The problem

An inequality-constrained nonlinear programming problem may be posed in the form

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & c(x) \geq 0, \end{array} \tag{1}$$

where $f(x)$ is a nonlinear function and $c(x)$ is an m-vector of nonlinear functions with ith component $c_i(x)$, $i = 1, \ldots, m$. We shall assume that f and c are *sufficiently* smooth. Let x^* denote a solution to (1). We are mainly concerned about smoothness in the neighborhood of x^*. In such a neighborhood we assume that both the gradient of $f(x)$ denoted by $g(x)$ and the $m \times n$ Jacobian of $c(x)$ denoted by $J(x)$ exist and are Lipschitz continuous. As is the case with the unconstrained problem a solution to this problem may not exist. Typically additional assumptions are made to ensure a solution does exist. A common assumption is to assume that the objective $f(x)$ is bounded below on the feasible set. However, even this is not sufficient to assure a minimizer exists but it is obviously a necessary condition for an algorithm to be assured of converging. If the feasible region is compact then a solution does exist. We shall only be concerned with local solutions.

First-order optimality conditions

The problem is closely related to the equality-constrained problem. If it was known which constraints were active (exactly satisfied) at a solution and which were slack (strictly positive) then the optimality conditions for (1) could be replaced by the optimality conditions for the equality case. Note that this does not imply the inequality problem could be replaced by an equality problem when it comes to determining a solution by an algorithm. The inequality problem may have solutions corresponding to different sets of constraints being active. Also an equality problem may have solutions that are not solutions of the inequality problem. Nonetheless this equivalence in a local neighborhood enables us to determine the optimality conditions for this problem from those of an equality-constrained problem. In order to study the optimality conditions it is necessary to introduce some notation.

Let $\widehat{c}(x)$ and $\bar{c}(x)$ denote the constraints active and slack at x respectively. Likewise, let $\widehat{J}(x)$ and $\bar{J}$ denote their respective Jacobians. Assume that $\widehat{J}(x^*)$ is full rank. Points at which the Jacobian of the active constraints is full rank are said to be *regular*. It follows from the necessary conditions

for the equality case that

$$\begin{aligned} g(x^*) - \widehat{J}(x^*)^T\hat{\lambda} &= 0, \\ \widehat{c}(x^*) &= 0, \\ \bar{c}(x^*) &> 0, \end{aligned}$$

where $\hat{\lambda}$ is vector of Lagrange multipliers. These equations may be written in the form:

$$\begin{aligned} g(x^*) - J(x^*)^T\lambda^* &= 0, \\ c(x^*) &\geq 0, \\ \lambda^{*T}c(x^*) &= 0, \end{aligned}$$

where λ^* is the *extended* set of Lagrange multipliers. The set is extended by defining a multiplier to be zero for the slack constraints at x^* ($\bar{c}(x^*)$).

The above first-order optimality conditions are not the only necessary conditions. Unlike the equality case there may be a feasible arc that moves off one or more of the active constraints along which the objective is reduced. In other words we need some characterization that is necessary for the active set to be binding. The key to identifying the binding set is to examine the sign of $\hat{\lambda}$.

It follows from the definition of $\hat{\lambda}$ that

$$\hat{\lambda} = (\widehat{J}\widehat{J}^T)^{-1}\widehat{J}g, \tag{2}$$

where the argument x^* has been dropped for simplicity. Note that (2) implies that $\|\hat{\lambda}\|$ is bounded.

Define p as

$$\widehat{J}p = \delta e + e_j,$$

where $\delta > 0$, e denotes the vector of ones and e_j is the unit column with one in the jth position. It follows from the assumption on the continuity of the Jacobian that $x^* + \alpha p$ is feasible for $0 \leq \alpha \leq \bar{\alpha}$, if $\bar{\alpha}$ is sufficiently small. From the mean-value theorem we have

$$f(x^* + \alpha p) = f(x^*) + \alpha p^T g(x^* + \xi\alpha p),$$

where $0 \leq \xi \leq 1$. The Lipschitz continuity of g implies M exists such that

$$p^T g(x^* + \xi\alpha p) \leq p^T g(x^*) + \alpha M.$$

It follows that

$$f(x^* + \alpha p) \leq f(x^*) + \alpha(p^T g(x^*) + \alpha M).$$

From the necessary conditions on x^* we get

$$p^T g(x^*) = p^T \widehat{J}^T \hat{\lambda},$$

which implies

$$f(x^* + \alpha p) \leq f(x^*) + \alpha(p^T \widehat{J}^T \hat{\lambda} + \alpha M).$$

Using the definition of p gives

$$f(x^* + \alpha p) \leq f(x^*) + \alpha(\delta e^T \hat{\lambda} + \hat{\lambda}_j + \alpha M).$$

It follows from the boundedness of $\hat{\lambda}$ that if $\hat{\lambda}_j < 0$ then for δ sufficiently small there exists $\bar{\alpha}$ such that for $0 < \alpha \leq \bar{\alpha}$,

$$f(x^* + \alpha p) < f(x^*).$$

Consequently, a *necessary* condition for x^* to be a minimizer under the assumptions made is that $\hat{\lambda} \geq 0$. Equivalently, $\lambda^* \geq 0$.

