A Merit Function for Inequality Constrained Nonlinear Programming Problems *

Paul T. Boggs † Jon W. Tolle ‡ Anthony J. Kearsley §

February 10, 1993

Abstract

We consider the use of the sequential quadratic programming (SQP) technique for solving the inequality constrained minimization problem $\min_x f(x)$ subject to: $g_i(x) \leq 0,\ i = 1, \ldots, m$. SQP methods require the use of an auxiliary function, called a merit function or line-search function, for assessing the steps that are generated. We derive a merit function by adding slack variables to create an equality constrained problem and then using the merit function developed earlier by the authors for the equality constrained case. We stress that we *do not* solve the slack variable problem, but only use it to construct the merit function. The resulting function is simplified in a certain way that leads to an effective procedure for updating the squares of the slack variables. A globally convergent algorithm, based on this merit function, is suggested, and is demonstrated to be effective in practice.

*Contribution of the National Institute of Standards and Technology and not subject to copyright in the United States. This research partially supported by AFOSR Contract #88-0267

†Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

‡Mathematics Department, University of North Carolina, Chapel Hill, NC 27599

§Department of Computational & Applied Mathematics, Rice University, Houston, Texas 77251-1892

1. Introduction

We consider the inequality-constrained minimization problem,

$$\begin{aligned} &\min_x \ f(x) \\ &\text{subject to:} \quad g(x) \le 0 \end{aligned} \tag{1.1}$$

where $x \in \Re^n$, and $f : \Re^n \to \Re$, and $g : \Re^n \to \Re^m$ are smooth functions. One of the most successful methods for solving (1.1) is the sequential quadratic programming (SQP) method in which at each iteration a quadratic program is solved to obtain the step direction toward the next iterate. In particular, given a current approximation, x^c, to the solution, x^*, one forms the quadratic program

$$\begin{aligned} &\min_\delta \ \nabla f(x^c)^\mathsf{T}\delta + \tfrac{1}{2}\delta^\mathsf{T} B^c \delta \\ &\text{subject to:} \quad \nabla g(x^c)^\mathsf{T}\delta + g(x^c) \le 0 \end{aligned} \tag{1.2}$$

where B^c is usually taken to be a current approximation to the Hessian of the Lagrangian of (1.1). Let δ^c be the solution of (1.2). Then the next approximation, x^+, is calculated by

$$x^+ = x^c + \alpha\delta^c \tag{1.3}$$

where α is a scalar *steplength*. This procedure is repeated until convergence.

Actual algorithms based on the above have been formulated and coded, and have performed well in practice on a wide variety of problems. Certain aspects of the general SQP scheme are still the subject of much current research, and the theory that explains the performance is still incomplete. In this paper we address one issue that has not yet been satisfactorily resolved; namely, a means of choosing an appropriate steplength, α, so as to guarantee global convergence. In particular, we propose a *merit function* for the inequality-constrained problem (1.1). The approach we take is based on our earlier work, [BogT84] and [BogT89], in which a merit function was provided for equality-constrained problems.

By a merit function we mean a function that can be used as a test to assure that the potential step given by (1.3) will make x^+ a better approximation to the solution than x^c. For unconstrained optimization problems, the objective function serves this purpose; one simply chooses α such that

$$f(x^+) < f(x^c)$$

(with appropriate restrictions on α to assure that a sufficient decrease is achieved). For constrained optimization, the possible decrease in the objective value must be weighed against the requirement that feasibility must also be achieved — at least in the limit. Therefore, if one is using a method, such as an SQP algorithm, that does not maintain feasibility at each step, the objective function by itself is not an acceptable merit function.

Numerous papers have been published that set forth suggested merit functions; the reader is referred to [BogT89] for a survey of some of the more recent research. Most of the work that has appeared has been directed towards the equality-constrained problem; straightforward generalizations to the inequality-constrained problem are made difficult by the complicated nature and, in many cases, the non-differentiability of the merit function on the boundary of the feasible region. Work that has been directed to, or encompasses, the inequality-constrained problem includes that of [Han77], [ChaLLP82], [Sch83], and [GilMSW86]. Han, in his initial papers on the SQP methods employed an l_1 penalty function for his merit function. Chamberlain et al. [ChaLLP82] developed their "watchdog" technique to correct a deficiency in Han's merit function that permitted a step length of one to be rejected near the solution, thereby possibly precluding q-superlinear convergence. Schittkowski used an augmented Lagrangian penalty function that seems to work well on highly nonlinear problems; Gill *et al.* have incorporated a version of Schittowski's merit function into their NPSOL program.

Our work on equality constrained problems employs an exact penalty function originally due to Fletcher [Fle72], but derived differently. (See [BogT84].) Powell and Yuan [PowY86] use a similar function. While both approaches are effective, Shanno and Phua [ShaP89] have studied the class and prefer our method in numerical computations.

The merit function proposed here differs from most, if not all, others in that it does not have an unconstrained local minimum at x^* but, rather, a constrained minimum at (x^*, z^*) where the variables z are *nonnegatively constrained* slack variables. The implementation of the SQP algorithm proposed for this merit function will assure that these slack variables never become negative so that, practically speaking, the constraints are of no consequence in the computation. This merit function has excellent theoretical properties; it is smooth in its variables and exact. Moreover, locally the SQP step will always be a descent direction for this function and the length of the step will not be so restricted as to prevent q-superlinear convergence, i.e., the Maratos effect [ChaLPP82] will not occur. As in [BogT89] we require a modification to obtain a globally convergent procedure. In this paper, however, we go beyond our previous work and propose an efficient, implementable algorithm, based on using both the proposed merit function and an "approximate" version of it, for which a global convergence theorem can be proved.

An outline of the paper is as follows. In §2 we motivate our proposed merit function and establish its basic properties. In §3 we suggest a modification of the function necessary to achieve global convergence, and we incorporate these ideas in a somewhat detailed algorithm. We then state and sketch a proof of the global convergence of the procedure. Finally we discuss the consequences of relaxing some of the conditions. §4 contains further algorithmic details, the results of numerical experiments, and some comments on their significance.

Because the structure and spirit of this paper are similar to that of [BogT89] we do not give complete proofs of every proposition. These omissions can be filled in; but only at some cost in simplicity of exposition and with no compensating increase in clarity and insight. Where omissions occur the reader is directed to our earlier work.

One of the motivations for this work was the development of algorithms for large scale optimization problems. For these problems we employ an iterative interior-point method for approximately solving the quadratic subproblems. We show (see [BogTK92]) that these approximate steps are descent steps for the merit function that is proposed here and that much of the convergence theory carries over to this case. The current paper therefore provides the basis for the large scale case.

