δw, and Δy directly into the system of equations, giving

$$Q\Delta x - A^T \Delta y = -Qx - c + A^T y, \tag{21}$$

$$Y\Delta w + W\Delta y = \mu e - WYe - \Delta W \Delta Ye, \tag{22}$$

$$A\Delta x - \Delta w = b - Ax + w, \tag{23}$$

where ΔW and ΔY are the diagonal matrices with diagonal elements Δw and Δy, respectively. The system (21)-(23) is implicit, with Δw and Δy terms appearing on both sides of (22). The predictor-corrector algorithm first solves the system

$$Q\widehat{\Delta x} - A^T \widehat{\Delta y} = -Qx - c + A^T y, \tag{24}$$

$$Y\widehat{\Delta w} + W\widehat{\Delta y} = -WYe, \tag{25}$$

$$A\widehat{\Delta x} - \widehat{\Delta w} = b - Ax + w, \tag{26}$$

then dynamically chooses μ depending on the maximum step size which the solution to (24)-(26) can take to maintain $w \geq 0$. Current values for $\mu\widehat{\Delta w}$, and $\widehat{\Delta y}$ are then substituted into

the right hand side of (21)-(23), and the system is solved again to obtain the search direction. This process can be repeated by at each subsequent step substituting the latest solution pair Δw, Δy into the right hand side of (21)-(23). Recent papers by Gondzio [7] and Jarre and Wechs [9] indicate that computational efficiency can be improved by repeating this several times, where the number of times is dependent on both the cost of the factorization relative to a backsolve and on progress of the higher order terms toward a solution.

The effect of these higher order corrections is to attempt to move the approximate solution closer to the central path, the path on which

$$WYe = \mu e$$

for each value of μ. The central path has been essential theoretically in order to prove convergence of algorithms, and as noted has improved computational performance for linear programming. It would then seem attractive to replace (6) with

$$\begin{bmatrix} H(x.y) & 0 & -A^T(x) \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A^T(x)y \\ \mu e - WYe - \Delta W \Delta Y e \\ -h(x) + w \end{bmatrix}$$

and, without changing the coefficient matrix, attempt to iterate to a solution closer to the central path. Unfortunately, for nonconvex problems, the central path need not exist. To see this, consider the problem

$$\begin{array}{ll} \text{minimize} & x - x^2 \\ \text{subject to} & x \geq 0. \end{array}$$

This problem has a local minimum at 0 and a global minimum at infinity. The first order conditions for the barrier problem associated with this problem are

$$1 - 2x - \frac{\mu}{x} = 0,$$

which has the solution

$$x = \frac{1 \pm \sqrt{1 - 8\mu}}{2}.$$

Clearly, there is no real solution for $\mu > \frac{1}{8}$. While this may seem contrived, it apparently is very real, as we tried this higher order scheme on the Hock and Schittkowski test set [8], where the algorithm regularly failed when the criterion for ending an iteration was to enter a cone surrounding the central path. This may also explain some of the failures of the original barrier method of Fiacco and McCormick [5].

In view of this, another way to attempt to use a factorization more than once was tried. Here the reduced KKT matrix (8) was formed and factored using the dual ordering described in the previous section. If the factorization determined that $H(x, y)$ was indefinite, an appropriate diagonal perturbation (10) was found to assure that the matrix factored was positive definite.

The resulting system

$$\begin{bmatrix} \widehat{H}(x,y) & 0 & -A^T(x) \\ 0 & Y & W \\ A(x) & -I & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta w \\ \Delta y \end{bmatrix} = \begin{bmatrix} -\nabla f(x) + A^T(x)y \\ \mu e - WYe - \Delta W \Delta Y e \\ -h(x) + w \end{bmatrix} \tag{27}$$

was then considered as if it were the quadratic programming problem (21)-(23) and the standard quadratic programming predictor-corrector direction computed as described above. While this direction is generally a very good direction for convex quadratic problems, this is certainly not always the case for nonconvex nonlinear problems. It is shown in [14] that if the current solution is feasible, the solution to (7) is always a descent direction for the merit function (9). That need not be the case for the solution to (27). Thus if the problem is feasible and the predictor-corrector direction is not a descent direction for the merit function, this seems to be a clear indication to abandon the predictor-corrector in favor of the standard direction.

