A TRUNCATED SQP ALGORITHM FOR LARGE SCALE NONLINEAR PROGRAMMING PROBLEMS *

PAUL T. BOGGS
Applied and Computational Mathematics Division, National Institute of Standards and Technology, Gaithersburg, MD 20899

JON W. TOLLE
Mathematics Department, University of North Carolina, Chapel Hill, NC 27599

and

ANTHONY J. KEARSLEY
Department of Computational and Applied Mathematics, Rice University, Houston, Texas 77251

Abstract. We consider the inequality constrained nonlinear programming problem and an SQP algorithm for its solution. We are primarily concerned with two aspects of the general procedure, namely, the approximate solution of the quadratic program, and the need for an appropriate merit function. We first describe an (iterative) interior-point method for the quadratic programming subproblem that, no matter when it is terminated, yields a descent direction for a suggested new merit function. An algorithm based on ideas from trust-region and truncated Newton methods is suggested and some of our preliminary numerical results are discussed.

1. Introduction

Large scale optimization problems are gradually submitting to the power of advanced algorithmic development and of modern computing environments, leading to the formulation of models requiring solutions of these problems in a variety of scientific areas. Two excellent recent reviews are given by Coleman (Coleman, 1992) and Conn, Gould and Toint (Conn *et al.*, 1992), who survey some important applications as well as recent trends in algorithms and consider the impact of parallel computing architectures for large scale optimization.

Following these authors we take the term *large scale* to mean any optimization problem that is large enough so that the exploitation of special structure is important. In this paper we are particularly concerned with sparsity, although, as they point out, other problem structures may be important as well. We assume the general nonlinear programming problem to be of the form

$$\begin{array}{l} \min\limits_x f(x) \\ \text{subject to: } \; g(x) \le 0 \end{array} \qquad (NLP)$$

where $f : \Re^n \to \Re^1$, and $g : \Re^n \to \Re^m$. We note that we could include nonlinear equality constraints in (NLP) without incurring any analytical difficulties, but at the expense of distracting technicalities. We thus omit them for the purposes of the exposition here, but they have been included in our program.

Our basic tool for the solution of (NLP) is the sequential quadratic programming (SQP) algorithm in which, given an approximate solution-multiplier pair, (x^k, λ^k), (NLP) is approximated by a quadratic program of the form

$$\begin{aligned} &\min_{\delta} \nabla f(x^k)^T \delta + \tfrac{1}{2}\delta^T B^k \delta \\ &\text{subject to:} \quad \nabla g(x^k)^T \delta + g \leq 0. \end{aligned} \tag{QP}$$

Here B^k is taken to be an approximation to the Hessian of the Lagrangian for (NLP), i.e., for

$$\ell(x, \lambda) = f(x) + g(x)^T \lambda$$

we choose

$$B^k \approx \nabla^2_{xx} \ell(x^k, \lambda^k).$$

In this form the solution to (QP) provides a search direction for improving the current iterate, x^k. A steplength is chosen in this direction so as to reduce a *merit function.* Roughly speaking, a merit function is a scalar valued function with a minimum at the solution to (NLP). Thus reducing this function ensures progress and allows for the construction of a globally convergent scheme. (See e.g., (Boggs and Tolle, 1989) and (Boggs *et al.*, 1991).) In a previous paper, (Boggs *et al.*, 1991), the authors introduced a merit function for (NLP) and showed that it is appropriate for use with the SQP algorithm. In this paper we apply these ideas to the large scale case, solving (QP) only approximately by an iterative interior-point algorithm that we can stop prematurely. Such ideas are in the spirit of *truncated* or *inexact* Newton methods. (See (Dembo *et al.*, 1982) and, for a recent discussion of these methods, (Eisenstat and Walker, 1991).)

To be more specific, we use the interior-point quadratic program solver of (Boggs *et al.*, 1992). At each iteration this method constructs a low-dimensional subspace and solves (QP) restricted to that subspace. We can show that halting this procedure after any number of steps yields a descent direction for the merit function. The details of this solver and its properties relative to its use in an SQP algorithm are discussed in §2.

The actual merit function and a related approximate merit function are reviewed in §3. We then state the results just mentioned, namely that the inexact directions are compatible with these functions. In §4 we give the details of the algorithm. One of the problematic points is how to control the number of iterations on (QP). Here we use some ideas from *trust region* methods. We attempt to assess how well (QP) approximates the behavior of the merit function by maintaining an estimate of a trust region radius. §4 also contains a summary of the results of some numerical experimentation with the algorithm on a few large problems of interest. Our results indicate that our procedure is viable for large scale problems. Suggestions for further research are contained in §5.