2. A Merit Function for the Inequality Constrained Problem

In this section we present a merit function for use with an SQP method for solving inequality-constrained problems. This new merit function is motivated by introducing squared slack variables and applying our earlier merit function to the resulting equality-constrained problem. We stress, however, that the result is not simply an extension of our earlier work to the problem with these squared slack variables; there are several significant changes that are required. First, we *do not* solve the slack variable equality-constrained problem; we solve (1.1) by solving the approximating inequality-constrained quadratic subproblems (1.2). Second, our merit function does not update the slack variables themselves but, rather, their squares. Finally, the solution to the original problem corresponds to a *constrained* minimum of the merit function. A version of this approach was first outlined in [BogT88]; here a revised and more complete development is given.

As suggested, we motivate our merit function by first converting (1.1) to an equality-constrained problem by the addition of squared slack variables, yielding

$$\begin{array}{l} \min\limits_{x,s} f(x) \\ \text{subject to:} \quad g(x) + \frac{1}{4}S^2 e = 0 \end{array} \tag{2.1}$$

where

$$S = \operatorname{diag}\{s_1, \ldots, s_m\}.$$

Here, as elsewhere in this paper, e represents a vector of ones. (The constant $\frac{1}{4}$ is chosen for convenience.)

It is well-known that if x^* is a solution to (1.1) then (x^*, s^*) with s_i^* defined to be $(-g_i(x^*))^{1/2}$ is a solution to (2.1) and conversely. Most researchers consider the actual solution of (2.1) rather than (1.1) to be impractical, for both computational and theoretical reasons. However, following [BogT84], we construct a merit function for (2.1) of the form

$$\phi_d(x,s) = f(x) + \lambda(x,s)^{\mathsf{T}} c(x,s) + \frac{1}{d} c(x,s)^{\mathsf{T}} A(x,s)^{-1} c(x,s)$$

where

$$c(x,s) = g(x) + \frac{1}{4}S^2 e,$$

$$\begin{aligned} A(x,s) &= \nabla c(x,s)^{\mathsf{T}} \nabla c(x,s) \\ &= \nabla g(x)^{\mathsf{T}} \nabla g(x) + \frac{1}{4}S^2, \end{aligned}$$

and

$$\begin{aligned} \lambda(x,s) &= -A(x,s)^{-1} \nabla c(x,s)^{\mathsf{T}} \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix} \\ &= -A(x,s)^{-1} \nabla g(x)^{\mathsf{T}} \nabla f(x). \end{aligned}$$

The small scalar parameter d is chosen to penalize infeasibility. Observe that $\lambda(x,s)$ is the least squares estimate of the multiplier vector that solves the first order optimality conditions for (2.1):

$$\nabla \left[f(x) + \lambda^{\mathsf{T}} c(x,s) \right] = 0.$$

Here, and in the remainder of the paper, the symbol $\nabla h(u, v)$ refers to the differentiation of the function h with respect to all of its variables. Differentiation with respect to a subset of the variables is denoted by a subscript, e.g., $\nabla_u h(u, v)$ means the vector of partial derivatives of h with respect to the variables in the vector u. A similar convention holds for second order derivatives.

The observation that $\phi_d(x, s)$ depends only on the squared components of s leads us to replace the term $\frac{1}{4}s_i^2$ by z_i and to consider the resulting function:

$$\psi_d(x, z) = f(x) + \bar{\lambda}(x, z)^{\mathsf{T}} \bar{c}(x, z) + \frac{1}{d}\bar{c}(x, z)^{\mathsf{T}} \bar{A}(x, z)^{-1}\bar{c}(x, z)$$

where

$$\begin{aligned} \bar{c}(x, z) &= g(x) + Ze \\ \bar{A}(x, z) &= \nabla g(x)^{\mathsf{T}} \nabla g(x) + Z \\ \bar{\lambda}(x, z) &= -\bar{A}(x, z)^{-1} \nabla g(x)^{\mathsf{T}} \nabla f(x) \end{aligned}$$

and

$$Z = \operatorname{diag}\{z_1, \ldots, z_m\}.$$

It is important to note that $\psi_d(x, z)$ is *not* the function that would result from constructing the merit function for (2.1) after replacing $\frac{1}{4}s_i^2$ by z_i . This observation notwithstanding we consider the z_i as nonnegative slack variables for (1.1), and, as a consequence, we have the constrained minimization of ψ_d:

$$\begin{aligned} &\min_{x,z} \ \psi_d(x, z) \\ &\text{subject to:} \quad z \geq 0. \end{aligned} \tag{2.2}$$

Our strategy for using the merit function requires that we maintain $z \geq 0$; thus the nonnegativity constraints create neither theoretical nor computational difficulties.

In the following we develop the theoretical properties of the merit function $\psi_d(x, z)$ and relate the solutions of the problems (1.1) and (2.2). Throughout this paper we use the term "strong solution" of a constrained optimization problem to mean a local optimal point together with a multiplier vector at which the following hold.

1. The active constraint gradients are linearly independent.

2. Strict complementary slackness holds.

3. The second order sufficient conditions hold.

Our first assumption is the following:

A1: The local solutions to (1.1) are strong solutions.

It is well known that a strong local solution is isolated and is stable with respect to small perturbations in the data of the problem. It can also be inferred, following the lines of argument in [Rob74] that if $\{x^k\}$ is a sequence converging to x^*, a strong local solution to (1.1), and $\{B^k\}$ is a sequence of uniformly bounded positive definite matrices, then the active constraints of (1.1) at x^* and the active constraints of (1.2) at x^k will coincide for k sufficiently large. This is an essential result for the theoretical justification of our proposed algorithm, as well as for most other SQP algorithms. In particular it means that the necessary and sufficient conditions for superlinear convergence of an SQP method for equality-constrained problems given in [BogTW82] can be carried over to the inequality-constrained problem in a straightforward manner by using the projection onto the null space of the active constraint gradients.

The preceding formal derivation of the merit function $\psi_d(x, z)$ does not address the question of whether or not it is defined, i.e., if $\bar{A}(x, z)$ is nonsingular, at each point of (x, z) encountered in the algorithm. There is, of course, no way to assure this in general; some assumptions on the structure of the problem must be made. Below we make an assumption that will guarantee that $\psi_d(x, z)$ is well-defined and give conditions under which it holds. The discussion of the implementation of the algorithm in the remainder of this section and in Section 3 will show that this assumption is relatively benign in terms of any restrictions on the algorithm's application.

To formulate this assumption we denote the index set for the constraints by M and we partition M into two subsets a and u. Then we can write, without loss of generality,

$$g(x) = \begin{pmatrix} g_a(x) \\ g_u(x) \end{pmatrix}$$

and, in a corresponding manner,

$$z = \begin{pmatrix} z_a \\ z_u \end{pmatrix}.$$

Usually the index subset a will correspond to the set of active constraints for either (1.1) or (1.2). For example, at iteration k we could set $a^k = \{i : \nabla g_i(x^k)^\mathsf{T}\delta^k +$

$g_i(x^k) = 0\}$ and hence $u^k = \{i : \nabla g_i(x^k)^\mathsf{T}\delta^k + g_i(x) \neq 0\}$. The basic assumption is as follows:

A2: For a given point (x, z) there is a partition of M such that the set $\{\nabla g_i(x) : \; i \in a\}$ is linearly independent and $z_u > 0$.

