Active Set and Interior Methods for Nonlinear Optimization

Richard H. Byrd and Jorge Nocedal

Abstract. We discuss several fundamental questions concerning the problem of minimizing a nonlinear function subject to a set of inequality constraints. We begin by asking: What makes the problem intrinsically difficult to solve, and which characterizations of the solution make its solution more tractable? This leads to a discussion of two important methods of solution: active set and interior points. We make a critical assessment of the two approaches, and describe the main issues that must be resolved to make them effective in the solution of very large problems.

1991 Mathematics Subject Classification: 65K05 90C30
Keywords and Phrases: nonlinear optimization, large-scale optimization, nonlinear programming

The most important open problem in nonlinear optimization is the solution of large constrained problems of the form

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & h(x) = 0 \\ & g(x) \leq 0, \end{array} \tag{1}$$

where the functions $f : R^n \to R,\ h : R^n \to R^m$ and $g : R^n \to R^t$ are assumed to be smooth.

Assuming that certain regularity assumptions hold, the solution of (1) is characterized by the Karush-Kuhn-Tucker conditions [4]. They state that any solution x^* must satisfy the system

$$\nabla f(x^*) + A_h(x^*)\lambda_h^* + A_g(x^*)\lambda_g^* = 0 \tag{2}$$
$$h(x^*) = 0 \tag{3}$$
$$g(x^*) \leq 0 \tag{4}$$
$$g(x^*)^T \lambda_g^* = 0 \tag{5}$$
$$\lambda_g^* \geq 0, \tag{6}$$

for some Lagrange multiplier vectors λ_h^* and λ_g^*. Here A_h and A_g denote the matrices whose columns are the gradients of the functions h and g. The first equation can be written as $\nabla_x L(x^*, \lambda^*) = 0$, where L is the Lagrangian function

$$L(x, \lambda) = f(x) + \lambda_h^T h(x) + \lambda_g^T g(x). \tag{7}$$

This mathematical characterization is, however, not suitable for computation because finding a pair (x^*, λ^*) that satisfies the Karush-Kuhn-Tucker system (2)-(6) is a very hard problem.

Indeed we could attempt to guess the *optimal active set*, i.e. the set of inequality constraints that will be satisfied as equalities at the solution x^*. Based on this guess, we could then replace (4) by a set of equalities, remove (5) and (6), and define all Lagrange multipliers corresponding to inactive inequality constraints to be zero. This transforms (2)-(6) into a system of nonlinear equations, which is much more tractable. Unfortunately, the set of all possible active sets grows exponentially with the number t of inequality constraints. Moreover, not all pairs (x, λ) satisfying the Karush-Kuhn-Tucker conditions are solutions of (1); some of them could be, for example, maximizers. Therefore this type of approach can only be practical if we make intelligent guesses of the active set. We will return to this question below.

The fact that it is impractical to solve the Karush-Kuhn-Tucker system directly has given rise to a variety of constrained optimization methods which make use of two fundamental ideas:

transformation and approximation.

In the rest of the paper we describe how these ideas are used in some of the most powerful methods for nonlinear optimization.

1 Exact Penalty Functions

A very appealing idea is to replace (1) by a single unconstrained optimization problem. At first glance this may seem to be impossible since the general nonlinear optimization problem (1) must be much more complex than the minimization of any unconstrained function.

Nevertheless, several "exact penalty functions" have been discovered [4], and can be used in practice to solve nonlinear programming problems. The best example is the ℓ_1 penalty function

$$\psi(x; \rho) = f(x) + \rho \sum_{i=1}^{m} |h_i(x)| + \rho \sum_{i=1}^{t} g_i^+(x), \tag{8}$$

where $a^+ = \max\{0, a\}$. Here ρ is a positive penalty parameter whose choice is problem dependent. One can show that if the value of ρ is large enough, then local solutions of the nonlinear program (1) are normally local minimizers of (8).

The beauty and simplicity of this approach is undeniable. But it has two drawbacks. First of all, the function ϕ function is not differentiable, and thus minimizing it is far more difficult than minimizing a smooth function. One could use the tools of non-differentiable optimization, but an approach that may be much more effective is to make linear-quadratic approximations of ϕ, and use them to generate a series of estimates of the solution [4]. Interestingly enough, this leads to a method that is closely related to the active set method described in the next section.

