Large Scale Optimization: State of the Art, pp. 319–338
W. W. Hager, D. W. Hearn and P.M. Pardalos, Editors

A Numerical Comparison of Barrier and Modified-Barrier Methods For Large-Scale Bound-Constrained Optimization

Stephen G. Nash, R. Polyak, and Ariela Sofer
George Mason University, Fairfax, VA 22030 USA

Abstract

When a classical barrier method is applied to the solution of a nonlinear programming problem with inequality constraints, the Hessian of the barrier function becomes increasingly ill-conditioned as the solution is approached. As a result, it may be desirable to consider alternative numerical algorithms. We compare the performance of two methods motivated by barrier functions. The first is a stabilized form of the classical barrier method, where a numerically stable approximation to the Newton direction is used when the barrier parameter is small. The second is a modified barrier method where a barrier function is applied to a shifted form of the problem, and the resulting barrier terms are scaled by estimates of the optimal Lagrange multipliers. When implemented appropriately the condition number of the Hessian of the modified barrier function remains bounded as the solution to the constrained optimization problem is approached. Both of these techniques can be used in the context of a truncated-Newton method, and hence can be applied to large problems, as well as on parallel computers. In this paper, both techniques are applied to problems with bound constraints and we compare their practical behavior.

Keywords: nonlinear programming, barrier method, modified barrier method, Newton's method, truncated-Newton method, large-scale optimization.

1 Introduction

We will examine the solution of nonlinear programming problems of the form

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

Here $x = (x_1, \ldots, x_n)^T$ and the functions f and $\{ c_i \}$ will be assumed to be twice continuously differentiable. We have in mind cases where n is large.

The methods we will consider for solving (1) will be based on classical barrier functions. The constrained problem is converted to a sequence of unconstrained problems. If the logarithmic barrier function is used, then the unconstrained problems have the form

$$\beta(x, \mu) = f(x) - \mu \sum_{i=1}^{m} \ln(c_i(x)),$$

involving a "barrier parameter" $\mu > 0$. If $x^*(\mu)$ denotes a minimizer of $\beta(x, \mu)$ then, under appropriate conditions, it can be shown that (as $\mu \to 0$) any limit point x^* of the sequence $\{ x^*(\mu) \}$ is a solution of (1) [FiaM68]. In addition, the associated Lagrange multiplier estimates converge to the Lagrange multipliers at x^*.

It is well known that the Hessian of the barrier function becomes increasingly ill-conditioned as $\mu \to 0$ and a solution to (1) is approached. (This will be discussed in more detail in Section 3.) More specifically, if k constraints are binding at x^* and $0 < k < n$ then

$$\lim_{\mu \to 0^+} \mathrm{cond}(\nabla^2 \beta(x^*(\mu), \mu)) = +\infty.$$

Thus the classical barrier method "breaks down" as the method converges to the solution of the original constrained problem.

We will examine two approaches that avoid this "structural" ill-conditioning (i.e., the ill-conditioning associated with the method, as distinct from the conditioning of the underlying optimization problem). The first uses a numerically stable approximation to the Newton direction for the classical barrier function [NasS93a]. The second uses a modified barrier function, where the resulting barrier terms are scaled by estimates of the Lagrange multipliers at the solution [Poly92]. When appropriately implemented the Hessian of the modified barrier function can be shown to have bounded condition number. Both techniques solve a sequence of unconstrained optimization problems involving a (possibly modified) barrier function, for a sequence of barrier parameters and/or Lagrange multiplier estimates.

In this paper, each of these unconstrained problems will be solved using a truncated-Newton method. In this method, the Newton equations for a search direction are solved approximately using the conjugate-gradient method. Why choose a truncated-Newton method? It is a Newton-type method, that requires only first derivatives (although second derivatives may be utilized if desired); it has low storage costs; it can be adapted to solve nonconvex problems; and it vectorizes well. Thus method reduces the costs of Newtons method while maintaining rapid convergence, and is therefore suitable for large-scale problems. In practice the method has proven to be robust, effective and competitive on a wide set of unconstrained minimization problems.

The stabilized barrier method is the same as in [NasS93a], although it is tested here on a larger set of problems (and using a different computer). The modified

barrier method software is new, although it was obtained by modifying the software for the stabilized barrier method. Because much of the software for the two methods is the same, we believe that this gives a clearer comparison of the properties of the two methods.

2 The Truncated-Newton Method

In both the modified barrier method and the stabilized barrier method, the unconstrained subproblems will be solved using a modifed version of the truncated-Newton software described in [NasN91]. A summary of this method will be given here, as applied to an unconstrained problem

$$\underset{x}{\text{minimize}}\, f(x).$$

The notation $\nabla f = \nabla f(x)$ is used for the gradient of f evaluated at a point x.

Given some initial guess x_0, at the j-th iteration the new estimate $\hat{x}$ of the solution is given by

$$\hat{x} = x + \alpha p.$$

The search direction p must satisfy $p^T \nabla f < 0$ (i.e., it is a descent direction for f at the point x).

The step length $\alpha > 0$ is chosen to guarantee that $f(\hat{x}) < f(x)$, along with other conditions designed to guarantee convergence to a local minimizer [OrtR70]. The particular line search algorithms used are discussed below. performance.

The search direction p is computed as an approximate solution of the Newton equations

$$(\nabla^2 f)p = -\nabla f \tag{2}$$

where $\nabla^2 f \equiv \nabla^2 f(x)$ is the Hessian matrix of second derivatives at the current point x. The approximate solution is obtained by applying the conjugate-gradient method to (2). This iterative method is "truncated" before the exact solution is obtained. On parallel computers, a block conjugate gradient method could be used to solve (2) [NasS91], resulting in a parallel barrier method. This idea has been applied to bound-constrained problems for the stabilized barrier method [Nash92].