2. An Interior-Point QP Solver

Interior-point methods for linear programming have been demonstrated to be very successful, especially on large problems; thus it is natural to consider their extension to quadratic programs (QP). One method that has performed well on linear programs, and has been extended to QP with both good numerical results and particularly interesting properties with respect to the SQP method is the *optimal subspace* method of Boggs, et al. (Boggs *et al.*, 1992). (See also (Domich *et al.*, 1991).) We take the QP of §1 to be of the form

$$\begin{array}{l} \min\limits_{\delta} c^T\delta + \frac{1}{2}\delta^T Q\delta \\ \text{subject to:} \;\; A^T\delta + b \le 0 \end{array} \tag{1}$$

where $c, \delta \in \Re^n$, $Q \in \Re^{n\times n}$, $A \in \Re^{n\times m}$, and $b \in \Re^m$.

The assumptions on (1) that are necessary to apply the algorithm are that the problem be bounded; that A have full column rank; that there exist feasible points (i.e., that the constraints be consistent); and that Q be positive semidefinite. Note that a full dimensional interior is not required. We comment further on these assumptions at the end of this section.

Briefly, the general algorithm can be expressed as follows.

O3D Algorithm for Quadratic Programming

1. Given a feasible point, δ^0; set $j := 0$.
2. Generate 3 independent search directions

$$p_i, \qquad i = 1, \ldots, 3.$$

   Let P^j be the matrix whose columns are p_i.
3. Form and solve the restricted quadratic program

$$\begin{array}{l} \min\limits_{\zeta} c^T\tilde{\delta} + \frac{1}{2}\tilde{\delta}^T Q\tilde{\delta} \\ \text{subject to:} \;\; A^T\tilde{\delta} + b \le 0 \end{array}$$

   where $\tilde{\delta} = \delta^j + P^j\zeta$, and $\zeta \in \Re^3$. Call the solution ζ^*.
4. Set $\delta^{j+1} := \delta^j + \rho P^j \zeta^*$ for an appropriate value of the steplength ρ.
5. If stopping criteria are met, set $J = j$, $\delta_J = \delta^j$ and exit.
6. Go to 2.

The details of the actual algorithm can be found in (Boggs *et al.*, 1992) and (Domich *et al.*, 1991); here we describe those that are the most important for its application in the SQP setting. One of the three directions is always a descent direction with respect to the objective function, thus assuring descent in the objective value at each step. Specifically, the algorithm uses directions that are solutions to

$$\left[AD^2A^T + Q/\beta\right] p_i = t_i \tag{2}$$

where β is a positive scalar depending on the current iterate,

$$D = \operatorname{diag}\left\{1/r_k,\ k = 1, \ldots, m\right\},$$

$r_k = -(A_k^T \delta^j + b_k)$, and t_i is a particular right hand side. The form of the matrix in (2) allows for efficient exploitation of the sparsity. Note that if Q is positive semi-definite, then this matrix is positive definite for all interior points. The steplength ρ (cf. Step 4) is set either to obtain the optimal solution in the given direction or to advance 99% of the distance to the boundary.

An important aspect of the algorithm is the procedure for obtaining an initial feasible point, since we certainly do not require that a feasible point be given. The algorithm uses a "Big M" method to construct the Phase I problem:

$$\begin{aligned} &\min_{\delta,\theta} \; c^T\delta + \tfrac{1}{2}\delta^T Q\delta + M\theta \\ &\text{subject to:} \quad A^T\delta + b - e\theta \leq 0 \end{aligned} \tag{3}$$

where e is a vector of all ones and θ is the "artificial" variable. Clearly for θ^* large enough the point $(\delta, \theta) = (0, \theta^*)$ is feasible for (3). The above procedure is thus used until the artificial variable is negative, at which point the current value of δ is feasible, and the $M\theta$ and $e\theta$ terms are dropped. If no such value of the artificial variable can be found, then the QP is not consistent, i.e., no feasible point exists, and the algorithm stops. In this case, however, one can show that the optimal solution satisfies

$$\theta = \min_{\delta} \; \max_{j} \{A_j^T\delta + b_j\},$$

and the resulting δ is a reasonable direction for (NLP).