The second drawback may be potentially fatal: the approach appears to be very sensitive to the choice of the penalty parameter ρ. Small values of ρ may lead to unbounded solutions, and excessively large values will slow the iteration because the nonlinear constraints will be followed closely. It is interesting that even though this exact penalty approach [4] was proposed more than 15 years ago, it has not yet been firmly established whether the difficulty in choosing the penalty parameter is serious enough to prevent it from becoming a powerful technique for large-scale optimization.

There is another open question concerning this, and most other methods for constrained optimization. It concerns the use of a merit function to determine whether a step is acceptable. We could regard a step p to be acceptable only if it gives a reduction in ψ. Some analysis, as well as numerical experience indicates that this strategy may be overly conservative and that it may be preferable to allow controlled increases in the merit function. How to do this is still an active area of research; an interesting recent proposal is described in [5].

2 Active Set Methods

Let us now consider a different approach, which is based on the strategy of making a series of intelligent guesses of the optimal active set, mentioned in the introduction.

Suppose that x is an estimate of the solution of (1) and that we wish to compute a displacement p leading to a better estimate $x^+ = x + p$. We can do this by making a linear-quadratic approximation – but this time of the original problem (1) — and solving the following subproblem in the variable p,

$$\begin{array}{ll} \text{minimize} & \nabla_x L(x,\lambda)^T p + \frac{1}{2} p^T \nabla^2_{xx} L(x,\lambda) p \\ \text{subject to} & h(x) + A_h(x)^T p = 0 \\ & g(x) + A_g(x)^T p \le 0. \end{array} \tag{9}$$

This subproblem is much more tractable than (1). In fact, if $\nabla^2_{xx} L(x,\lambda)$ is positive definite, then (9) is not much more difficult to solve than a linear program. For this reason it is common to either modify $\nabla^2_{xx} L(x,\lambda)$, so that it is always positive definite in the null space of constraints, or to replace it — directly or indirectly — by a positive definite approximation. (A recently developed algorithm [5]) deviates from this standard practice by formulating indefinite quadratic programming subproblems, but it is too early to determine if it will supersede the current approaches.)

The step p is considered to be acceptable only if it leads to a reduction in a *merit function*. An example of such a merit function is (8), but many other choices that combine constraint satisfaction and objective function decrease are possible [4]. This method is called *Sequential Quadratic Programming* and is currently regarded as the most powerful active set method.

There is a good mathematical justification [9, 8] for generating steps by means of the quadratic subproblem (9). One can show that the step is a direction of descent for a variety of merit functions. Moreover, the model (9) has the precise balance between constraint satisfaction and decrease in the objective function. Unlike approaches, such as reduced gradient methods, that attempt to satisfy the original constraints of the problem at each step (which can be computationally very demanding) the quadratic programming model (9) applies successive linearizations to the constraints – which is the idea behind Newton's method for solving equations. Thus we can expect that the iterates generated by this active set approach will decrease a measure of feasibility at a quadratic rate.

There are really two different ideas in the method we have just described. The first is to use the subproblem (9) to provide us with an informed guess of the optimal active set: our guess is the active set identified in the solution of the quadratic subproblem. The second idea is to use the right level of approximation to the objective function and constraints, as discussed above. In the interior methods described next, we no longer attempt to guess the optimal active set, but retain the idea of making linear-quadratic approximations.

3 Interior Point Methods

Let us use slack variables s to transform (1) into the following equivalent problem in the variables x and s,

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subjet to} & h(x) = 0 \\ & g(x) + s = 0 \\ & s \geq 0. \end{array}$$

Even though the only inequalities are now simple non-negativity constraints, a little reflection shows that this problems is just as complex as (1). Let us now *soften* the inequalities by introducing a barrier term in the objective function to obtain the new problem

$$\begin{array}{ll} \text{minimize} & \phi(x;\mu) = f(x) - \mu \sum_{i=1}^{t} \ln s_i \\ \text{subject to} & h(x) = 0 \\ & g(x) + s = 0, \end{array} \tag{10}$$

where μ is a positive parameter. Note that we have removed the bound $s \geq 0$ because we will assume that the initial value of s is positive, and the barrier term prevents us from generating negative values of s – or for that matter, values that are close to zero.