The conjugate-gradient method corresponds to minimizing the quadratic model $Q(p) = \frac{1}{2} p^T \nabla^2 f p + p^T \nabla f$ as a function of p over a sequence of subspaces of increasing dimension. These are called the Krylov subspaces.

The truncated-Newton software used here includes automatic preconditioning strategies designed to accelerate convergence of the conjugate-gradient method. These were not modified in the computational tests used in this paper, because of the special form of the bound constraints. For problems with more general constraints, it is likely that the preconditioners would have to be adjusted to take into account the

special structure of the barrier subproblems. Techniques for doing this are discussed in [NasS93b].

3 The Stabilized Barrier Method

The discussion here is adapted from [NasS93a], and presents a summary of the stabilized barrier method. For a more complete discussion, the reference should be consulted.

We will assume that a strictly feasible initial guess of the solution has been provided. For problems with bound constraints, such a point can be easily found. In addition, we make the following standard assumptions: (a) the feasible set is compact and has a non-empty interior; (b) a solution x^* lies in the closure of the interior of the feasible region; (c) x^* is a regular point of the constraints (i.e., the gradients of the active constraints at x^* are linearly independent) which satisfies the second-order sufficiency conditions for optimality (see [FiaM68]).

The logarithmic barrier method converts the problem (1) to a sequence of unconstrained problems:

$$\underset{x}{\text{minimize}}\, \beta(x,\mu) = f(x) - \mu \sum_{i=1}^{m} \ln(c_i(x)), \tag{3}$$

for a sequence of positive barrier parameters $\mu \to 0$. Let $x^*(\mu)$ denote an unconstrained minimizer of $\beta(x,\mu)$. Under quite mild conditions it can be shown that any limit point x^* of the sequence $x^*(\mu)$ is a solution of (1). Furthermore if we define

$$\lambda_i(\mu) \equiv \mu / c_i(x^*(\mu)),$$

then as $x^*(\mu) \to x^*$, $\lambda(\mu) \to \lambda^*$, where λ^* is the vector of Lagrange multipliers corresponding to x^* [FiaM68].

The Newton direction for the barrier subproblem (3) at the point x is obtained by solving

$$Bp = -b,$$

where b and B are the gradient and Hessian, respectively, of the logarithmic barrier function:

$$\begin{aligned} b &= \nabla f - \mu \sum_{i=1}^{m} \frac{\nabla c_i}{c_i}, \\ B &= \nabla^2 f - \mu \sum_{i=1}^{m} \frac{\nabla^2 c_i}{c_i} + \mu \sum_{i=1}^{m} \frac{\nabla c_i \nabla c_i^T}{c_i^2}. \end{aligned} \tag{4}$$

(To simplify the formulas, f is written for $f(x)$, etc.) If we define $\lambda_i = \mu/c_i$, then (4) can be expressed in the form

$$\begin{aligned} b &= \nabla f - \sum_{i=1}^{m} \lambda_i \nabla c_i, \\ B &= \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i + \frac{1}{\mu} \sum_{i=1}^{m} \lambda_i^2 \nabla c_i \nabla c_i^T. \end{aligned} \tag{5}$$

The final term in (5) reveals the ill-conditioning in the barrier subproblem. If a constraint is active at the solution, and its corresponding Lagrange multiplier is non-zero, then the ratio $\lambda_i^2/\mu \to \infty$ as $\mu \to 0$. Thus the Hessian matrix becomes progressively more ill conditioned as the solution (see [Murr71]).

The stabilized barrier method avoids this ill conditioning by using an approximation to the Newton direction for the barrier function. This approximation differs from the Newton direction by terms of $O(\mu)$ and so becomes more accurate as $\mu \to 0$. The approximation is obtained by examining the range- and null-space components of the search direction, defined in terms of a "working set" of constraints, analogous to the working set used in an active-set method for constrained optimization (see, for example, [GiMW81]). The approach we propose does not require that the Hessian of the barrier be formed explicitly. A different approach that avoids the ill conditioning but that requires explicit matrix factorizations is described in [Wrig92].

To develop the formulas for the search direction, we define $\mathcal{I}$ to be the index set of those constraints that contribute to the ill conditioning of the Hessian. This set is a prediction of the set of constraints that are binding at the solution of (1). Let N be the Jacobian matrix of the constraints in $\mathcal{I}$, and assume that N has full rank. (The columns of N consist of the vectors ∇c_i.) We define $D = \mathrm{diag}(\lambda_{\mathrm{i}}^2, \mathrm{i} \in \mathcal{I})$, and choose Z as a basis for the null space of N^T. Let $N^{\#}$ be a pseudo-inverse for N. (For bound-constrained problems, the columns of N and Z are just columns of the identity matrix.) Finally, define

$$H = \nabla^2 f - \sum_{i=1}^{m} \lambda_i \nabla^2 c_i + \frac{1}{\mu} \sum_{i \notin \mathcal{I}} \lambda_i^2 \nabla c_i \nabla c_i^T,$$

i.e., the "good" part of the Hessian B, omitting the ill-conditioned terms.

Using these definitions the Newton direction can be approximated via

$$p \approx p_1 + \mu p_2,$$

where

$$\begin{aligned} p_1 &= -Z(Z^T H Z)^{-1} Z^T b, \\ \hat{\lambda} &= N^{\#}(H p_1 + b), \\ p_2 &= -(N^{\#})^T D^{-1} \hat{\lambda}. \end{aligned} \tag{6}$$

These formulas correspond to an $O(\mu)$ approximation to the Newton direction. (A related stabilized formula for the search direction was derived in [Murr71].)