Clearly, under this assumption $\bar{A}(x, z)$ is nonsingular and hence $\psi_d(x, z)$ is well-defined. The following proposition, which is easily proved, cites situations in which **A2** is valid.

Proposition 1. **A2** *holds in a neighborhood of* (x, z) *if any of the following hold:*

(i) x *is a strong solution of (1.1) and* $z_i = -g_i(x)$.

(ii) *(1.2) has a strong solution,* δ, *at* $x^c = x$ *and* $z_i > 0$ *if* $\nabla g_i(x)^\mathsf{T}\delta + g_i(x) < 0$.

(iii) $z > 0$.

(i) shows that **A2** is essentially subsumed by **A1** in neighborhoods of the solutions to (1.1). The first part of (ii) is an implicit assumption of most algorithms using the SQP method and our method for updating z will generally assure that the second part is satisfied. In Section 3 we will remark on the possibility of updating the z such that (iii) is satisfied at every point, thus guaranteeing that **A2** is always satisfied.

We can now show that the solutions to (1.1) lead to special solutions of (2.2) and vice versa. It will be useful to make the following definition which will be used throughout the remainder of this paper. Let

$$\mathcal{S} = \{(x^*, z^*) \colon \; x^* \text{ is a strong local solution of (1.1) and } z^* = -g(x^*)\}.$$

Proposition 2. *Let* $(x^*, z^*) \in \mathcal{S}$ *with multiplier vector* λ^*. *Then if* d *is sufficiently small it is the case that* (x^*, z^*) *is a strong local solution to (2.2) with multiplier* $\bar{\lambda}(x^*, z^*) = \lambda^*$.

Proof: We note that the conditions on x^* and z^* assure that **A2** is satisfied at (x^*, z^*) and hence that $\psi_d(x, z)$ is well-defined in a neighborhood of (x^*, z^*). Also $\bar{c}(x^*, z^*) = 0$. Thus

$$\nabla_x \psi_d(x^*, z^*) = \nabla f(x^*) + \nabla g(x^*)\bar{\lambda}(x^*, z^*)$$

and

$$\nabla_z \psi_d(x^*, z^*) = \bar{\lambda}(x^*, z^*).$$

From its definition, $\bar{\lambda}(x,z)$ is seen to be the least squares solution of

$$\begin{pmatrix} \nabla g(x) \\ Z^{1/2} \end{pmatrix} \bar{\lambda} + \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix} = 0. \tag{2.3}$$

But from the properties of x^* and the definition of z^* it follows that $z_i^* \lambda_i^* = 0$ for each i. Thus λ^* is an exact solution of (2.3) at (x^*, z^*), and so $\bar{\lambda}(x^*, z^*)$ must be λ^*. It now follows that $\nabla_x \psi_d(x^*, z^*) = 0$, $\nabla_z \psi_d(x^*, z^*) = \lambda^*$, and strict complementary slackness holds. Therefore (x^*, z^*, λ^*) is a solution to (2.2). The proof that the second order sufficient conditions hold for this problem is straightforward.

□

The proof of the converse follows essentially the same argument.

Proposition 3. *Let (x^*, z^*) be a strong local solution to (2.2) having multiplier $\lambda^*$and satisfying $g(x^*) + z^* = 0$ and $z^* \geq 0$. Then $(x^*, z^*) \in \mathcal{S}$ with multiplier λ^* and $\lambda^* = \bar{\lambda}(x^*, z^*)$.*

The fundamental part of the algorithm described in the next section determines the direction of the step for changes in the current iterates x and z. Given the current values x, z, and positive definite Hessian approximation B, the step direction δ is obtained as the solution of (1.2). The step direction for z is taken to be q where

$$q = -\left[\nabla g(x)^{\mathsf{T}} \delta + g(x) + z\right]. \tag{2.4}$$

The choice of q given in (2.4) can be motivated by considering the slack variable problem

$$\begin{array}{ll} \min\limits_{x,z} f(x) & \\ \text{subject to:} & \begin{array}{rcl} g(x) + z & = & 0 \\ z & \geq & 0. \end{array} \end{array}$$

A structured quadratic programming approximation to this problem can be seen, after some algebra, to have the form

$$\begin{array}{ll} \min\limits_{\delta,q} \nabla f(x)^{\mathsf{T}} \delta + \frac{1}{2} \delta^{\mathsf{T}} B \delta & \\ \text{subject to:} & \begin{array}{rcl} \nabla g(x)^{\mathsf{T}} \delta + q + g(x) + z & = & 0 \\ q + z & \geq & 0. \end{array} \end{array}$$

The equality constraint suggests the choice of q given by (2.4).

The new iterates will now be $(x+\alpha\delta, z+\alpha q)$ for some $\alpha \geq 0$. It is easily seen from the constraints of (1.2) that if the initial value of z is nonnegative then all succeeding values of z will also be nonnegative provided that α is not greater than one. In particular, if (a, u) is a partition of M with u corresponding to the inactive constraints of (1.2) at optimality we have that

$$z_u + \alpha q_u = (1-\alpha) z_u - \alpha \left[\nabla g_u^{\mathsf{T}} \delta + g_u \right] > 0.$$

So, for $\alpha \in (0, 1]$, inactive constraints of (1.2) lead to strictly positive values for the corresponding component of z.

Before going further we give some formulas that will be useful in establishing later results. In the following, (x, z) will be a point at which **A2** holds and (δ, q) will be the step direction obtained from (1.2) and (2.4).

Since

$$\nabla \bar{c}(x, z) = \begin{pmatrix} \nabla g(x) \\ I \end{pmatrix}$$

and, from (2.4),

$$\bar{c}(x, z) = g(x) + z = -\left(\nabla g(x)^{\mathsf{T}} \delta + q \right), \tag{2.5}$$

we have

$$\nabla \bar{c}(x, z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} = \left(\nabla g(x)^{\mathsf{T}} \delta + q \right) = -\bar{c}(x, z). \tag{2.6}$$

Two useful formulas that can be derived from the expression for $\bar{\lambda}(x, z)$ are

$$\begin{aligned} \nabla_x \bar{\lambda}(x, z) \quad = \quad & -W(x, \nabla_x L(x, \bar{\lambda}(x, z))) \\ & -\nabla_{xx} L(x, \bar{\lambda}(x, z)) \nabla g(x) \bar{A}(x, z)^{-1} \end{aligned} \tag{2.7}$$

and

$$\nabla_z \bar{\lambda}(x, z) = -\Lambda \bar{A}(x, z)^{-1} \tag{2.8}$$

where $L(x, v) = f(x) + g(x)^{\mathsf{T}} v$ is the Lagrangian function, $W(x, y) = O(y)$ uniformly in x, and $\Lambda = \mathrm{diag}\{\lambda_1, \ldots, \lambda_m\}$. Also we note that if λ is the multiplier for (1.2) then the KKT condition for (1.2) is

$$\nabla f(x) + \nabla g(x) \lambda = -B\delta. \tag{2.9}$$

This formula together with the complementary slackness conditions for (1.2) and the definitions of $\bar{A}(x, z)$ and q imply

$$\bar{\lambda}(x, z) - \lambda = \bar{A}(x, z)^{-1} \nabla g(x)^{\mathsf{T}} B\delta - \bar{A}(x, z)^{-1} \Lambda q. \tag{2.10}$$

We now identify some of the properties enjoyed by the merit function relative to the step direction (δ, q). An additional assumption that will hold throughout the remainder of this paper is required.