For different assumptions such as $\widehat{J}$ not being full rank the condition need not hold as the following simple case illustrates. Suppose we have an equality-constrained problem with $c(x) = 0$ then an equivalent inequality-constrained problem is

$$\begin{array}{lrcl} \underset{x \in I\!R^n}{\text{minimize}} & f(x) & & \\ \text{subject to} & c(x) & \geq & 0, \\ & -c(x) & \geq & 0. \end{array}$$

It follows that all constraints are active at a solution. We know in this case there are no necessary conditions on the Lagrange multipliers. Clearly the Jacobian of the active constraints is not full rank. Geometrically what breaks down is that there is no perturbation from x^* that moves feasible with respect to one constraint without violating at least one other constraint.

The condition $c(x^*)^T \lambda^* = 0$ is a *complementarity* condition. At least one of $(c_i(x^*),\ \lambda_i^*)$ must be zero. It is possible for both to be zero. If there is no index for which both are zero then $c(x^*)$ and λ^* are is said to satisfy *strict* complementarity.

If $\widehat{J}(x^*)$ is full rank then it follows from (2) that λ^* is an isolated point.

The function $L(x, \lambda)$

$$L(x, \lambda) = F(x) - \lambda^T c(x),$$

is known as the Lagrangian. The optimality condition

$$g(x^*) - J(x^*)^T \lambda^* = 0$$

is equivalent to $\nabla_x L(x^*, \lambda^*) = 0$. It is also equivalent to $Z(x^*)^T g(x^*) = 0$, where the columns of $Z(x)$ are a basis for the null space of the rows of $\widehat{J}(x)$. The vector $Z(x)^T g(x)$ is called the *reduced* gradient.

Clearly Lagrange multipliers play a significant role in defining the solution of an inequality-constrained problem. There is a significant difference in that role between linear and nonlinear constraints. In the case of linear constraints the numerical value of the multiplier plays no role in defining x^* only the sign of the multiplier is significant. For nonlinear constraints the numerical value as well as the sign is of significance. To appreciate why it first necessary to appreciate that for problems that are nonlinear in either the constraints or the objective, curvature of the functions are relevant in defining x^*. More precisely the curvature of the Lagrangian. It easily seen that curvature of the objective is relevant since for unconstrained problems no solution would exist otherwise. To appreciate that curvature in $c(x)$ is relevant note that *any* problem can be transformed into a problem with just a linear objective by adding an extra variable. For example, add the constraint $x_{n+1} - f(x) \geq 0$ and minimize x_{n+1} instead of $f(x)$. Since we have established the curvature of $f(x)$ is relevant that relevance must still be there even though $f(x)$ now appears only within a constraint. It is harder to appreciate that it is the relative curvature of the various constraints and objective that is of significance.

1 Second-order optimality conditions

We shall now assume that the problem functions are twice continuous differentiable. From the unconstrained case it is known that a necessary condition is that $\nabla^2 f(x^*)$ is positive semidefinite. Obviously a generalization of this condition needs to hold for (1). Again the Lagrangian will be shown to play a key role. We start by examining the behavior of $f(x)$ along a feasible arc emanating from x^*. Although the first-order optimality conditions make the first-order change in the objective along a feasible arc non-negative, it could be zero. Consequently, the second-order change needs to be non-negative for arcs where this is true.

We restrict our interest to feasible arcs that remain on the set of constraints active at x^*. If $x(\alpha)$ represents a twice differentiable arc, with $x(0) = x^*$, that lies on the active set then $\widehat{c}(x(\alpha)) = 0$. Define $p \equiv d(x(0))/d\alpha$ and $h \equiv d^2(x(0))/d\alpha^2$. We have

$$\begin{aligned}
\frac{d}{d\alpha}\widehat{c}_i\big(x(\alpha)\big) &= \nabla(\widehat{c}_i(x(\alpha))^T\frac{d}{d\alpha}x(\alpha). \\
\frac{d^2}{d\alpha^2}\widehat{c}_i\big(x(\alpha)\big) &= \frac{d}{d\alpha}x(\alpha)^T\nabla^2\widehat{c}_i(x(\alpha))\frac{d}{d\alpha}x(\alpha) \\
&\quad +\nabla\widehat{c}_i(x(\alpha))^T\frac{d^2}{d\alpha^2}x(\alpha).
\end{aligned}$$