The criteria for convergence of the algorithm are that at least one of the following hold: (a) the relative change in two successive values of the objective function is small; (b) the relative difference between the primal and the dual objective function values is small; or (c) the relative difference between two successive iterates is small. To this list, we have added the criterion (d) the scaled solution vector exceeds a specified length. This last condition has been implemented to allow the use of trust region strategies to monitor the quality of the (QP) approximation. In particular, this procedure will cause the algorithm to halt if (QP) is unbounded, again with a reasonable direction.

Note that the assumptions set forth above ensure that a solution to (3) exists, but that the quadratic subproblems arising in the SQP algorithm may not have solutions. Nevertheless, the directions calculated by O3D are useful directions in the solution of (NLP).

3. The Merit Function

A merit function for (NLP) is typically a scalar valued function that has an unconstrained minimum at x^*, the solution to (NLP). Thus a reduction in this function implies that progress is being made towards the solution.

In (Boggs *et al.*, 1991) we derived a merit function for (NLP) based on the work in (Boggs and Tolle, 1984) and (Boggs and Tolle, 1989) for equality constrained problems. This was done by considering the slack variable problem (see (Tapia, 1980))

$$\begin{aligned} &\min_{x,s} \; f(x) \\ &\text{subject to:} \quad g(x) + \tfrac{1}{4}S^2 e = 0 \end{aligned} \tag{4}$$

where

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

The merit function in (Boggs and Tolle, 1984) was then applied to (4). Since the resulting merit function only contained references to s_i^2, and not to just s_i, it was natural to rephrase this merit function in terms of $z_i = s_i^2$. This led to the rather unusual situation of having a *constrained* merit function, i.e., a merit function whose constrained minimum corresponds to the solution of (NLP). Our merit function is

$$\psi_d(x,z) = f(x) + \bar{\lambda}(x,z)^T \bar{c}(x,z) + \tfrac{1}{d}\bar{c}(x,z)^T \mathcal{A}(x,z)^{-1}\bar{c}(x,z) \tag{5}$$

where d is a scalar,

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

and

$$Z = \mathrm{diag}\,\{z_1, \ldots, z_m\}$$

with the z vector constrained to be nonnegative. Although the merit function is constrained, our algorithm ensures that the z_i always remain positive; thus the bounds present neither a theoretical nor a computational difficulty. For a direction, δ^k, in x obtained as the solution to (QP), we take the direction for the change in z to be

$$q^k = -[\nabla g(x^k)\delta^k + g(x^k) + z^k].$$

Thus the next step is

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

for some value of α. Observe that if $z^k \geq 0$ and δ^k is feasible, then

$$q^k + z^k = -[\nabla g(x^k)^T \delta^k + g(x^k)] \geq 0 \tag{6}$$

and it follows that for $\alpha \in (0,1]$, $z^{k+1} = z^k + \alpha q^k \geq 0$.

We show in (Boggs *et al.*, 1991) that ψ_d has certain desirable properties for d sufficiently small. First, under mild conditions, a constrained minimum of ψ_d corresponds to a solution of (NLP). Furthermore, the directions (δ^k, q^k) are descent directions for ψ_d for (x^k, z^k) sufficiently close to feasibility; a steplength of one is acceptable near the solution if the method is converging q-superlinearly; and the directions are always descent directions for $r(x,z) = ||\bar{c}(x,z)||^2$.

Despite these useful properties, ψ_d has two deficiencies that preclude using it directly in an algorithm. First, as stated above, δ^k is only a descent direction near feasibility, and second, it requires the evaluation of gradients and other nontrivial computation to assess a prospective value of α. Thus we employ an approximate merit function and a globalization strategy that overcome these deficiencies. We use

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

where

$$\begin{aligned}\mathcal{A}^k &= \nabla g(x^k)^T \nabla g(x^k) + Z^k \\ \bar{\lambda}^k &= -(\mathcal{A}^k)^{-1} \nabla g(x^k)^T \nabla f(x^k).\end{aligned}$$

We show in (Boggs *et al.*, 1991) that (δ^k, q^k) is a descent direction for ψ_d^k everywhere, that ψ_d^k will not interfere with rapid local convergence, and that the globalization strategy described in §4 is effective.

The main theoretical result described here is that O3D and the merit function are compatible. Specifically, if δ_J^k is only a partial solution to (QP) obtained by J iterations of O3D (see step 5), the above results continue to hold. We state the assumptions that guarantee this. We use the term *strong local solution* to mean an optimal point, together with a multiplier vector, of (NLP) at which the following hold.

A1: The active constraint gradients are linearly independent.

A2: Strict complementary slackness holds.