Of course, (10) is not equivalent to (1) and we have introduced a parameterization of the problem that is controlled by the barrier parameter μ. Note that (10) contains only equality constraints, and is much simpler to solve than an inequality constrained problem. Once the barrier problem (10) is approximately solved, we decrease μ, and repeat the process. This will lead to a sequence of iterates x_μ that will normally converge to a solution of (1) as $\mu \to 0$.

The set of estimates x_μ obtained by this approach is interior to the region $s > 0$, but is not necessarily feasible with respect to the inequalities $g(x) \leq 0$. Thus the term "interior point method" must be interpreted in a broad sense. The ability to generate infeasible iterates turns out to be highly advantageous in practice because finding a feasible point for a nonlinear system is computationally expensive, and it is more efficient to perform the minimization while searching for a feasible point.

Barrier methods for nonlinear programming have been known for a long time [3]. But they fell out of favor in the 1970s, and have been resurrected only recently, in a variation that we now call interior point methods. There are three recent developments that have made barrier methods more effective in solving large problems. We will discuss each of these separately.

3.1 Primal-Dual Steps

Let us consider the problem of finding an approximate solution of the barrier problem (10) for a fixed value of the parameter μ. The Karush-Kuhn-Tucker conditions take the form

$$\begin{aligned} \nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g &= 0 \\ -\mu S^{-1}e + \lambda_g &= 0 \\ h(x) &= 0 \\ g(x) + s &= 0, \end{aligned} \tag{11}$$

where $e = (1, ..., 1)^T$ and $S = \mathrm{diag}(s_1, ..., s_t)$. This is a nonlinear system of equations in x, λ_h and λ_g. We can ignore (for the moment) the fact that s and λ_g must be positive, and simply apply Newton's method to (11) to compute a displacement p in x and new values of the multipliers. We obtain the iteration

$$\begin{bmatrix} \nabla^2_{xx}L & 0 & A_h(x) & A_g(x) \\ 0 & \Sigma & 0 & I \\ A_h^T(x) & 0 & 0 & 0 \\ A_g^T(x) & I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ \lambda_h^+ \\ \lambda_g^+ \end{bmatrix} = \begin{bmatrix} -\nabla f(x) \\ \mu S^{-1}e \\ -h(x) \\ -g(x) - s \end{bmatrix}, \tag{12}$$

where $\Sigma = \mu S^{-2}$. This approach is very similar to the barrier techniques used in the 1980s (cf. [10]) and is called a *primal* barrier method.

An important observation is that (11) is not well suited for Newton's method because the second equation is rational. But if we multiply this equation by S we obtain the equivalent system

$$\begin{aligned} \nabla f(x) + A_h(x)\lambda_h + A_g(x)\lambda_g &= 0 \\ S\lambda_g - \mu e &= 0 \\ h(x) &= 0 \\ g(x) + s &= 0. \end{aligned} \tag{13}$$

This nonlinear transformation is very beneficial because the rational equation has now been transformed into a quadratic – and Newton's method is an excellent technique for solving quadratic equations.

Applying Newton's method to (13) gives the iteration (12) but now Σ is defined as

$$\Sigma = \Lambda S^{-1}, \tag{14}$$

where Λ is a diagonal matrix containing the entries of λ_g. This *primal-dual* iteration is at the heart of most interior point methods. After the step is computed, one can backtrack along it to make sure that s and the λ_g remain positive.

Note that, in contrast to standard practice, we have not used any duality arguments in deriving the primal-dual step computation. Indeed the term "primal-dual" is not very descriptive of the key idea, which consists of applying a nonlinear transformation that changes the optimality conditions (11) into the equivalent system (13). Even though these two systems have the same solutions, Newton's method will produce different iterates, and the primal-dual step is known to be superior [13].

An interesting question is whether the nonlinear transformation we used is the best possible.

3.2 Coping With Ill-Conditioning

The barrier function $\phi(x;\mu)$ defined in (10) is inherently ill conditioned. A simple computation shows that the Hessian of ϕ has condition number of order $O(1/\mu)$. This is reflected in the primal-dual iteration (12) where the matrix $\Sigma = \Lambda S^{-1}$ becomes unbounded as $\mu \to 0$. Nevertheless, solving (12) by a direct method, as is done in most linear programming codes, does not lead to significant roundoff errors, even when μ is very small [11, 12].