It is the usual case, however, that feasibility and optimality are achieved at the same time. When the problem is infeasible, determining that the predictor-corrector direction is a poor direction is more difficult. To see this, let Δx, Δw, and Δy be the predictor-corrector solution to (27). The algorithm then tries to choose a step size that reduces the merit function, but in order to do this it first tests whether the direction is a descent direction for Ψ, and if not increases β sufficiently to assure that the direction is a descent direction for Ψ. Now Ψ may be written

$$\Psi_{\beta,\mu}(x, w) = b_\mu(x, w) + \frac{\beta}{2}\|\rho(x, w)\|^2$$

where

$$b_\mu(x, w) = f(x) - \mu \sum_{i=1}^{m} \ln(w_i)$$

is the classic barrier function for the problem. Thus

$$\nabla_x \Psi_{\beta,\mu}(x, w) = \nabla_x b_\mu(x, w) + \beta A^T(x)\rho(x, w)$$

and

$$\nabla_w \Psi_{\beta,\mu}(x, w) = \nabla_w b_\mu(x, w) - \beta\rho(x, w).$$

Thus

$$\begin{bmatrix} \nabla_x \Psi_{\beta,\mu} \\ \nabla_w \Psi\beta, \mu \end{bmatrix}^T \begin{bmatrix} \Delta x \\ \Delta w \end{bmatrix} = \Delta x^T \nabla_x b_\mu + \Delta w^T \nabla_w b_\mu + \beta(A\Delta x - \Delta w)^T \rho. \tag{28}$$

But from the third equation of (27)

$$A\Delta x - \Delta w = -\rho,$$

so the final term in (28) becomes $-\beta\|\rho\|^2$. Thus as long as the gradients of the barrier function and the step sizes remain bounded, if the problem is infeasible ($\rho \neq 0$) a β can always be found that ensures that the step direction is a descent direction for the merit function. We have therefore developed a dual strategy to determine whether to use the predictor-corrector step or the original step. If β increases by a factor of 100 at any iteration, we resort to the original

algorithm. If it does not, but we are reducing the step size α significantly, where in the computational results in the final section of the paper, significantly was three or more reductions on any iteration, we also revert to the original algorithm. Once the predictor-corrector has been abandoned, we do not retry it. This algorithm is admittedly ad hoc, and in the future we hope to improve it significantly. Nonetheless, it shows significant gains over the original algorithm, and lends credence to the hope for even greater future improvement.

4. AN ALGORITHMIC FAILURE

To date, no convergence theory is known for the algorithms studied in this paper. As noted previously, convergence results for interior point methods for linear and convex quadratic programming rely on adhering to the central path. As we have seen, this will not work for nonconvex problems, as the central path may not exist. In [4], the authors demonstrate convergence to a KKT point for the merit function

$$\Psi_0(x, w, y) = \|\nabla f(x) - A^T(x)y\|^2 + \|WYe\|^2 + \|h(x) - w\|^2.$$

This merit function has proved unsatisfactory in practice, as it does not discriminate between local minima, maxima, and saddle points [11]. In proving convergence to a KKT point, however, they need the requirement that the coefficient matrix of (7) remain nonsingular. We now show an example where the algorithm of this paper fails when this matrix becomes singular. The problem is a pure root finding problem, and stated in the context of the paper, is

$$\text{minimize } 0$$

$$\text{subject to } x(x-1)(x+1) + 6 = 0.$$

This function has a single root at $x = -2$ and inflection points at $x = \pm\frac{1}{\sqrt{3}}$. The Lagrangian for this problem is

$$L(x, y) = y(x(x-1)(x+1) + 6)$$

and the Newton system

$$\begin{bmatrix} 6xy & 3x^2 - 1 \\ 3x^2 - 1 & 0 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} = \begin{bmatrix} -y(3x^2 - 1) \\ -x(x-1)(x+1) - 6 \end{bmatrix}.$$

The system is clearly singular at either inflection point, but cannot reduce the infeasibility in the constraint. When run with LOQO, the algorithm converges to the correct solution for any $x^{(0)} < -\frac{1}{\sqrt{3}}$, goes to $\frac{1}{\sqrt{3}}$ for $\frac{-1}{\sqrt{3}} < x^{(0)} < \frac{1}{\sqrt{3}}$, and behaves randomly between the two choices for larger values of $x^{(0)}$. Thus any convergence theory will have to address the problem of singularities of the coefficient matrix.

5. COMPUTATIONAL RESULTS

In [14], the basic algorithm described in the introductory section of this paper was tested against MINOS and LANCELOT, and shown to be competitive with these algorithms across a spectrum of test problems. In this paper, we are interested primarily in how a higher order method compares to the basic algorithm, and hence include comparative tests of just these two algorithms. All of our runs were done using LOQO with AMPL [6], which provides analytic first and second partial derivatives. The tests were performed on a R4600 SGI workstation with 160 Mbytes of real memory, 16Kbytes of data cache, and a 133 MHz clock. There are three components to the stopping rule for LOQO: (i) primal feasibility, (ii) dual feasibility, and (iii) lack of duality gap. The default rule declares a solution primal/dual feasible if the relative infeasibility is less than **`1.0e-6`** and declares the problem optimal if, in addition, there are 8 or more digits of agreement between the primal and dual objective functions.