A3: The matrices B appearing in (1.2) are uniformly positive definite.

First we observe that the step is a descent direction for

$$r(x,z) = |\bar{c}(x,z)|^2.$$

Indeed, as a result of (2.6) we have

Proposition 4. *Let δ and q be defined as above. Then*

$$\nabla r(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} = -2r(x,z).$$

Thus, unless $\bar{c}(x,z) = 0$ the step always initially reduces constraint infeasibility as measured by $|\bar{c}(x,z)|^2$. The next proposition shows that when $\bar{c}(x,z)$ is close enough to zero the direction is also one of descent for $\psi_d(x,z)$.

Proposition 5. *Let $(\bar{x}, \bar{z})$ satisfy $\bar{c}(\bar{x},\bar{z}) = 0$ with $g(\bar{x}) \le 0$. Let G be any compact subset of $\Re^{n+m}$ containing $(\bar{x},\bar{z})$ such that $G \cap \mathcal{S} = \emptyset$. Then for each $d > 0$ there exists an $\epsilon(d) > 0$ such that for all $(x,z) \in G$ with $r(x,z) < \epsilon(d)$,*

$$\nabla \psi_d(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} < 0.$$

Proof: Let $(x,z) \in G$ with $\bar{c}(x,z) = 0$ and $g(x) \le 0$. Then

$$\nabla \psi_d(x,z) = \begin{pmatrix} \nabla f(x) \\ 0 \end{pmatrix} + \begin{pmatrix} \nabla g(x) \\ I \end{pmatrix} \bar{\lambda}(x,z).$$

From (2.5) and (2.9), it follows that

$$\nabla \psi_d(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} = -\delta^{\mathsf{T}} B \delta - \lambda^{\mathsf{T}} \nabla g(x)^{\mathsf{T}} \delta$$

where λ is the multiplier vector for (1.2). But if λ_j is positive then it is the case that

$$(\nabla g(x)^{\mathsf{T}} \delta + g(x))_j = 0$$

and, since $g_j(x) \le 0$,

$$\lambda^\mathsf{T} \nabla g(x)^\mathsf{T} \delta \ge 0.$$

Thus

$$\nabla \psi_d(x,z)^\mathsf{T} \begin{pmatrix} \delta \\ q \end{pmatrix} \le -\delta^\mathsf{T} B \delta < 0$$

unless $\delta = 0$ (which implies that x is a solution to (1.1)). The result now follows by a continuity argument.

□

The preceding proposition shows that the direction of the step in (x, z) is a descent direction for our merit function when $\bar{c}(x, z)$ is close to zero except possibly in neighborhoods of points in $\mathcal{S}$. We now patch that gap by showing that if $(x^*, z^*) \in \mathcal{S}$ then the step is a descent direction for $\psi_d(x, z)$ for (x, z) in a neighborhood of $(x^*\, z^*)$. First we recall that if x is close enough to x^* then the active constraints of (1.2) at x are the same as the active constraints of (1.1) at x^*. Thus for x sufficiently close to x^*, (1.2) is equivalent to

$$\begin{aligned} &\min_{\delta} \ \nabla f(x)^\mathsf{T} \delta + \tfrac{1}{2} \delta^\mathsf{T} B \delta \\ &\text{subject to:} \ \ \nabla g_a(x)^\mathsf{T} \delta + g_a(x) = 0 \end{aligned} \tag{2.11}$$

where (g_a, g_u) denotes the partition into active and inactive constraints at x^*. It follows from the theory for equality-constrained SQP methods (see, for example, [BogTW82]) that for x sufficiently close to x^* there is a positive constant K such that the solution of (2.11) satisfies

$$|\delta| \le K|x - x^*|.$$

If, in addition, z is close to $z^* = -g(x^*)$ and q is given by (2.4) then using the partition above we obtain

$$q_a = -\left(\nabla g_a(x)^\mathsf{T} \delta + g_a(x) + z_a\right) = -z_a$$

so

$$|q_a| \le |z_a| = |z_a - z_a^*|.$$

Since $z_u^* = -g_u(x^*)$,

$$\begin{aligned} |q_u| &= |\nabla g_u(x)^\mathsf{T} \delta + g_u(x) - g_u(x^*) - z_u^* + z_u| \\ &\le K_1|x - x^*| + K_2|x - x^*| + |z_u - z_u^*| \end{aligned}$$

for some constants K_1 and K_2. Thus δ and q are both of order $(|x - x^*| + |z - z^*|)$. We can now prove the following.

Proposition 6. *Let* $(x^*, z^*) \in \mathcal{S}$ *with multiplier vector* λ^*. *Then there exist positive constants* ρ *and* ξ *such that*

$$\nabla\psi_d(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} < 0$$

whenever $0 < |(x,z) - (x^*, z^*)| \leq \rho$ *and* $0 < d \leq \xi$.

Proof: We assume ρ is sufficiently small that the bounds on δ and q preceding the statement of the proposition are valid. By Proposition 1 it is seen that $\bar{A}(x,z)^{-1}$ is well-defined and uniformly positive definite for ρ small. From (2.5) – (2.8),

$$\begin{aligned} \nabla\psi_d(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} &= \nabla f(x)^{\mathsf{T}}\delta + \bar{\lambda}(x,z)^{\mathsf{T}} w - \frac{2}{d} w^{\mathsf{T}} \bar{A}(x,z)^{-1} w \\ &\quad + O(p|w|) + \frac{1}{d} O(p|w|^2) \end{aligned}$$

where $p = \min\{|\delta|, |q|\}$ and $w = \left(\nabla g(x)^{\mathsf{T}}\delta + q\right)$. Letting λ be the multiplier vector for (1.2), using (2.9) and (2.10), and simplifying gives

$$\begin{aligned} \nabla\psi_d(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} &= -\delta^{\mathsf{T}} B\delta + q^{\mathsf{T}}\lambda - \frac{2}{d} w^{\mathsf{T}} \bar{A}(x,z)^{-1} w \\ &\quad + O(p|w|) + \frac{1}{d} O(p|w|^2). \end{aligned}$$

Now $q_u^{\mathsf{T}}\lambda_u = 0$ and $q_a^{\mathsf{T}}\lambda_a = -z_a^{\mathsf{T}}\lambda_a \leq 0$, so using the uniform positive definiteness properties of the B and $\bar{A}(x,z)^{-1}$ we have that there are positive constants $\gamma_1, \gamma_2, \gamma_3$, and γ_4 such that

$$\nabla\psi_d(x,z)^{\mathsf{T}} \begin{pmatrix} \delta \\ q \end{pmatrix} \leq -\gamma_1 p^2 + \gamma_2 p|w| - \frac{1}{d}\gamma_3 |w|^2 + \frac{1}{d}\gamma_4 p|w|^2.$$

It can be seen that for ρ and d sufficiently small that the right-hand side is negative and the proposition is proven.