The key observation in this roundoff error analysis can be better explained if we consider Newton-like methods for solving the unconstrained problem min $f(x)$. Here the step p is computed by solving a system of the form

$$Ap = -\nabla f(x),$$

where A is either the Hessian matrix $\nabla^2 f(x)$ or some other related matrix. It is easy to see that the quality of the search direction is very sensitive to the accuracy with which $\nabla f(x)$ is calculated, but is not particularly sensitive to changes in A. The ill-conditioning of the barrier function can cause errors in the factorization of the iteration matrix, but very significant errors can be tolerated before the quality of the iteration is degraded — and simple safeguards ensure that high accuracy is obtained in most cases [12].

All of this assumes that a direct method is used to solve (12). But in many practical applications, the problem is so large that direct methods are impractical due to the great amount of fill that occurs in the factorization. In other applications, the Hessians of f, g or h are not be available, and only products of these Hessians times vectors can be computed. In these cases it is attractive to use the linear conjugate gradient (CG) method to solve the Newton equations (12). This system is indefinite, but by eliminating variables, one obtains a positive definite reduced system to which the projected conjugate gradient method can be applied [2, 1].

When using the conjugate gradient method to solve the Newton equations, ill-conditioning is a grave concern. The unfavorable distribution of eigenvalues of the matrix in (12) may require a large number of CG iterations, and may even prevent us from achieving sufficient accuracy in the step computation. Fortunately, since the barrier function is separable and the portion that gives rise to the ill-conditioning is known explicitly, we can apply preconditioning techniques. To describe them let us recall that the step given by (12) has been decomposed in terms of its x and s-components, $p = (p_x, p_s)$. Then the change of variables

$$\tilde{p}_s = \mu S^{-2} p_s,$$

transforms the primal-dual matrix $\Sigma = \Lambda S^{-1}$ into $\Sigma = \mu^{-1}\Lambda S$. The second equation in (11) implies that ΛS converges to μI, showing that the new matrix Σ will not only be bounded, but will converge to the identity matrix. The CG iteration can now be effectively applied to the transformed system [1]. One should note, however, that this preconditioning comes at a price, and increases the cost of the CG iteration [1].

In summary, we have learned how to cope with ill-conditioning in barrier methods for nonlinear optimization. These observations also indicate that developing quasi-Newton variants of the interior methods just described may not pose significant difficulties provided that we approximate only the Hessian of the Lagrangian (7) of the original problem (1), as opposed to the Hessian of the Lagrangian of the barrier problem (10) which contains structural ill-conditioning.

3.3 Predictor-Corrector Strategy

The third key contribution of interior point methods has been the idea of using probing schemes to determine how fast to reduce the barrier parameter, and at the same time to determine (indirectly) how accurately to solve the barrier problem [7]. We cannot describe these *predictor-corrector* techniques here, and refer the reader to [13] for an excellent treatment of this subject.

We will only outline the key ideas of this approach which, at present, has only been implemented in the context convex optimization. Its most interesting feature is that it goes beyond the principle of Newton's method which computes a step based on an approximation of the problem at the current point. Instead, one first probes the problem by attempting to solve (12) with $\mu = 0$, which amounts to trying to solve the original nonlinear program (1). By gathering information in this probing iteration (the predictor), we can make a decision on how much to decrease the barrier parameter. At the same time, and at minimal cost, we can compute a primal-dual type step that corrects the predictor step and generates an iterate that is closer to the solution of the current barrier problem.

4 Final Remarks

Let us contrast the active set and interior approaches described in the previous sections by comparing the way in which they generate steps.

In the active set method we compute an *exact* solution of the subproblem (9). This is a full-fledged inequality constrained problem which can be costly to solve when the number of variables and constraints is large – particularly if the Hessian of the model is not positive definite. This is the main disadvantage of active set methods.

The great virtue of the active set approach is that it gives us, at every iteration, a guess of the optimal active set. As the iterates approach the solution, the active set of the subproblem (9) does not change, or undertakes minimal changes. This allows great savings in the solution of the subproblem because a warm start can be used: the solution of a new subproblem (9) can start from the active set identified at the previous iteration, and one can also re-use certain matrix factorizations [6].