Table 1 shows comparions for LOQO (version 4.03) run both with and without predictor-corrector on the Hock and Schittkowski test set [8]. For these problems we only recorded iteration counts. Times were not significant because all of these problems are very small and solve very quickly. The iteration counts when predictor-corrector was turned off are different from those reported in [14] due to numerous small corrections/modifications between the old version 3.10 and the current version 4.03. The changes include using AMPL's presolve as a default, not requiring the initial point to satisfy simple bounds as a default, and an improved method for finding the perturbation parameter λ. As before, we omitted HS013 from the test as it has no KKT point. LOQO reported that problem HS067 was dual infeasible with both methods apparently due to a complicated embedded fixed point calculation. With the predictor-corrector option turned off, LOQO failed to solve HS085 in the default 200 iterations. Turning off AMPL's presolve corrects this problem.

The total number of iterations with the predictor-corrector option off is 2787 whereas with it on the total number is 2404. While individual problems can vary significantly, overall the predictor-corrector method seems to do very well.

In addition to the Hock and Schittkowski test sets, we compared the two algorithms on three models from the second author's website [12] and 15 models from the CUTE [3] library of test problems. Some of these problems are larger (and therefore more interesting). As the problems get more difficult, the benefits of using the predictor-corrector method become more apparent. The results are contained in the table 2.

While this work is still preliminary, and the test sets relatively small, the results give real hope that the predictor-corrector algorithm can improve interior point methods for nonconvex nonlinear programming significantly. Several comments are in order. On the problem AVION, where predictor-corrector was far inferior to the standard method, the predictor-corrector quickly found an infeasible super-optimal point, from which recovery of feasibility was slow. This phenomenon was a common inefficiency of sequential quadratic programming algorithms, and deserves further study in this context. Also, on SOSQP1, the predictor-corrector method eliminated only two iterations, yet cut execution time in half. This improvement arose from the fact that the standard method required extensive reductions in step size in order to

Name	without pred corr Iters	with pred corr Iters	Name	without pred corr Iters	with pred corr Iters	Name	without pred corr Iters	with pred corr Iters
hs001	33	30	hs040	9	7	hs079	9	8
hs002	19	14	hs041	16	19	hs080	9	9
hs003	11	11	hs042	10	8	hs081	20	17
hs004	8	7	hs043	11	10	hs083	16	15
hs005	10	9	hs044	20	20	hs084	14	18
hs006	11	9	hs045	23	32	hs085	200	64
hs007	14	26	hs046	21	20	hs086	14	11
hs008	9	9	hs047	21	20	hs087	24	19
hs009	10	8	hs048	8	8	hs088	22	22
hs010	15	13	hs049	24	22	hs089	26	40
hs011	13	10	hs050	16	14	hs090	27	25
hs012	10	8	hs051	8	8	hs091	29	23
hs014	11	9	hs052	8	8	hs092	23	20
hs015	24	21	hs053	12	12	hs093	13	11
hs016	18	20	hs054	33	33	hs095	15	11
hs017	27	26	hs055	11	17	hs096	18	38
hs018	18	14	hs056	17	23	hs097	30	19
hs019	17	19	hs057	17	19	hs098	35	19
hs020	24	14	hs059	22	26	hs099	24	19
hs021	12	12	hs060	18	10	hs100	11	10
hs022	9	8	hs061	11	9	hs101	163	28
hs023	18	14	hs062	13	14	hs102	51	33
hs024	13	35	hs063	10	15	hs103	29	24
hs025	27	29	hs064	27	18	hs104	14	10
hs026	15	17	hs065	17	13	hs105	17	21
hs027	55	23	hs066	14	10	hs106	21	25
hs028	8	8	hs067	138	136	hs107	113	46
hs029	10	10	hs068	58	103	hs108	20	25
hs030	9	9	hs069	16	17	hs109	78	96
hs031	13	10	hs070	27	16	hs110	11	17
hs032	23	11	hs071	13	11	hs111	17	24
hs033	12	10	hs072	23	21	hs112	19	11
hs034	16	10	hs073	21	12	hs113	16	11
hs035	10	8	hs074	18	19	hs114	27	20
hs036	15	16	hs075	19	20	hs116	126	157
hs037	11	17	hs076	11	8	hs117	19	17
hs038	44	43	hs077	13	12	hs118	15	15
hs039	15	20	hs078	9	9	hs119	32	20

TABLE 1. Iteration counts on Hock-Schittkowski problems.