The formulas (6) only require $(Z^THZ)^{-1}$. In our algorithm this is implemented by applying the conjugate-gradient method to

$$Z(Z^THZ)^{-1}Z^Tp_1 = -b,$$

with the iteration truncated as in the unconstrained case. The costs of finding the search direction in this approach are comparable to those of a naive barrier method that does not deal with the ill conditioning. The approximate direction obtained using the formulas (6), together with a truncated conjugated-gradient iteration, can be shown to be a descent direction for the barrier function under appropriate assumptions.

A number of computational enhancements were used to improve the performance of the stabilized barrier method. These are discussed briefly in Section 5.

4 The Modified Barrier Method

We now describe the modified barrier method for the constrained problem (1). An extensive discussion of the theory of modified barrier methods can be found in [Poly92].

At each major iteration of the modified barrier method the unconstrained problem

$$\underset{x}{\text{minimize}}\, \mathcal{M}(x, \lambda, \mu) \tag{7}$$

is solved where

$$\mathcal{M}(x, \lambda, \mu) = f(x) - \mu \sum_{i=1}^{m} \lambda_i \psi(\mu^{-1} c_i(x) + 1),$$

and the solution $\hat{x}$ is used to update $\{ \lambda_i \}_{i=1}^m$ via

$$\hat{\lambda}_i = \lambda_i \psi'(\mu^{-1} c_i(\hat{x}) + 1). \tag{8}$$

The parameters $\{ \lambda_i \}$ are estimates of the Lagrange multipliers at the solution x^*. The function ψ is a monotone, strictly concave, and twice continuously differentiable function defined on the interval $(0, +\infty)$; one possible choice is $\psi(\cdot) = \ln(\cdot)$, although our algorithm will use a more complicated definition of ψ. It is also possible to use the inverse function $\psi(\cdot) = 1/(\cdot)$ although this choice is not tested here.

If, for example, $\psi(\cdot) = \ln(\cdot)$, then the feasible region for (1) is equivalent to the set

$$\left\{ x : \mu\psi(\mu^{-1} c_i(x) + 1) \geq 0 \right\}.$$

Thus the modified barrier function is the classical Lagrangian for the problem (1) with the constraints expressed in this equivalent form. The use of the barrier term

$$\psi(\mu^{-1} c_i(x) + 1)$$

corresponds to perturbing the constraints so that they have the form

$$c_i(x) \geq -\mu.$$

This represents an expansion of the feasible region. Hence the implied "feasible region" for the modified barrier subproblem varies with the barrier parameter μ.

Unlike the classical logarithmic barrier function, the modified barrier function and its derivatives exist at a solution x^* for any positive barrier parameter μ. In particular, if λ^* is the vector of Lagrange multipliers corresponding to x^*, and if $\psi(\cdot) = \ln(\cdot)$, then the modified barrier function has the following properties for any $\mu > 0$:

P1. $\quad \mathcal{M}(x^*, \lambda^*, \mu) = f(x^*)$

P2. $\quad \nabla_x \mathcal{M}(x^*, \lambda^*, \mu) = \nabla f(x^*) - \sum_{i=1}^{m} \lambda^* \nabla c_i(x^*) = 0$

P3. $\quad \nabla_x^2 \mathcal{M}(x^*, \lambda^*, \mu) = \nabla^2 f(x^*) - \sum_{i=1}^{m} \lambda_i^* \nabla^2 c_i(x^*) + \mu^{-1} \nabla c(x^*) \mathrm{diag}(\lambda^*)^2 \nabla \mathrm{c}(\mathrm{x}^*)^{\mathrm{T}}$

When the problem is a a convex program, it follows from P2 that

$$\text{P4.} \qquad x^* = \arg\min \{ \mathcal{M}(x, \lambda^*, \mu) \} \quad \text{for any } \mu > 0.$$

This means that if the optimal Lagrange multipliers λ^* are known, one can solve the constrained problem (1) using a single unconstrained optimization problem regardless of the value of the barrier parameter. Moreover, if the constrained optimization problem is nonconvex but the second-order sufficiency and strict complementarity conditions are satisfied at x^* then there exists a $\bar{\mu}$ and a $\bar{\rho} > 0$ such that:

$$\text{P5.} \qquad \min \mathrm{eig} \nabla_x^2 \mathcal{M}(x^*, \lambda^*, \mu) \geq \bar{\rho} \quad \text{for } \mu < \bar{\mu}.$$

Thus it is again possible to solve (1) using a single unconstrained optimization problem of the form (7) provided that the barrier parameter is sufficiently small. Of course, in practice only a local minimizer may be found.

Polyak [Poly92] has shown that if the initial Lagrange multipliers are positive, and the barrier parameters are below some threshhold value $\bar{\mu}$, then the method converges. Furthermore, for sufficiently small μ, the successive iterates satisfy

$$\max \left\{ \|\hat{x} - x^*\|, \left\|\hat{\lambda} - \lambda^*\right\| \right\} \leq c\mu \|\lambda - \lambda^*\|. \tag{9}$$

The constant $c > 0$ is independent of $\mu \leq \bar{\mu}$.

For a convex programming problem it is possible to prove a stronger result. Under mild conditions on the primal and dual feasible regions the modified barrier method converges for any *fixed* positive value of the barrier parameter μ, provided that the initial vector of Lagrange multipliers is positive (see [JenP92]).

The result (9) shows that the modified barrier method converges at a superlinear rate if the barrier parameter is changed from subproblem to subproblem in such a way

that $\mu \to 0$. However it is not necessary that $\mu \to 0$ in order to achieve convergence; it is only necessary that μ be reduced below the threshhold value $\bar{\mu}$. Thus the condition number of the Hessian of the modified barrier function can remain bounded as the solution is approached, unlike in the classical case.