Since $\widehat{c}(x(\alpha)) = 0$ it follows that

$$\frac{d^2}{d\alpha^2}\widehat{c}_i\big(x(0)\big) = \nabla\widehat{c}_i(x^*)^Th + p^T\nabla^2\widehat{c}_i(x^*)p = 0. \tag{3}$$

Similarly we get

$$\frac{d^2}{d\alpha^2}f(x(0)) = g(x^*)^Th + p^T\nabla^2 f(x^*)p.$$

Since

$$\frac{d}{d\alpha}f(x(0)) = g(x^*)^Tp = 0$$

(otherwise there would be a descent direction from x^*) we require that

$$g(x^*)^Th + p^T\nabla^2 f(x^*)p \geq 0.$$

Substituting for $g(x^*)$ using the first-order optimality conditions gives

$$h^TJ(x^*)^T\lambda^* + p^T\nabla^2 f(x^*)p \geq 0.$$

It follows from (3) and the definition of the extended multipliers that we require

$$-\sum_{i=1}^{m}\lambda_i^*p^T\nabla^2 c_i(x^*)p + p^T\nabla^2 f(x^*)p \geq 0.$$

From the definition of $L(x^*,\lambda^*)$ and $\widehat{J}(x^*)p = 0$ this condition is equivalent to requiring that $Z(x^*)^T\nabla^2 L(x^*,\lambda^*)Z(x^*)$ be positive semi-definite. This matrix is called the *reduced Hessian of the Lagrangian*. Since the condition is on the second derivatives it is termed a second-order optimality condition. It can now be appreciated that the numerical value of the Lagrange multipliers play a role in defining the solution of a nonlinearly-constrained problem. Note that when there are n active constraints then there is no feasible arc that remains on the active set and the second-order optimality condition is

empty. When $\widehat{J}$ has n rows then the reduced Hessian has zero dimension. For convenience we can define symmetric matrices of zero dimension to be positive definite.

Necessary *and* sufficient conditions for x^* to be a minimizer are complex. However, sufficient conditions are easy to appreciate. We have established no feasible descent direction exists that moves off any of the active constraints. Consequently, if $\hat{\lambda} > 0$ then $f(x)$ increases along any feasible arc emanating from x^* that moves off a constraint. We now only need to be sure the same is true for all arcs emanating from x^* that remaining on the active set. This is assured if

$$\frac{d^2}{d\alpha^2} f(x(0)) = g(x^*)^T h + p^T \nabla^2 f(x^*) p > 0,$$

which implies $Z(x^*)^T \nabla^2 L(x^*, \lambda^*) Z(x^*)$ is positive definite. Assuming that x^* is a regular point, strict complementarity hold, the first-order necessary conditions hold, and the reduced Hessian at x^* is positive definite then x^* is a minimizer and an isolated point.

Algorithms

Algorithms for inequality problems have a combinatorial element not present in algorithms for equality-constrained problems. The simplest case of linear programming (LP) illustrates the point. Under mild assumptions the solution of an LP is given by the solution of a set of linear equations, ie. a vertex of the feasible region. The difficult issue is determining which of the constraints define those equations. If there are m inequality constraints and n variables there are $m!/n!(n-m)!$ choices of active constraints. Even for modest values of m and n the possible choices are astronomical. This clearly rules out methods based on exhaustive search.

One class of methods to solve inequality problems are so-called *active-set methods* an example being the simplex method for LP. First a guess is made of the active set (called the working set) and then an estimate to the solution of the resulting equality-constrained problem is computed (in the case of LP or quadratic programming (QP) this would be precise) and at the new point a new guess is made of the active set. The estimate of the solution of the equality-constrained problem is usually made by finding a point that satisfies an approximation to the first-order necessary conditions. Unless an intelligent guess is made of the active set such algorithms are doomed to fail. Typically after the initial active set such algorithms generate subsequent working sets automatically. For linearly-constrained problems

this is usually a very simple procedure. Assuming the current iterate is feasible an attempt is made to move to the new estimate of the solution. If this is infeasible the best (or a point better than the current iterate) is found along the direction to the new estimate. The constraints active at the new feasible point are then used to define the working set. Usually the active set will be the working set but occasionally we need to move off a constraint that is currently active. How to identify such a constraint is usually straightforward and can be done by examining an estimate to the Lagrange multipliers (obtained from the solution to the approximation of the first-order necessary conditions). More complex strategies are possible that move off several constraints simultaneously. An initial feasible point is found by solving an LP. One consequence of this strategy is that it is only necessary to consider working sets for which the objective function has a lower value than at the current iterate. Once we are in a neighborhood of the solution the working set does not change if strict complementarity holds at the solution and x^* is a regular point. Typically the change in the working set at each iteration of active-set methods for linearly-constrained problems is small (usually one), which results in efficiencies when computing the estimate to the new equality-constrained problem. In practice active-set methods work well and usually identify the active set at the solution with very little difficulty. For an LP the number of iterations required to identify the active set usually grows linearly with the size of the problem. However, pathological cases exist in which the number of iterations is astronomical and real LP problems do arise where the number of iterates required is much greater than the typical case. Nonetheless algorithms for linearly-constrained problems based on active-set methods are highly successful.