□

A final important property of the merit function is that it doesn't impede the progress of q-superlinear convergence. That is, if the sequence is converging q-superlinearly then a steplength of one is acceptable near the solution. Since, as stated above, the quadratic program will identify the correct active constraints near the solution, the value of δ in that area will be given as the solution of (2.11). If we denote by P_a the projection onto the space orthogonal to the gradients of the active constraints at x^* then the characterization of the q-superlinear convergence of the sequence $\{x^k\}$ generated by the SQP algorithm is ([BogTW82]),

$$\lim_{k\to\infty} \frac{|P_a(\nabla^2_{xx}L(x^k,\lambda^k) - B^k)\delta^k|}{|\delta^k|} = 0 \tag{2.12}$$

where (δ^k, λ^k) is the optimal solution-multiplier pair for the quadratic program (2.11).

To prove the result we need a further, mild assumption, namely "tangential convergence" of the iterates x^k. To explain, let $Q_a = I - P_a$. Then $\{x^k\}$ is said to converge tangentially if

$$\frac{\left|Q_a\delta^k\right|}{|\delta^k|} \to 0.$$

In [BogT89] it is shown that tangential convergence implies q-superlinear convergence and [BogT92] we argue that the converse nearly always holds, especially in the nonconvex case. Our computational experience and that of others support this conclusion.

Proposition 7. *Let the hypotheses of Proposition 6 hold and assume that the sequence $\{x^k\}$ generated by the SQP converges to x^* tangentially and q-superlinearly. Then there exists a $\bar{d} > 0$ such that for each $d \in (0, \bar{d})$ there is a positive integer $J(d)$ satisfying*

$$\psi_d(x^k + \delta^k, z^k + q^k) < \psi_d(x^k, z^k)$$

for $k \geq J(d)$.

Proof: For ease of notation we drop the superscript k henceforth. Defining p and w as in the proof of Proposition 6 and using equations (2.5) – (2.10) it follows from the Taylor Series that

$$\begin{aligned}
\Delta &= \psi_d(x+\delta, z+q) - \psi_d(x,z) \\
&= -\delta^\top B\delta + \frac{1}{2}\delta^\top \nabla_{xx}L(x,\lambda)\delta + \lambda^\top q \\
&\quad -q^\top \Lambda \bar{A}(x,z)^{-1}w + \delta^\top B\nabla g(x)\bar{A}(x,z)^{-1}w \\
&\quad -\frac{1}{d}w^\top \bar{A}(x,z)^{-1}w \; + (1+\frac{1}{d})O(p^3).
\end{aligned} \tag{2.13}$$

If x is close enough to the solution that the active set has been identified at each iterate $(\nabla g_a(x)^\mathsf{T}\delta + g_a = 0)$ then

$$z_a + q_a = -(\nabla g_a(x)^\mathsf{T}\delta + g_a) = 0.$$

Thus $q_a = -z_a$. Since $\lambda_a \geq 0$ it follows that $\lambda^\mathsf{T} q \leq 0$. Now letting $Q_a = I - P_a$ and adding and subtracting $\frac{1}{2}\delta^\mathsf{T} B\delta$ on the right-hand side of (2.13) we obtain

$$\begin{aligned}
\Delta \quad = \quad & -\frac{1}{2}\delta^\mathsf{T} B\delta + \frac{1}{2}\delta^\mathsf{T} P_a(\nabla_{xx}L(x,\lambda) - B)\delta \\
& +\frac{1}{2}\delta^\mathsf{T} Q_a(\nabla_{xx}L(x,\lambda) - B)\delta - q^\mathsf{T}\Lambda\bar{A}(x,z)^{-1}w \\
& -\frac{1}{d}w^\mathsf{T}\bar{A}(x,z)^{-1}w + \delta^\mathsf{T} B\nabla g(x)\bar{A}(x,z)^{-1}w \qquad (2.14) \\
& +(1+\frac{1}{d})O(p^3).
\end{aligned}$$

Thus letting $\epsilon = |w|/|\delta|$ in (2.14) and using **A3**, we can write

$$\begin{aligned}
\frac{\Delta}{|\delta|^2} \quad \leq \quad & -\frac{1}{2}\mu + K_1\epsilon - \frac{1}{d}\gamma\epsilon^2 + K_2\frac{|Q_a\delta|}{|\delta|} \\
& +\frac{1}{2}\frac{|P_a(\nabla_{xx}L(x,\lambda) - B)\delta|}{|\delta|} + (1+\frac{1}{d})O(p).
\end{aligned}$$

From (2.12) and the assumption of tangential convergence it can be seen that for d sufficiently small the right-hand side can be made negative provided p is small, thus proving the result.

□

3. A Globally Convergent Algorithm

From the basic propositions presented in the preceding section a global convergence theory can be constructed using the merit function $\psi_d(x,z)$. Some modification is, however, required since the SQP step is not necessarily a descent direction outside of a neighborhood of the feasible region. Moreover, the merit function $\psi_d(x,z)$ is relatively expensive to evaluate at points other than iterates and such evaluations are necessary when carrying out a line search. In this section we define an "approximate" merit function that can be used in conjunction with $\psi_d(x,z)$ to create a globally convergent algorithm. We then state the algorithm and briefly discuss its theory

We define our approximate merit function at the kth iteration to be

$$\psi_d^k(x,z) = f(x) + (g(x)+z)^{\mathsf{T}}\bar{\lambda}^k + \frac{1}{d}(g(x)+z)^{\mathsf{T}}\bar{A}_k^{-1}(g(x)+z)$$

where

$$\bar{A}_k = \nabla g(x^k)^{\mathsf{T}}\nabla g(x^k) + Z^k$$

and

$$\bar{\lambda}^k = -\bar{A}_k^{-1}\nabla g(x^k)^{\mathsf{T}}\nabla f(x^k).$$

Note that the use of $\psi_d^k(x,z)$ as a merit function requires only the evaluation of f and g to test a prospective new iterate. While it is cheap to evaluate, however, it is not a single merit function, and thus needs to be combined with a strategy that will ensure global convergence.

The relevant properties of $\psi_d^k(x,z)$ are identical to those of $\psi_d(x,z)$ provided in the preceding section except that the descent property of $\psi_d^k(x,z)$ is stronger than that for $\psi_d(x,z)$. In fact, (δ^k, q^k) is *always* a descent direction for $\psi_d^k(x,z)$, for d sufficiently small, as is a consequence of the following proposition.