On practical problems, it is not possible to know *a priori* whether the initial parameter chosen is indeed below the threshhold $\bar{\mu}$, a general-purpose code for solving (1) must also include some mechanism for reducing the barrier parameter. However some caution is required. If a solution $\hat{x}(\mu)$ to a modified barrier subproblem has been found, and μ is reduced to a new value $\hat{\mu}$ it is possible that $\hat{x}(\mu)$ will be "infeasible" for the new subproblem:

$$c_i(\hat{x}(\mu)) \not\geq -\hat{\mu}.$$

Suppose that the function ψ is chosen as $\psi(\cdot) = \ln(\cdot)$. Then if $\hat{\mu} < \mu$ and $c_i(\hat{x}) < 0$ it is possible that

$$\psi(\hat{\mu}^{-1} c_i(\hat{x}) + 1) = \ln(\hat{\mu}^{-1} c_i(\hat{x}) + 1)$$

might be undefined. This limits the flexibility of the modified barrier method (it limits how quickly μ can be reduced) and it can greatly complicate software for this algorithm, particularly if the constraints are nonlinear [BreS93].

For this reason we have chosen to use a more elaborate definition of the function ψ, a definition that varies with the value of μ. In our implementation we use a modification that has been suggested in [BeTY92]. Let $t = c_i(x)$. If $t \geq -\mu/2$ then we define

$$\psi(\mu^{-1} t + 1) = \ln(\mu^{-1} t + 1).$$

If $t < -\mu/2$ then we define

$$\psi(\mu^{-1} t + 1) = q(t)$$

where $q(t)$ is a quadratic function for which $q(-\mu/2)$, $q'(-\mu/2)$, and $q''(-\mu/2)$ match the corresponding values for the logarithm function at the point $t = -\mu/2$. Since the quadratic function does not have a singularity at $-\mu$ (or at any other point), the barrier parameter can be reduced at any desired rate without worrying whether the modified barrier function will become undefined or singular.

Our software for the modified barrier algorithm was obtained by adapting the software for the stabilized barrier method. The underlying unconstrained optimization method is the same truncated-Newton method. More specific details (chosen as a result of considerable numerical testing) are discussed in Section 5.

5 Implementation

A number of computational enhancements were used to improve the performance of the stabilized barrier method. We give a brief description of these enhancements and discuss their effect when implemented within a modified barrier method.

5.1 The Line Search

Because the logarithmic barrier function has a singularity at the boundary of the feasible region, standard line search algorithms based on low-order polynomial interpolation may not be effective. For example, in implementing an inverse cubic interpolation line search we found that an usually large proportion (often more than 50%) of the overall computational effort was spent within the line search. Replacing this line search by a Armijo-type strategy reduced the fraction of time spent in the line search but increased the overall computational effort substantially.

For this reason we implemented a special line search [MurW76] devised specifically for the logarithmic barrier function. This line search approximates the barrier function along the search direction with a one-dimensional function consisting of a quadratic term plus a logarithmic singularity. We have found this line search to be effective when implemented within a classical barrier method. For example, on a set of problems tested in [NasS93], the special line search led to a 27% reduction in the overall computational effort.

The special line search was not as beneficial when implemented within a modified barrier method. This may be due to the fact that our elaborate definition of ψ no longer has a logarithmic singularity. The line search currently implemented in our software is a standard line search for unconstrained minimization based on inverse cubic interpolation with an acceptance test based on a Wolfe condition (the "default" line search for the truncated-Newton method).

5.2 Extrapolation

A (classical) barrier method can be improved significantly by extrapolation. This technique uses the solutions of the subproblems for previous barrier parameters to fit a low-order polynomial to the barrier trajectory. The polynomial is then used to predict the solution of the subproblem for the new barrier parameter. This provides a better initial guess for the new problem.

Our own experience indicates that substantial gains may be obtained by using quadratic extrapolation, and that modest additional gains may be obtained by using cubic interpolation instead. The stabilized barrier software uses cubic extrapolation.

Our attempts to accelerate the modified barrier using either linear, quadratic or cubic extrapolation were not successful. The reason is that the solutions to the modified barrier subproblems do not lie on a simple trajectory parameterized by μ, as is true for the classical barrier function. Thus in the current code, no extrapolation is used to obtain the initial guess for a new subproblem, and the solution to the previous subproblem is used as an initial guess without modification.

5.3 Initializing the barrier parameter

The selection of the initial barrier parameter can have a dramatic effect on the running time of the algorithm. A parameter that is too small may cause the subproblem to be ill-conditioned and therefore difficult to solve. A parameter that is too large may require the solution of too many subsequent subproblems.

The best initialization scheme that we found for the stabilized barrier method is a heuristic that attempts to find the barrier parameter corresponding to the point on the barrier trajectory which is "closest" to the initial point. The same scheme does not appear to be effective for the modified barrier method: the resulting initial parameter tends to be "too large." Better results were obtained by setting the initial barrier parameter to a relatively small value.

5.4 Preconditioning

To be effective, a truncated-Newton method must use preconditioning. The truncated-Newton software uses a preconditioner based on a limited-memory quasi-Newton formula obtained from consecutive truncated-Newton iterations, which in turn is scaled by a diagonal approximation to the Hessian obtained from the conjugate gradient iterations. The stabilized barrier software uses the final preconditioner from one subproblem as the initial preconditioner for the next subproblem. The modified barrier method uses the same strategy.