For nonlinear problem the issue of identifying the active set at the solution is usually less significant since even when the active set is known the number of iterations required to solve a problem may be large. A more relevant issue is that not knowing the active set causes some problems such as making the linear algebra routines much more complicated. For small problems this is of little consequence but in the large-scale case it complicates the data structures required.

Nonlinearly-constrained problems are usually an order of magnitude more complicated to solve than linearly-constrained problems. One reason is that algorithms for problems with nonlinear constraints usually do not maintain feasible iterates. If a problem has just one nonlinear equality constraint then generating each member of a sequence that lies on that constraint is itself an infinite process. Methods that generate infeasible interates need to have some means of assessing whether a point is better than

another point. For feasible-point algorithms this is a simple issue since the objective provides a measure of merit. A typical approach is to define a merit function, which balances a change in the objective against the change in the degree of infeasibility. A commonly used merit function is

$$M(x, \rho) = f(x) + \rho \sum_{i=1}^{m} \max\{0, \ -c_i(x)\},$$

where ρ is a parameter that needs to be sufficiently large. Usually it will not be known what "sufficiently" large is so this parameter is adjusted as the sequence of iterates is generated. Note that $M(x, \rho)$ is not a smooth function and has a discontinuity in its derivative when any element of $c(x)$ is zero. In particular it is not continuous at x^* when a constraint is active at x^*. Were this not the case then constrained problems could be transformed to unconstrained problems and solved as such. While transforming a constrained problem into a simple single smooth unconstrained problem is not possible the transformation approach is the basis of a variety of methods. A popular alternative to direct methods is to transform the problem into that of solving a *sequence* of smooth linearly-constrained problems. This is the method at the heart of MINOS (see [8, 9]) one of the most widely used methods for solving problems with nonlinear constraints. Other transformation methods transform the problem to that of solving a sequence of unconstrained or bounds-constrained problem. Transformation methods have an advantage of over direct methods when developing software. For example, if you have a method for solving large-scale linearly-constrained problems then it can be used as a kernal in an algorithm to solve large-scale nonlinearly-constrained problems.

References

[1] Bazaraa, M. S., Sherali, H. D., and Shetty, C. M.: *Nonlinear Programming: Theory and Algorithms*, second ed., John Wiley and Sons, New York, 1993, ISBN 0-471-55793-5.

[2] Bertsekas, D. P.: *Nonlinear Programming*, Athena Scientific, Belmont, 1999 (second edition), ISBN 1-886529-14-0.

[3] Fletcher, R.: *Practical Methods of Optimization*, John Wiley and Sons, 1988, ISBN 0-471-91547-5

[4] Karush, W.: Minima of functions of several variables with inequalities as side constraints, Master's thesis, Department of Mathematics, University of Chicago, 1939.

[5] Gill, P. E., Murray, W. and Wright, M. H.: *Practical Optimization*, Academic Press, 1981, ISBN 0-12-283952-8

[6] Kuhn, H. W.: 'Nonlinear programming: a historical note', in J. K. Lenstra, A. H. G. Rinnooy Kan, and A. Schrijver (eds.): *History of Mathematical Programming: A Collection of Personal Reminiscences*, Elsevier Science Publishers B. V., 1991, pp. 82–96.

[7] Kuhn, H. W., and Tucker, A. W.: 'Nonlinear programming', in J. Neyman (ed.): *Proceedings of the Second Berkeley Symposium on Mathematical Statistics and Probability*, University of California Press, 1951, pp. 481–492.

[8] B. A. Murtagh and M. A. Saunders: A projected Lagrangian algorithm and its implementation for sparse nonlinear constraints. *Math. Prog. Study*, 16:84–117, 1982.

[9] B. A. Murtagh and M. A. Saunders: MINOS 5.4 User's Guide. Report SOL 83-20R, Department of Operations Research, Stanford University, Stanford, CA, 1993.

[10] Nash, S. G., and Sofer, A.: *Linear and Nonlinear Programming*, McGraw-Hill, New York, 1996, ISBN 0-07-046065-5.