Proposition 8. *Let C be a compact subset of $\Re^n$ not containing a solution of (1.1). There exists a $\bar{d}$ sufficiently small such that if (x^k, z^k) is a point of C at which $\nabla\psi_d^k(x^k,z^k) \neq 0$ then*

$$\nabla\psi_d^k(x^k,z^k)^{\mathsf{T}}\begin{pmatrix}\delta^k\\ q^k\end{pmatrix} < 0$$

provided $d \leq \bar{d}$.

Proof: It is seen that

$$\begin{aligned}
\Delta &= \nabla\psi_d^k(x^k,z^k)^{\mathsf{T}}\begin{pmatrix}\delta^k\\ q^k\end{pmatrix}\\
&= \nabla f(x^k)^{\mathsf{T}}\delta^k\\
&+ (\bar{\lambda}^k)^{\mathsf{T}}\begin{pmatrix}\nabla g(x^k)\\ I\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}\delta^k\\ q^k\end{pmatrix}\\
&- \frac{1}{d}\begin{pmatrix}\delta^k\\ q^k\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}\nabla g(x^k)\\ I\end{pmatrix}\bar{A}_k^{-1}\begin{pmatrix}\nabla g(x^k)\\ I\end{pmatrix}^{\mathsf{T}}\begin{pmatrix}\delta^k\\ q^k\end{pmatrix}.
\end{aligned}$$

Using (2.9) and (2.10) yields, after some cancellation,

$$\begin{aligned}\Delta &= -(\delta^k)^\top B^k \delta^k + \lambda^\top q^k \\ &+ (\delta^k)^\top B^k \begin{pmatrix} \nabla g(x^k) \\ I \end{pmatrix} \bar{A}_k^{-1} \begin{pmatrix} \delta^k \\ q^k \end{pmatrix} - (q^k)^\top \Lambda \bar{A}_k^{-1} \begin{pmatrix} \delta^k \\ q^k \end{pmatrix} \\ &- \frac{1}{d} \begin{pmatrix} \delta^k \\ q^k \end{pmatrix}^\top \begin{pmatrix} \nabla g(x^k) \\ I \end{pmatrix} \bar{A}_k^{-1} \begin{pmatrix} \nabla g(x^k) \\ I \end{pmatrix}^\top \begin{pmatrix} \delta^k \\ q^k \end{pmatrix}.\end{aligned}$$

As in the proof of Proposition 6, $\lambda^\top q^k \leq 0$. Defining w as in the same proof it follows that there exist a constant K such that

$$\Delta \leq -(\delta^k)^\top B^k \delta^k + K|w| - \frac{1}{d} w^\top \bar{A}_k^{-1} w.$$

Using assumption **A3** it is seen that the right hand side of the above inequality has a negative maximum for all d sufficiently small.

□

The proof for the case when x^k is in a neighborhood of a solution to (1.1) is nearly identical to that of Proposition 6 and is not given here. Similarly the proof that if the algorithm is converging q-superlinearly, then a steplength of one is acceptable for $\psi_d^k(x, z)$ near the solution parallels the proof of Proposition 7.

Our strategy for global convergence depends on the fact (Propositions 5, 6, and 8) that there is a positive constant η such that if $|g(x) + z| \leq \eta$ then both the function $\psi_d(x, z)$ and $\psi_d^k(x, z)$ are reduced in the direction (δ, q). For a given η, if $|g(x) + z| \leq \eta$ then our strategy is to insist that $\psi_d^k(x, z)$ be reduced. In most cases, $\psi_d(x, z)$ will then be reduced also. If not, then the current value of η is too large and it is reduced. Generally, we prefer η not to be too small initially so that the constraints are not followed too closely. If $|g(x) + z| > \eta$ our procedures call for both $\psi_d^k(x, z)$ and $|g(x) + z|$ to be reduced. This is possible because of the stronger descent property of $\psi_d^k(x, z)$. The details are given in the algorithm description below. This description assumes that we have an appropriate value of d; we comment on this problem in §4.

Algorithm

1. Given $d > 0$, x^0, B^0:
   Calculate $f(x^0)$, $\nabla f(x^0)$, $g(x^0)$, $\nabla g(x^0)$ and $c(x^0, z^0)$.

Initialize ${z^0}_i = -g_i$ for all $g_i < 0$ and ${z^0}_i = 1$ otherwise, and set $\eta = 2\left|g(x^0)\right|^2$.
Calculate $\bar{\lambda}^0$ and $\psi_d(x^0, z^0)$, and set $k = 0$.

2. Solve

$$\min_{\delta} \nabla f(x^k)^{\mathsf{T}}\delta + \tfrac{1}{2}\delta^{\mathsf{T}} B^k \delta$$
$$\text{subject to: } \nabla g(x^k)^{\mathsf{T}}\delta + g(x^k) \leq 0$$

   to obtain δ^k.

3. Set $q^k = -[\nabla g(x^k)^{\mathsf{T}}\delta^k + g(x^k) + z^k]$.

4. Compute $\alpha^k \leq 1$ such that

$$\psi_d^k(x^k + \alpha^k\delta^k, z^k + \alpha^k q^k) < \psi_d^k(x^k, z^k).$$

5. If $\left|c(x^k + \alpha^k\delta^k, z^k + \alpha^k q^k)\right| \geq \left|c(x^k, z^k)\right|$
   and $\left|c(x^k\alpha^k\delta^k, z^k + \alpha^k q^k)\right| > \eta$,
   then reduce α^k until $\left|c(x^k + \alpha^k\delta^k, z^k + \alpha^k q^k)\right| < \left|c(x^k, z^k)\right|$.

6. Set

$$\begin{aligned} x^{k+1} &:= x^k + \alpha^k\delta^k \\ z^{k+1} &:= z^k + \alpha^k q^k. \end{aligned}$$

7. If $\psi_d(x^{k+1}, z^{k+1}) > \psi_d(x^k, z^k)$
   then set $\eta = \frac{1}{2}\left|c(x^k, z^k)\right|$.

8. Update B^k.

9. If convergence criteria are met, stop.

10. Set $k := k + 1$; goto 2.

Under our assumptions, the convergence of the Algorithm can be proved using the above results and the techniques in [BogT89]. In the next section, we discuss some modifications of the Algorithm for functions that do not satisfy these conditions. Here we give a brief sketch of the proof of global convergence. We begin by defining

$$\mathcal{C}(\eta) = \{(x,z) : \ |c(x,z)| \le \eta\}\,.$$

It can be shown that once η is sufficiently small, all iterates will remain in $\mathcal{C}(\eta)$, even though once in $\mathcal{C}(\eta)$ the value of $|c(x,z)|$ can increase in a controlled way. As in [BogT89] we can show that for η sufficiently small, a sufficient reduction in $\psi_d^k(x,z)$ implies a sufficient reduction in $\psi_d(x,z)$, and local convergence follows.