5.5 Customized matrix-vector product

The stabilized barrier method uses a customized matrix-vector product for the conjugate-gradient iteration that isolates the terms associated with the working set $\mathcal{I}$. This is necessary so that rounding errors from the ill- conditioned terms do not contaminate the well-conditioned terms in the Hessian, and hence destroy the effects of the stabilized approximation to the Newton direction.

If B denotes the Hessian of the barrier function then the product Bu is computed via the formula

$$Bu = (\nabla^2 f)u - \mu \sum_{i=1}^{m} \frac{(\nabla^2 c_i)u}{c_i} + \mu \sum_{i=1}^{m} \frac{(\nabla c_i^T u)\nabla c_i^T}{c_i^2}.$$

The terms $(\nabla^2 f)u$ and $(\nabla^2 c_i)u$ are computed via finite differencing:

$$(\nabla^2 f)u \approx \frac{\nabla f(x + hu) - \nabla f(x)}{h},$$

where h is (approximately) the square root of the machine precision. It is not safe to apply finite differencing directly to Bu because of the singularity of the logarithmic function. The final summation in the formula for Bu is computed straightforwardly

from the formulas above. When the stabilized formulas for the search direction are used, the product Hu must be computed. This is done in the same way, except that the ill-conditioned terms are omitted from the final summation.

The modified barrier uses a similar approach, except applied to the Hessian of the modified barrier function.

6 Computational Tests

In this section we compare the modified barrier method and the stabilized barrier method on a set of test problems with bound constraints.

Many of our test problems are derived from a set of unconstrained optimization problems; see Table 1. For more detailed information about problems 1–52, see [NasN91]. Problems 54 and 55 are from [CoGT88]. The final two problems are from release 2 of the Minpack-2 collection [AvCM91]. They are DPJBFG (pressure in a journal bearing) and DEPTFG (elastic-plastic torsion). These are the only two minimization problems in this collection which have bound constraints that are binding at the solution. For problem DPJBFG we set $nx = ny = \sqrt{n}$, $ecc = 0.1$, and $b = 10$. For problem DEPTFG we set $nx = ny = \sqrt{n}$, and $c = 5$.

The constrained problems 1–55 are as in [NasS93a]. In each case, we first solve the corresponding unconstrained problem, computing $\hat{x}$ satisfying

$$\|g(\hat{x})\|_\infty \leq 10^{-5}(1 + |f(\hat{x})|)$$

using the standard initial point x_0. Lower and upper bounds are then derived from $\hat{x}$. If i is odd then

$$-100 \leq x_i \leq 100;$$

if i is a multiple of 4 then

$$(\hat{x})_i + 0.1 \leq x_i \leq (\hat{x})_i + 10.0;$$

if i is even but not a multiple of 4 then

$$(\hat{x})_i - 10.0 \leq x_i \leq (\hat{x})_i - 0.1.$$

Then a strictly feasibly initial point for the barrier method is generated. If $(x_0)_i < \ell_i$ then $(x_0)_i = \ell_i + 0.5$; if $(x_0)_i > u_i$ then $(x_0)_i = u_i - 0.5$. If $(x_0)_i = \ell_i$ then $(x_0)_i = \ell_i + 10^{-4}$; if $(x_0)_i = u_i$ then $(x_0)_i = u_i - 10^{-4}$. Then x_0 is used as the initial point for the barrier method.

The algorithms were programmed in Fortran 77 and the runs were made using double precision on an IBM 320H RISC workstation. The "stabilized" algorithm uses the stabilized formula for the Newton direction when μ is small; the "modified" algorithm uses the modified barrier method. The two methods incorporate the enhancements described in Section 5.

Both methods compute a search direction using a conjugate-gradient iteration terminated as in [NasS90a], using a rule based on the value of the quadratic model with tolerance 0.5. Both barrier methods were terminated when the norm of the complementary slackness vector (scaled by $1 + |f(x)|$) was less than $\epsilon_1 = 10^{-8}$, and when the norm of the Lagrangian gradient (also scaled by $1 + |f(x)|$) was less than $\epsilon_2 = 10^{-5}$. In addition, we required that the solution from the modified barrier method not be infeasible with respect to any constraint by more than $\epsilon_1 = 10^{-8}$.

We list here a number of implementation details for the stabilized barrier method. For further information, see [NasS93a].

- The line search was terminated using an Armijo-type test with parameter $\eta = 0.2$.
- The barrier parameter was updated using $\mu_{k+1} = \mu_k/10$.
- The truncated-Newton method (for a given μ) was terminated when the norm of the gradient (scaled as above) was less than $\epsilon_3 = 10^{-3}$, and when the smallest Lagrange multiplier estimate was greater than $-\epsilon_4$, where $\epsilon_4 = 10^{-6}$.
- The stabilized formula for the Newton direction was invoked when the norm of the scaled complementary slackness vector was less than $\epsilon_5 = 10^{-4}$.

We made many test runs using the modified barrier method, and some of the more interesting ones are described below. However, we will only be providing detailed results for the best of these runs, for which the following parameter settings were used:

- The line search was terminated using a Wolfe-type test with parameter $\eta = 0.25$.
- the initial barrier parameter was the same for all test problems, $\mu_0 = 10^{-3}$; the barrier parameter was updated using $\mu_{k+1} = \mu_k/2$;
- the initial Lagrange multiplier estimates were chosen to be $\lambda_i = 1, i = 1, \ldots, m$;
- for the first subproblem, the truncated-Newton method was terminated when the norm of the scaled gradient was less than $\epsilon_3 = 10^{-3}$;
- for subsequent subproblems, the truncated-Newton method was terminated when the norm of the scaled gradient was less than $\epsilon_3 = 10^{-6}$;

For a particular algorithm, a single set of parameter settings was used to solve all of the test problems. The algorithms were not "tuned" to particular problems.