Let us now consider interior point methods. The primal-dual iteration (12) is only a local method, and must be modified to be capable of dealing with non-convex problems. The interior methods described in [1] and [14] compute the step by solving a quadratic subproblem obtained by making a linear-quadratic approximation of the barrier problem. This approximation is such that, asymptotically, the iteration reduces to the primal-dual iteration (12). In both of these approaches there is an explicit bound on the step p_s in the slack variables. It takes the form

$$p_s \geq 0.995s,$$

and is known as a "fraction to the boundary rule".

This subproblem appears to be very similar to (9) since it also contains inequality constraints, but the presence of the barrier terms in the objective softens these constraints. Whereas in the active set approach the solution of the subproblem will normally lie on the boundary of the feasible region, in the interior approach this will not be the case, and solving the subproblem is simpler. This is one of the great advantages of interior methods.

A drawback of interior methods is that they normally do not provide a clear indication of the optimal active set until the solution is computed to high accuracy. This is undesirable in some applications, and future interior point codes may need to switch to an active set iteration, if necessary. Another weakness of interior methods is that they cannot efficiently re-use information from a previous subproblem. Roughly speaking, the solution of every subproblem requires the same amount of work. Finally, it is not yet known if interior point methods will prove to be as robust as active set methods for solving difficult non-convex problems.

These observations are based on the limited numerical experience that has been accumulated for both approaches when solving large problems. Once we have gained a better understanding of their practical behavior, and after new variants have been proposed, we will undoubtly discover that other unforeseen issues will tilt the balance towards one approach or the other.

REFERENCES

[1] R.H. Byrd, M.E. Hribar, and J. Nocedal. *An Interior Point Algorithm for Large Scale Nonlinear Programming*, Technical Report OTC 97/05, Optimization Technology Center, Northwestern University (1997).

[2] T.F. Coleman and A. Verma. *A Preconditioned Conjugate Gradient Approach to Linear Equality Constrained Minimization*, Technical Report, Computer Science Dept., Cornell University, Ithaca, New York.

[3] A.V. Fiacco and G.P. McCormick. *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, Wiley & Sons, 1968.

[4] R. Fletcher. *Practical Methods of Optimization, Second Edition*, John Wiley & Sons, New York, 1990.

[5] R. Fletcher and S. Leyffer. *Nonlinear Programming Without a Penalty Function*, Technical Report NA/171, Department of Mathematics, University of Dundee, Dundee, Scotland.

[6] P.E. Gill, W. Murray, and M.A. Saunders. *An SQP algorithm for large-scale optimization*, Technical Report, Mathematics Department, University of California at San Diego, San Diego, California.

[7] S. Mehrotra. *On the Implementation of a Primal-Dual Interior Point Method*, SIAM Journal on Optimization, 2, pp. 575-601, 1992.

[8] M.J.D. Powell. *The Convergence of Variable Metric Methods for Nonlinearly Constrained Optimization Calculations*, in “Nonlinear Programming 3”, (O. Mangasarian, R. Meyer and S. Robinson, eds), pp. 27-64, Academic Press, New York.

[9] S.M. Robinson. *Perturbed Kuhn-Tucker Points and Rates of Convergence for a Class of Nonlinear Programming Algorithms*, Math. Programming, 7, pp. 1-16, 1974.

[10] M.H. Wright. *Interior Point Methods for Constrained Optimization*, Acta Numerica 1992 (A. Iserles ed.), pp. 341-407, Cambridge University Press.

[11] M.H. Wright. *Ill-Conditioning and Computational Error in Interior Methods for Nonlinear Programming*, Technical Report 97-4-04, Computing Sciences Research Center, Bell Laboratories, Murray Hill, New Jersey.

[12] S.J. Wright. *Finite-Precision Effects on the Local Convergence of Interior-Point Algorithms for Nonlinear Programming*, Preprint ANL/MCS P705-0198, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, Illinois.

[13] S.J. Wright. *Primal-Dual Interior Point Methods*, SIAM, 1997.

[14] H. Yamashita. *A Globally Convergent Primal-Dual Interior Point Method for Constrained Optimization*, Technical Report, Mathematical Systems Institute Inc, Tokyo, Japan.

Richard H. Byrd
Computer Science Dept.
University of Colorado,
Boulder CO 80309
USA
richard@cs.colorado.edu

Jorge Nocedal
ECE Department
Northwestern University
Evanston IL 60208
USA
nocedal@ece.nwu.edu