Steps 5, 6, and 7 of the Algorithm ensure that η will become small enough and that the iterates will eventually remain in $\mathcal{C}(\eta)$. Global convergence now follows if we assume that the algorithm cannot produce a sequence of iterates that becomes unbounded. As in our previous paper, we simply assume that the iterates remain in a compact set, but see that work for a discussion of conditions on the functions that imply this. The proof of the following theorem is a formalization of the above ideas.

Theorem 3.1. *Let assumptions* **A1**–**A3** *hold and assume that the sequence of iterates generated by the Algorithm remains in a compact set. Then the limit points of the sequence are local solutions of (NLP).*

Some comments are in order. First, it is possible to ensure that $\psi_d(x,z)$ is well-defined at each point encountered in the algorithm by requiring z to remain positive throughout the course of the computations (See Proposition 1.) This can be done by redefining the update for z to be

$$z^{k+1} = z^k + \alpha(1-\sigma)q^k,$$

for any $\sigma \in (0,1)$. An examination of the theory of this and the preceding section shows that for σ small this change does not alter the results. Thus, even if the quadratic programming subproblem at some iteration does not have a strong solution, the merit function is still well-defined and a move away from such a point is possible. For example, one could simply reduce constraint infeasibilities, or move in a descent direction of the merit function.

Second, the quadratic subproblem may be infeasible. In this case, a move as just described is still possible. There may be, however, alternate possibilities depending on the action of the quadratic programming solver. For example, in the large scale case described in [BogTK92] the interior-point solver always produces a descent direction for $\psi_d^k(x,z)$ even if the quadratic program is infeasible.

4. Numerical Results

The algorithm described in this paper was implemented in Fortran and tested on a selection of problems from [HocS81] and [Sch87]. The problems chosen were the higher dimensional ones with a large number of nonlinear inequality constraints. (The low dimensional problems converged too quickly to be of much interest.) Because equality constraints can be handled in a straightforward manner [BogT89], no problems with equality constraints were chosen. Several starting points were used, including those suggested in the references. The tests were run on a **SPARCstation 2** workstation in double precision arithmetic (16 decimal digits). The quadratic subproblems were solved using the Nag subroutine E04NAF.

To implement the Algorithm of §3 requires numerous important decisions that affect the performance. We thus provide some relevant details of the actual implementation.

Because we want no more than a single call per iteration to the subroutine E04NAF, no special action is taken when an inconsistent QP is encountered, i.e., our procedure merely stops. In small, well-scaled problems such as the ones solved here, this difficulty only occurred once. For large scale problems, more sophisticated techniques can be employed to handle an inconsistent QP [BogTK92].

We found it important to update the approximation of the Hessian of the Lagrangian at each iteration, while still maintaining positive definiteness. Accordingly we use the numerically proven damped BFGS update of [Pow78], defined as follows. In the standard BFGS one uses the vectors

$$\begin{aligned} s &= x^{k+1} - x^k \\ y &= \nabla_x L(x^{k+1}, \lambda^{k+1}) - \nabla_x L(x^k, \lambda^{k+1}) \end{aligned}$$

in the update formula. Since $s^{\mathsf{T}} y$ need not be positive, Powell recommends replacing y by

$$y^P = \theta y + (1-\theta) B_k s$$

where the parameter $\theta \in (0,1]$ is chosen as

$$\theta = \begin{cases} 1 & \text{if } s^T y \geq \zeta s^T B^k s \\ (1-\zeta) s^{\mathsf{T}} B_k s / (s^{\mathsf{T}} B_k s - s^T y) & \text{otherwise.} \end{cases}$$

and the value of ζ was chosen to be (.1). More theoretically justified updating schemes exist, e.g., [Tap88], but because of their local nature, they are not employed here.

A pre-update scaling strategy suggested for unconstrained optimization [OreS76] is used. This pre-update scaling is only performed on the initial iteration as suggested by [ShaP78]. The matrix B^0 is set to I and the first step is computed; s and y are calculated. Before the above updating procedure is used, however, B^0 is set to τI where

$$\tau = \begin{cases} s^T y / s^T s & \text{if } s^T y > 0 \\ 1 & \text{otherwise} \end{cases} .$$

In the discussion of global convergence, we assume knowledge of an appropriate value of d. In practice this is seldom the case. Accordingly, a dynamic scheme for adjusting the parameter d was devised and employed. We compute an initial value of d by setting

$$d = \max\{1, \psi_2 / |\psi_1|\}$$

where

$$\begin{aligned} \psi_1 &= f(x^k) + (g(x^k) + z^k)^\mathsf{T} \bar{\lambda}^k \\ \psi_2 &= (g(x^k) + z^k)^\mathsf{T} \bar{A}_k^{-1} (g(x^k) + z^k). \end{aligned}$$

This has the effect of "balancing" the two terms in the merit function. We then decrease d at each iteration, if necessary, to ensure that the directional derivative of $\psi_d^k(x, z)$ is sufficiently negative in the direction (δ^k, q^k). We compute the directional derivative, w,

$$w(d) = \frac{(\nabla \psi_d^k)^\mathsf{T} \begin{pmatrix} \delta^k \\ q^k \end{pmatrix}}{\left| \nabla \psi_d^k \right| \cdot |(\delta, q)|}.$$

If $w(d)$ is not sufficiently negative, we reduce d. In particular, if $w(d) \geq (-.1)$, we reduce d until $w(d) = (-.5)$. (This procedure is safeguarded by not allowing d to decrease by more than a factor of 10 at any iteration.) Our procedure never allows d to increase, and only decreases it when necessary. If d were to be decreased frivolously it could become too small causing the iterates to 'hug' the constraints, hence impeding convergence. For this reason, the updating scheme for d is intentionally conservative. (See Remarks 5 and 6 below.)

It may happen that an unacceptably small steplength must be used to reduce ψ_d^k. In this case we abandon our effort to reduce ψ_d^k and compute a steplength that only is required to reduce $|\bar{c}(x, z)|$. In the tests reported below, this has always allowed the algorithm to continue successfully.

We use standard convergence criteria. We first insist that the constraints be satisfied to a close tolerance; specifically we require

$$g_i(x^k) \leq 10^{-7}.$$

We also require that either

$$\frac{\left|\nabla f(x^k) + \nabla g(x^k)\lambda^k\right|}{|f(x^k)|} \leq 10^{-7} \tag{4.1}$$

or

$$\left|x^k - x^{k-1}\right| \leq 10^{-8}(1 + \left|x^k\right|). \tag{4.2}$$

As noted above the test problems are taken from [HocS81] and [Sch83]. In Table 1 we give the problem number from these references, some relevant information about the problem, and the performance of the algorithm on that problem. In particular, we show the dimension, n, of the problem; the number of constraints, m; the number of those constraints that are simple bounds, b; and the number of active constraints at the solution. For each problem we consider several starting values, several *fixed* values of the parameter d and the dynamic choice of d discussed earlier. Then for each of these runs we report the number of iterations (which is equal to the number of evaluations of $\nabla f(x)$ and $\nabla g(x)$); the number of function and constraint evaluations; the iteration, $iter_{act}$, at which the correct active set is identified; the number of times that η had to be reduced (see Step 6); the number of times that $\psi_d^k(x, z)$ could not be reduced (see Step 4); and the convergence criteria satisfied where 1 implies that test (4.1) is satisfied, 2 implies that (4.2) is satisfied, 3 implies that both are, 7 implies numerical overflow in a function evaluation, and 8 signifies that an inconsistent QP was detected.