The detailed results are given in Table 2. The table records the costs of running the barrier method, but not the costs associated with determining the initial point and the bounds (that is, the costs of solving the initial unconstrained problem are ignored). An entry in the table consists of four numbers: "it" (the total number of outer iterations), "ls" (the number of gradient evaluations used in the line search), "cg" (the number of gradient evaluations used in the inner iteration to compute the Hessian-vector products), and "total" (the sum of "ls" and "cg").

The results in Table 2 indicate that the modified barrier method performs notably better than the stabilized barrier method on these problems. The modified barrier

method requires only 74% as many truncated-Newton iterations, and only 68% as many gradient evaluations. In examining individual problems it is seen that the stabilized barrier method only beats the modified barrier method on 9 of the 33 problems: problems 1 ($n = 100, 1000$), 12, 42, 49, 54, 102 ($n = 100, 1024$), 105 ($n = 100$). We should emphasize that these individual results are a by-product of our desire to use a single set of parameter settings for all test problems. By seeking parameter settings that minimize the grand total for the entire test set, the behavior of the method on individual problems can deteriorate. In particular, by fine-tuning the parameters for these problems it is possible to obtain much better performance from the modified barrier method (at the cost of poorer performance on other problems).

For the other computational tests of the modified barrier method we will only list the totals for the four table entries. Note that for the best version of the method that we were able to find, the totals were

$$(1592 \quad 3361 \quad 7613 \quad 10974)$$

We experimented with solving the first subproblem both more and less accurately, but this was less effective. When the initial subproblem was terminated when the norm of the scaled gradient was less than $\epsilon_3 = 10^{-2}$ (instead of $\epsilon_3 = 10^{-3}$) then the totals were:

$$(1748 \quad 3463 \quad 8705 \quad 12168)$$

Similar results were obtained when the first subproblem was terminated after a fixed number (6) truncated-Newton iterations. When the first subproblem was solved to "full" accuracy ($\epsilon_3 = 10^{-6}$), then the results were worse:

$$(1608 \quad 3480 \quad 8519 \quad 11999)$$

The overall convergence of the modified barrier method seems to be driven by the accuracy of the multipliers. By solving the first subproblem less accurately we hope to get better initial Lagrange multiplier estimates at relatively low expense. If the first subproblem is solved too crudely, however, it is possible to obtain poor estimates of the Lagrange multipliers. Solving the first subproblem to full accuracy can also be wasteful, though, because it does not make sense to accurately solve a subproblem with arbitrary Lagrange multipliers ($\lambda_i = 1$).

We experimented with "more sophisticated" choices of the initial Lagrange multiplier estimates, trying to use gradient and residual information at the initial point x_0 to compute first-order multiplier estimates. The results were poor (with grand totals near 20,000).

In another set of experiments we varied the choice of the initial barrier parameter μ_0 from the value used above ($\mu_0 = 10^{-3}$), but with the other parameter settings unchanged. The following totals were obtained with $\mu_0 = 10^{-1}$:

$$(2367 \quad 3928 \quad 10005 \quad 13933)$$

with $\mu_0 = 10^{-2}$:

$$(1875 \quad 3364 \quad 8650 \quad 12014)$$

with $\mu_0 = 10^{-4}$:

$$(2002 \quad 6836 \quad 12151 \quad 18987)$$

We also attempted to define μ_0 adaptively based on gradient information at x_0, as was done for the barrier function. This attempt failed, with grand totals near 20,000.

Tests were also performed where the subproblems were solved less accurately (using $\epsilon_3 > 10^{-6}$). These were not successful. The modified barrier method seems to require accurate Lagrange multiplier estimates, and these cannot be obtained without solving the subproblems accurately.

Finally we experimented with different rates of reducing the barrier parameter. A surprisingly successful strategy on a large number of the test problems was to leave the barrier parameter fixed at $\mu = 10^{-3}$ for all subproblems. However, this strategy behaved poorly on a few subproblems, making it noncompetitive overall. Reducing the barrier method more rapidly did not work well, in contrast to our experience with the stabilized barrier method. We think that it might be possible to reduce the barrier parameter more quickly if some form of extrapolation procedure could be found for the modified barrier method.

The strategies for running the two methods effectively are different. In the stabilized barrier method a larger number of subproblems are used, each one solved coarsely, and the barrier parameter is reduced quickly. Extrapolation techniques and other enhancements are then used to safeguard and accelerate the method. For the modified barrier method, fewer subproblems are used, each one is solved accurately, the barrier parameter is reduced slowly (and frequently need not be reduced at all). In Tables 3 and 4 these points are illustrated, with the two methods being applied to problem 51 with $n = 1000$.

For completeness we also illustrate the performance of a "naive" barrier method on the same problem. The "naive" barrier method is simply a barrier method without the special enhancements. It uses a line search based on inverse cubic interpolation, it does not use extrapolation, special initialization of μ, or the 1-inverse formula, and it does not save the preconditioners from one subproblem to the next. The results for this method are shown in Table 5.

7 Conclusions

We have compared the performance of a stabilized barrier method with the performance of a modified-barrier method. Our past experience indicates that the stabilized barrier method is a robust and effective method for solving bound-constrained problems. Our software for the stabilized barrier method is a result of much testing and enhancement, and represents a considerable improvement over "naive" barrier

techniques. In contrast, our software for the modified barrier method is less sophisticated. Nevertheless, its performance is superior to the stabilized barrier method on the bound-constrained problems that we have tested. We expect that we may obtain even better performance with further testing and enhancement. This suggests that modified barrier methods are a promising tool for solving large nonlinear programming problems.