A3: The second order sufficient conditions hold.

In addition we make the following assumptions on the (QP) subproblems:

A4: The matrices $\{B^k\}$ are uniformly positive definite.

A5: For each k (QP) has a strong local solution.

We also need an assumption that guarantees that the merit function is well defined i.e., that $\mathcal{A}$ is nonsingular. As in (Boggs *et al.*, 1991) we formulate this by partitioning the index set of the constraints into two subsets a and u. We can then write, without loss of generality,

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

and correspondingly,

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

Usually the index subset a will correspond to the set of active constraints for (NLP) or (QP). The necessary assumption in terms of a particular partition is the following.

A6: The set $\{\nabla g_i(x) \ : \ i \in a\}$ is linearly independent and $z_u > 0$.

A discussion of the implications of these assumptions for SQP algorithms is given in (Boggs *et al.*, 1991). The proofs of the results make use of the techniques in (Boggs *et al.*, 1991) combined with an induction argument.

4. Algorithm and Numerical Results

A brief statement of the final algorithm is as follows. Following the statement, we give a brief discussion of some important points.

SQP Algorithm

1. Given x^0, τ (trust region radius), η (globalization parameter), and d (merit function parameter):
   Set $k := 0$.
2. Using O3D, iterate while $||\delta|| < \tau$ on

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

   to obtain δ^k.
3. Set $q^k = -[\nabla g(x^k)^T \delta^k + g(x^k) + z^k]$.
4. (Globalization Step)
   Choose α^k such that ψ_d^k is reduced.
   If $||\bar{c}(x^k + \alpha^k \delta^k, z^k + \alpha^k q^k)|| \geq ||\bar{c}(x^k, z^k)||$ and $||\bar{c}(x^k, z^k)|| > \eta$,
   reduce α^k until $||\bar{c}(x^k + \alpha^k \delta^k, z^k + \alpha^k q^k)|| < ||\bar{c}(x^k, z^k)||$.
5. If $\psi_d(x^k + \alpha^k \delta^k, z^k + \alpha^k q^k) > \psi_d(x^k, z^k)$
   set $\eta = \frac{1}{2}||\bar{c}(x^k, z^k)||$.
6. Set

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

7. If convergence criteria are met, quit.
8. Adjust τ.
9. Set $k := k + 1$; goto 2.

A few comments are necessary. First, the globalization step is based on the work in (Boggs and Tolle, 1989). In brief, η is an estimate of the radius of the domain containing the feasible region in which the true merit function, $\psi_d(x, z)$, is reduced in the direction (δ, q). For all iterates, the algorithm first requires that the approximate merit function be reduced. If the current iterate lies outside the η-domain, then the algorithm also requires that the constraint infeasibilities be reduced. If the iterate lies inside the η-domain then the true merit function should also be reduced; if not, then η is reduced. This allows steps that may increase the merit function, but only in a controlled way. The steps that increase the merit function are usually seen only in early iterations or after active set changes. Second, our procedure for updating τ is to compute the predicted relative reduction of the merit function based on the (QP) and compare that with the actual relative reduction. This comparison of predicted and actual reductions is done using the approximate merit function if the current iterate lies outside the η-domain. The true merit function is employed otherwise. We then use standard updating strategies to adjust τ (see e.g. (Dennis and Schnabel, 1983) or (Moré and Sorensen, 1983)). Third, the penalty parameter,

d, is updated in a very straightforward manner. Essentially, an estimate of the condition of the problem is monitored. In the event that this estimate increases significantly, d is decreased. Provided that the initial value of d is reasonable, this updating does not occur often, and is only observed when the iterates are outside the η-domain. The iterates do not 'hug', or stick too closely to the constraint manifold, as is the case when the penalty parameter becomes too small. Computationally, this simple procedure for updating d appears to be effective even in the presence of highly nonlinear constraints.

We have used this procedure to solve several problems in the range of 100–500 variables with up to 500 constraints. Many of these problems have arisen from discretizations of control problems where the Hessian of the Lagrangian and the Jacobian of the constraints have some known sparsity structure. These problems are somewhat special, in that we know that the major expense in the calculation of an iterate comes from the solving of the (QP) . Typically, the constraints are nonlinear inequalities that, in some way, limit the control variables, and the objective function is an energy approximation. The number of constraints is greater than the number of variables in many of the problems we solved. In our testing, we use forward finite-difference approximations to gradients and Hessians, and modify the Hessian of the Lagrangian to be positive semidefinite in cases where it is not (e.g. (Gill *et al.*, 1981)). This latter procedure requires the addition of a non-negative diagonal matrix to the Hessian approximation.