Problem	Source	w/o pred corr		w/ pred corr	
		Iters	Time (secs)	Iters	Time (secs)
structure4	rvdb	43	157.0	39	137.4
nb_L2_bessel_real	rvdb	23	459.5	14	205.1
3.mod	rvdb	35	11.0	20	6.4
hadamard (n=12)	cute	13	5.8	10	4.6
hadamard (n=20)	cute	13	17.8	10	13.8
avion	cute	46	0.147	92	0.366
rk23	cute	12	0.011	10	0.010
eigenaco	cute	27	2.23	18	1.76
eigenb2 (N=10)	cute	20	1.73	27	3.46
eigenbco (N=10)	cute	45	5.75	44	5.89
xplin	cute	27	0.065	23	0.058
hager1 (N=1000)	cute	15	5.47	10	2.81
harkerp2 (N=100)	cute	37	2.45	25	1.69
msqrtals (P=10)	cute	17	1.62	16	1.58
sawpath	cute	108	19.3	44	5.2
sosqp1	cute	13	597.0	11	290.0
yao	cute	16	6.66	11	4.65
zigzag (T=10)	cute	46	0.143	50	0.182

TABLE 2. Iteration counts and times for larger problems. Numbers in parentheses represent settable parameters in the model.

assure that the merit function was reduced. With automatic differentiation, as provided by AMPL, first and second derivative calculations can be very cheap compared to function evaluations, so it is not necessarily always true that the time to compute and factor the reduced KKT matrix will dominate execution time. Linear searches can be relatively very expensive.

In summary, much remains to be learned about fully efficient and stable implementations of interior point methods for nonconvex nonlinear programming, but the progress to date is most encouraging.

REFERENCES

[1] I. Adler, N.K. Karmarkar, M.G.C. Resende, and G. Veiga. An implementation of Karmarkar's algorithm for linear programming. *Mathematical Programming*, 44:297–335, 1989.

[2] T.J. Carpenter, I.J. Lustig, J.M. Mulvey, and D.F. Shanno. Higher order predictor-corrector interior point methods with application to quadratic objectives. *SIAM Journal on Optimization*, 3:696–725, 1993.

[3] A.R. Conn, N. Gould, and Ph.L. Toint. Constrained and unconstrained testing environment. http://www.dci.clrc.ac.uk/Activity.asp?CUTE.

[4] A. El-Bakry, R. Tapia, T. Tsuchiya, and Y. Zhang. On the formulation and theory of the Newton interior-point method for nonlinear programming. *J. of Optimization Theory and Appl.*, 89:507–541, 1996.

[5] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrainted Minimization Techniques*. Research Analysis Corporation, McLean Virginia, 1968. Republished in 1990 by SIAM, Philadelphia.
[6] R. Fourer, D.M. Gay, and B.W. Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Scientific Press, 1993.
[7] J. Gondzio. Multiple centrality corrections in a primal dual method for linear programming. *Comp. Opt. and Appl.*, 6:137–156, 1996.
[8] W. Hock and K. Schittkowski. *Test examples for nonlinear programming codes*. Lecture Notes in Economics and Mathematical Systems 187. Springer Verlag, Heidelberg, 1981.
[9] F. Jarre and M. Wechs. Extending mehrotra's corrector for linear programs. Technical Report 219, Mathematische Institute der Universitaet Wuerzburg, 1997.
[10] S. Mehrotra. On the implementation of a (primal-dual) interior point method. *SIAM Journal on Optimization*, 2:575–601, 1992.
[11] D.F. Shanno and E.M. Simantiraki. Interior-point methods for linear and nonlinear programming. In I.S. Duff and G.A. Watson, editors, *The State of the Art in Numerical Analysis*, pages 339–362. Oxford University Press, New York, 1997.
[12] R.J. Vanderbei. AMPL models. http://www.sor.princeton.edu/˜rvdb/ampl/nlmodels/.
[13] R.J. Vanderbei. LOQO: An interior point code for quadratic programming. Technical Report SOR 94-15, Princeton University, 1994. Revised 11/30/98. To appear in *Optimization Methods and Software*.
[14] R.J. Vanderbei and D.F. Shanno. An interior-point algorithm for nonconvex nonlinear programming. Technical Report SOR-97-21, Statistics and Operations Research, Princeton University, 1997. To appear in *Computational Optimization and Applications*.

DAVID F. SHANNO, RUTGERS UNIVERSITY, NEW BRUNSWICK, NJ

ROBERT J. VANDERBEI, PRINCETON UNIVERSITY, PRINCETON, NJ