8 Acknowledgments

Stephen G. Nash and Ariela Sofer were partially supported by National Science Foundation grant DDM–9104670. R. Polyak was partially supported by NASA contract NAG3–1397 and National Science Foundation grant DMS–9300962.

References

[AvCM91] B.M. Averick, R.G. Carter, and J.J. Moré, *The Minpack-2 Test Problem Collection (Preliminary Version),* Report ANL/MCS–TM–150, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne IL (1991).

[BeTY92] A. Ben-Tal, M. Tsibulevskii, and I. Yusefovich, *Modified barrier methods for constrained and minimax problems,* Technical report, Optimization Laboratory, Faculty of Industrial Engineering and Management, Technion (Israel Institute of Technology) (1992).

[BreS93] M.G. Breitfeld and D.F. Shanno, *Preliminary computational experience with shifted log-barrier functions for large-scale nonlinear programming,* Technical Report, Rutgers Center for Operations Research, Rutgers University, New Brunswick NJ (1993).

[CoGT88] A.R. Conn, N.I.M. Gould, and P.L. Toint, *Testing a class of methods for solving minimization problems with simple bounds on the variables,* Math. Comp. 50 (1988) pp. 399–430.

[FiaM68] A.V. Fiacco and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques,* John Wiley and Sons, New York, 1968. (This book has been reprinted by SIAM, Philadelphia, 1990).

[GiMW81] P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization,* Academic Press, New York, 1981.

[JenP92] D.L. Jensen and R. Polyak, *The convergence of a modified barrier method for convex programming,* Technical Report 18570, IBM Thomas J. Watson Research Center, Yorktown Heights, NY (1992).

[Murr71] W. Murray, *Analytical expressions for the eigenvalues and eigenvectors of the Hessian matrices of barrier and penalty functions*, J. Opt. Th. App. 7 (1971) pp. 189–196.

[MurW76] W. Murray and M.H. Wright, *Efficient linear search algorithms for the logarithmic barrier function*, Report SOL 76–18, Dept. of Operations Research, Stanford University, Stanford CA (1976).

[Nash92] S.G. Nash, *A parallel barrier algorithm for large-scale bound-constrained optimization*, Technical Report, Operations Research and Engineering Department, George Mason University, Fairfax VA (1993).

[NasN91] S.G. Nash and J. Nocedal, *A Numerical Study of the Limited Memory BFGS Method and the Truncated-Newton Method for Large Scale Optimization*, SIAM J. Opt. 1 (1991) pp. 358–372.

[NasS90a] S.G. Nash and A. Sofer, *Assessing a search direction within a truncated-Newton method*, Operations Research Letters 9 (1990) pp. 219–221.

[NasS91] S.G. Nash and A. Sofer, *A general-purpose parallel algorithm for unconstrained optimization*, SIAM J. Opt 1 (1991) pp. 530–547.

[NasS93a] S.G. Nash and A. Sofer, *A barrier method for large-scale constrained optimization*, ORSA Journal on Computing 5 (1993) pp. 40–53.

[NasS93b] S.G. Nash and A. Sofer, *Preconditioning of reduced matrices*, Technical Report, Operations Research and Engineering Department, George Mason University, Fairfax VA (1993).

[OrtR70] J.M. Ortega and W.C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, London and New York, 1970.

[Poly92] R. Polyak, *Modified barrier functions (theory and methods)*, Mathematical Programming 54 (1992) pp. 177–222.

[Wrig92] M.H. Wright, *Determining subspace information from the Hessian of the barrier function*, Numerical Analysis Manuscript 92–02, AT& T Bell Laboratories, Murray Hill NJ (1992).

Problem	Name	n
1	Calculus of variations 1	100, 1000
2	Calculus of variations 2	100, 1000
3	Calculus of variations 3	100, 1000
6	Generalized Rosenbrock	100, 1000
8	Penalty 1	1000
9	Penalty 2	100
10	Penalty 3	1000
12	Quadratic	1000
28	Extended Powell singular	1000
30	Trigonometric	100
31	Brown almost-linear	100
38	Tridiagonal 1	1000
39	Linear minimal surface	961
40	Boundary-value problem	1000
41	Broyden tridiagonal nonlinear	1000
42	Extended ENGVL1	1000
43	Ext. Freudenstein and Roth	1000
45	Wrong extended Wood	1000
48	Extended Rosenbrock	1000
49	Extended Powell	1000
50	Tridiagonal 2	1000
51	Trigonometric	1000
52	Penalty 1 (2nd version)	1000
54	Toint 61	1000
55	Toint 62	1000
102	Minpack-2 (DJOURB)	100, 1024
105	Minpack-2 (DTOR)	100, 1024

Table 1: List of test problems.

Problem	n	Modified				Stabilized			
		it	ls	cg	total	it	ls	cg	total
1	1000	78	258	432	690	49	83	212	295
1	100	46	187	195	382	39	75	153	228
2	1000	70	103	361	464	82	140	789	929
2	100	33	46	86	132	44	68	144	212
3	1000	77	144	448	592	97	167	834	1001
3	100	50	103	201	304	59	88	279	367
6	1000	48	88	166	254	114	192	1321	1513
6	100	32	56	88	144	66	117	305	422
8	1000	16	36	42	78	23	49	49	98
9	100	42	86	165	251	114	284	301	585
10	1000	45	68	174	242	87	212	316	528
12	1000	69	146	434	580	57	116	448	564
28	1000	19	25	57	82	16	36	46	82
30	100	32	52	90	142	36	57	86	143
31	100	38	128	112	240	53	119	152	271
38	1000	60	128	324	452	76	135	501	636
39	961	56	128	572	700	98	243	1134	1377
40	1000	32	88	141	229	35	65	175	240
41	1000	39	82	164	246	50	92	240	332
42	1000	42	101	179	280	37	83	95	178
43	1000	38	57	147	204	55	111	292	403
45	1000	42	65	142	207	55	128	153	281
48	1000	29	86	89	175	60	157	149	306
49	1000	19	25	57	82	16	36	46	80
50	1000	63	139	339	478	72	129	599	728
51	1000	57	122	190	312	49	79	283	362
52	1000	54	148	283	431	69	113	579	692
54	1000	56	138	248	386	68	108	247	355
55	1000	59	120	232	352	255	413	801	1214
102	1024	91	151	685	836	82	134	573	707
102	100	47	82	176	258	40	74	148	222
105	1024	69	106	437	543	64	115	508	623
105	100	44	69	157	226	37	71	120	191
Totals		1592	3361	7613	10974	2154	4089	12078	16165