Our observations include the following.

- The number of major iterations is reasonable.
- The globalization procedure remains efficient, i.e. many full steps are accepted.
- Close to the solution, the trust region becomes inactive.
- The trust-region strategy is basically effective, i.e., it prevents long, unprofitable steps from being generated at the beginning and after the active set changes.
- In general a small number of iterations of O3D suffices at each major iteration, and a very small number near the solution.
- As in all of our previous work in this area, the parameter d in the merit function is not critical, i.e., the performance of the algorithm is not changed much by changes in the strategy for adjusting d.

5. Future Research

We have described a preliminary version of an extension of the SQP ideas to the large scale case. In doing so, we have used a combination of an interior-point method for solving (QP) with trust region and truncated Newton methods to create a promising algorithm. There are, however, many computational and theoretical aspects of this algorithm that need further analysis and testing. Computationally, we need to continue to test the procedure to discover its strengths and weaknesses. At the same time, the limitation on the QP solver, O3D, that the Hessian must be positive semidefinite, appears to be surmountable. In particular, we believe that the procedure for solving the reduced quadratic program (step 3 of O3D) can be modified to handle an indefinite (or negative definite) Hessian. This would allow us to avoid the extra work of ensuring that the Hessian is positive definite, and to explore directions of negative curvature.

Our theoretical analysis described in §3 relies on the usual strong assumptions that are typically satisfied in the small scale case. In large problems, some of these assumptions are not as likely to be satisfied. In particular, large problems may be highly degenerate. We know that the interior-point algorithms for LP and QP are not affected by these cases and, although the theory does not apply, our SQP algorithm had no difficulty in solving such problems. Also, in some of the problems, inconsistent quadratic subproblems occurred. This, too, caused no difficulty for the algorithm, but is a problem for the theory. Thus, obtaining good theoretical results under a weakened set of assumptions is an important task for further research.

References

Boggs, P. T., Domich, P. D., Rogers, J. E., Witzgall, C., (1991) "An interior point method for linear and quadratic programming problems." *Mathematical Programming Society COAL Newsletter,* 19:32-40.

Boggs, P. T., Tolle, J. W., (1984) "A family of descent functions for constrained optimization" *SIAM J. Num. Anal.*, 26:1146 - 1161.

Boggs, P. T., Tolle, J. W., (1989) "A strategy for global convergence in a sequential quadratic programming algorithm." *SIAM Journal of Numer. Anal.*, 21:600-623.

Boggs, P. T., Tolle, J. W., Kearsley, A. J., (1991) "A merit function for inequality constrained nonlinear programming problems.", Technical Report 4702, NIST, Gaithersburg, MD, 4702.

Coleman, T. F., (1992) "Large Scale Numerical Optimization: Introduction and overview." In *Encyclopedia of Computer Science and Technology.* Marcel Dekker, Inc., New York (to appear).

Conn, A. R., Gould, N. I. M., Toint, Ph. T., (1992) " Large-scale nonlinear constrained optimization." Technical Report 92-02, Facultés Universitaires de Namur, Départment de Mathématique, Namur, Belgium.

Dennis, J. E., Schnabel, R. B., (1983) *Numerical Methods for Unconstrained Optimization and Nonlinear Equations.* Prentice-Hall, Englewood Cliffs, NJ.

Dembo, R. S., Eisenstat, S. C., Steihaug, T., (1982) "Inexact Newton methods." *SIAM J. Num. Anal.*, 19:400-408.

Domich, P. D., Boggs, P. T., Rogers, J. E., Witzgall, C., (1991) "Optimizing over three-dimensional subspaces in an interior point method for linear programming." *Linear Algebra and its Applications*,152

Eisenstat, S. C., Walker, H. F., (1991) "Globally convergent inexact Newton methods." Technical Report, Utah State University, Department of Mathematics, Logan, UT.

Gill, P., Murray, W., and Wright, M., (1981) *Practical Optimization.* Academic Press, New York.

Moré, J. J., Sorensen, D. C., (1983) "Computing a trust region step." *SIAM J. Sci. Stat. Comput.,* 4:553 - 572.

Tapia, R. A., (1980) "On the role of slack variables in quasi-Newton methods for unconstrained optimization." In *Numerical Optimization of Dynamical Systems*, L. C. W. Dixon and G. P. Szegö, eds., North-Holland, Amsterdam: 235-246.