We make several observations concerning these results.

1. No attempt was made to optimize the code. For example, simple bounds should be handled separately and not be included in the merit function since they will always be satisfied. In the same spirit, the gradients of linear constraints should only be computed once. Such enhancements would not, of course, change the number of iterations.

2. We did not attempt a detailed comparison with other methods. Direct comparisons are confounded by a host of factors such as the updating strategy, line search, convergence criteria, and differences in the machines and compilers on which the codes were compiled and run. We note, however, that

our results on test problems using the "standard" starting values are similar to those of [HocS81] and [Sch83]. Furthermore, since our motivation is ultimately to solve large scale problems, the performance of this algorithm validates the basic approach.

3. The merit function performs well. Most of the steps have $\alpha^k = 1$, so no other merit function would do substantially better. If the set of constraints gradients became linearly dependent, the performance of the algorithm did deteriorate. In this case, as is theoretically expected, we failed to observe q-superlinear convergence, but the last steps preceding convergence were always full steps.

4. Our procedure for η caused no apparent problems; the code performed as in the equality constrained case reported in [BogT89].

5. The test problems were both small and well-scaled, hence the active set of constraints did not change often. The correct active set was identified in early iterations, and the total number of iterations was usually quite small. For this reason, many performance enhancing components of the algorithm were not relevant for solving these problems.

6. The algorithm does not appear to be overly sensitive to the parameter d, i.e., slightly different strategies for adjusting d did not give dramatically different results. We attempted to test this by multiplying the objective function by 1000. Solving the resulting, poorly scaled problems did cause some slight difficulty, but only when the parameter d was a fixed constant. Even in the poorly scaled problems a constant d would suffice, provided it was chosen according to our initialization procedure, but the dynamic scheme eliminated this difficulty completely. A more sophisticated updating of the parameter d has been successfully implemented in [BogTK92].

7. There were several instances where the line search in Step 4 failed and the procedure for reducing $|\bar{c}(x,z)|$ was invoked, allowing the algorithm to continue. This occurred primarily when initial points were far from the solution.

In conclusion, we believe that the merit function and the algorithm described here are competitive procedures for solving general nonlinear programming problems. More importantly, extensions of this algorithm to large sparse problems appear promising.

February 10, 1993

REFERENCES

[BogT84] Boggs, P., Tolle, J., *A family of descent functions for constrained optimization,* SIAM J. Numer. Anal. 21 (1984), pp. 1146-1161.

[BogT88] Boggs, P., Tolle, J., *Merit functions and nonlinear programming,* in Operational Research '87, G. K. Rand, ed., Elsevier Science Publishers B.V., North Holland, 1988.

[BogT89] Boggs, P., Tolle, J., *A strategy for global convergence in a sequential quadratic programming algorithm,* SIAM J. Numer. Anal. 26 (1989), pp. 600-623.

[BogT89] Boggs, P., Tolle, J., *Convergence properties of a class of rank-two updates,* (To appear in SIAM J. Optimization).

[BogTK92] Boggs, P., Tolle, J., Kearsley, A., *A Truncated SQP Algorithm for Large Scale Nonlinear Programming Problems,* Technical Report 4800, NIST, Gaithersburg, MD, 1992.

[BogTW82] Boggs, P., Tolle, J., Wang, P., *On the local convergence of quasi-Newton methods for constrained optimization,* SIAM J. Control Optim., 20 (1982), pp. 161-171.

[ChaLLP82] Chamberlain, R., Lemarechal, C., Pedersen, H. C., and Powell, M., *The watchdog technique for forcing convergence in algorithms for constrained optimization,* Math. Programming Stud. 16 (1982), pp. 1-17.

[Fle72] Fletcher, R., *A class of methods for nonlinear programming III: rates of convergence,* in Numerical Methods for Nonlinear Optimization, F. A. Lootsma, ed., Academic Press, New York, 1972, pp. 371-382.

[GilMSW86] Gill, P., Murray, W., Saunders, M., and Wright, M., *Some theoretical properties of an augmented Lagrangian merit function,* Technical Report 86-6, Stanford University, Dept. Operations Research, 1986.

[Han77] Han, S., *A globally convergent method for nonlinear programming,* J. Optim. Theory Appl. 22 (1977), pp. 297-309.

[HocS81] Hock, W., Schittkowski. K., *Test examples for nonlinear programming codes*, in Lecture Notes in Economics and Mathematical Systems 187, Springer-Verlag, Berlin, 1981.

[OreS76] Oren, S., Spedicato, E.,*Optimal conditioning of self-scaling variable metric algorithms,* Math. Programming, 10 (1976), pp. 70-90.

[Pow78] Powell, M. J. D., *A fast algorithm for nonlinearly constrained optimization calculation.* in Numerical Analysis Proceedings Dundee 1977, G. A. Watson, editor, Springer-Verlag, 1978, pp. 144-157.

[Pow84] Powell, M. J. D., *The performance of two subroutines for constrained optimization on some difficult test problems,* in Numerical Optimization 1984, P. T. Boggs, R. H. Byrd, and R. Schnabel, editors. SIAM, 1984.

[PowY86] Powell, M. J. D., Yuan, Y., *A recursive quadratic programming algorithm that uses differentiable exact penalty functions,* Math. Programming, 35 (1986), pp. 265-278.

[Rob74] Robinson, S., *Perturbed Kuhn-Tucker points and rates of convergence for a class of nonlinear-programming algorithms,* Math. Programming 7 (1974), pp. 1-16.

[Sch83] Schittkowski, K., *On the convergence of a sequential quadratic programming method with an augmented Lagrangian line search function,* Math. Operationsforsch. Statist. Ser. Optim. 14 (1983), pp. 197-216.

[Sch87] Schittkowski, K., *More test examples for nonlinear programming codes*, in Lecture Notes in Economics and Mathematical Systems 282, Springer-Verlag, Berlin, 1987.

[ShaP78] Shanno, D., Phua, K., *Matrix conditioning and nonlinear optimization*, Math. Programming 14 (1978), pp. 145-160.

[ShaP89] Shanno, D., and Phua, K., *Numerical experience with sequential quadratic programming algorithms for equality constrained nonlinear programming*, acm Trans. Math. Soft. 15 (1989), pp. 49 - 63.

[Tap77] Tapia, R., *Diagonalized multiplier methods and quasi-Newton methods for constrained optimization*, J. Optim. Theory Appl. 22 (1977), pp. 135-194.

[Tap88] Tapia, R., *On secant updates for use in general constrained optimization*, Math. Comp. 51 (1988), pp. 181-202.