Table 2: Results using (a) modified barrier method, (b) stabilized barrier method plus enhancements. Column "it" records the number of outer iterations, "ls" records the number of gradient evaluations used in the line search, "cg" records the number of gradient evaluations used in the inner iteration, and "total" records the total number of gradient evaluations ("ls" plus "cg").

μ	Individual			Cumulative				Gap	$\nabla\mathcal{L}$
	it	ls	cg	it	ls	cg	total		
1.46×10^{5}	1	2	2	1	2	2	4	6.8×10^{-3}	5.5×10^{-6}
1.46×10^{4}	5	9	29	6	11	31	42	1.1×10^{-2}	6.7×10^{-3}
1.46×10^{3}	6	7	69	12	18	100	118	5.4×10^{-3}	1.4×10^{-3}
1.46×10^{2}	5	7	30	17	25	130	155	2.6×10^{-3}	2.9×10^{-4}
1.46×10^{1}	9	16	46	26	41	176	217	6.4×10^{-4}	3.5×10^{-6}
1.46×10^{0}	11	18	62	37	59	238	297	8.5×10^{-5}	1.2×10^{-5}
1.46×10^{-1}	7	12	28	44	71	266	337	8.8×10^{-6}	1.1×10^{-5}
1.46×10^{-2}	3	4	13	47	75	279	354	8.8×10^{-7}	1.0×10^{-5}
$1.46 \times 10^{-3}*$	1	2	2	48	77	281	358	8.8×10^{-8}	2.8×10^{-9}
$1.46 \times 10^{-4}*$	1	2	2	49	79	283	362	8.8×10^{-9}	1.8×10^{-10}

Table 3: Using the stabilized barrier method to solve problem 51 with $n = 1000$. An $*$ indicates subproblems where the 1-inverse formula for the search direction was used. Column "it" records the number of outer iterations, "ls" records the number of gradient evaluations used in the line search, "cg" records the number of gradient evaluations used in the inner iteration, and "total" records the total number of gradient evaluations ("ls" plus "cg"). The column "Gap" records the (scaled) duality gap, and the column "$\|\nabla\mathcal{L}\|$" records the norm of the (scaled) Lagrangian function.

μ	Individual			Cumulative				Gap	$\nabla\mathcal{L}$	Infeasibility
1.00×10^{-3}	13	23	55	13	23	55	78	1.1×10^{-2}	2.1×10^{-3}	5.7×10^{-2}
5.00×10^{-4}	17	25	56	30	48	111	159	1.6×10^{-4}	5.5×10^{-7}	1.2×10^{-3}
2.50×10^{-4}	15	45	45	45	93	156	249	1.1×10^{-6}	1.5×10^{-6}	7.9×10^{-6}
1.25×10^{-4}	7	8	20	52	101	176	277	3.7×10^{-9}	3.5×10^{-7}	2.7×10^{-8}
6.25×10^{-5}	5	21	14	57	122	190	312	7.4×10^{-11}	7.7×10^{-7}	7.6×10^{-10}

Table 4: Using the modified barrier method to solve problem 51 with $n = 1000$. Column "it" records the number of outer iterations, "ls" records the number of gradient evaluations used in the line search, "cg" records the number of gradient evaluations used in the inner iteration, and "total" records the total number of gradient evaluations ("ls" plus "cg"). The column "Gap" records the (scaled) duality gap, the column "$\|\nabla\mathcal{M}\|$" records the norm of the (scaled) Lagrangian function, and the column "Infeasibility" records the infinity norm of the infeasibilities with respect to the bound constraints.

μ	Individual			Cumulative				Gap	$\nabla\mathcal{L}$
1.00×10^{2}	15	55	59	15	55	59	114	2.0×10^{-3}	3.8×10^{-6}
1.00×10^{1}	19	106	65	34	161	124	285	4.7×10^{-4}	3.6×10^{-7}
1.00×10^{0}	22	72	98	56	233	222	455	5.8×10^{-5}	8.7×10^{-9}
1.00×10^{-1}	20	122	107	76	355	329	684	6.0×10^{-6}	5.1×10^{-8}
1.00×10^{-2}	19	69	104	95	424	433	857	6.0×10^{-7}	4.3×10^{-8}
1.00×10^{-3}	19	102	118	114	526	551	1077	6.0×10^{-8}	8.1×10^{-8}
1.00×10^{-4}	19	59	97	133	585	648	1233	6.0×10^{-9}	2.7×10^{-8}

Table 5: Using a naive barrier method to solve problem 51 with $n = 1000$. Column "it" records the number of outer iterations, "ls" records the number of gradient evaluations used in the line search, "cg" records the number of gradient evaluations used in the inner iteration, and "total" records the total number of gradient evaluations ("ls" plus "cg"). The column "Gap" records the (scaled) duality gap, and the column "$\|\nabla\mathcal{L}\|$" records the norm of the (scaled) Lagrangian function.