Algorithms for Constrained Optimization

Methods for solving a constrained optimization problem in *n* variables and *m* constraints can be divided roughly into four categories that depend on the dimension of the space in which the accompanying algorithm works. Primal methods work in *n* – *m* space, penalty methods work in *n* space, dual and cutting plane methods work in *m* space, and Lagrangian methods work in *n* + *m* space. Each of these approaches are founded on different aspects of NLP theory. Nevertheless, there are strong interconnections between them, both in the final form of implementation and in performance. The rates of convergence of most practical algorithms are determined by the structure of the Hessian of the Lagrangian, much like the structure of the Hessian of the objective function determines the rates of convergence for most unconstrained methods.

In this appendix, we present several procedures for solving problem (1). The first is the now classical penalty approach developed by Fiacco and McCormick [1968] and is perhaps the simplest to implement. The second is Zoutendijk's feasible direction method. Other primal approaches discussed in the literature include the gradient projection method and the generalized reduced gradient (GRG) method that is a simplex-like algorithm. There are many commercial codes that implement these and related techniques. Sequential linear programming and sequential quadratic programming (SQP), for example, are two Lagrangian approaches that have proven to be quite effective. SQP is highlighted at the end of this appendix.

A.1 Penalty and Barrier Methods

The methods that we describe presently, attempt to approximate a constrained optimization problem with an unconstrained one and then apply standard search techniques to obtain solutions. The approximation is accomplished in the case of penalty methods by adding a term to the objective function that prescribes a high cost for violation of the constraints. In the case of barrier methods, a term is added that favors points in the interior of the feasible region over those near the boundary. For a problem with *n* variables and *m* constraints, both approaches work directly in the *n*-dimensional space of the variables. The discussion that follows emphasizes penalty methods recognizing that barrier methods embody the same principles.

Consider the problem

$$\text{Minimize}\{f(\mathbf{x}) : \mathbf{x} \quad \boldsymbol{S}\} \tag{23}$$

where *f* is continuous function on ^{n} and ***S*** is a constraint set in ^{n}. In most applications *S* is defined explicitly by a number of functional constraints, but in this section the more general description in (23) can be handled. The idea of a penalty function method is to replace problem (23) by an unconstrained approximation of the form

$$\text{Minimize}\{f(\mathbf{x}) + cP(\mathbf{x})\} \tag{24}$$

where c is a positive constant and P is a function on $\quad^n$ satisfying (i) $P(\mathbf{x})$ is continuous, (ii) $P(\mathbf{x}) \geq 0$ for all $\mathbf{x} \quad^n$, and (iii) $P(\mathbf{x}) = 0$ if and only if $\mathbf{x}$ S.

Example 16

Suppose S is defined by a number of inequality constraints: $S = \{\mathbf{x} : \mathrm{g_i}(\mathbf{x}) \leq 0,\ i = 1,\ldots, m\}$. A very useful penalty function in this case is

$$P(\mathbf{x}) = \tfrac{1}{2}\sum_{i=1}^{m}(\max\{0, g_i(\mathbf{x})\}^2 \tag{25}$$

which gives a quadratic augmented objective function denoted by

$$\theta(c,\mathbf{x}) \equiv f(\mathbf{x}) + cP(\mathbf{x}).$$

Here, each unsatisfied constraint influences **x** by assessing a penalty equal to the square of the violation. These influences are summed and multiplied by c, the *penalty parameter*. Of course, this influence is counterbalanced by $f(\mathbf{x})$. Therefore, if the magnitude of the penalty term is small relative to the magnitude of $f(\mathbf{x})$, minimization of $\theta(c,\mathbf{x})$ will almost certainly not result in an **x** that would be feasible to the original problem. However, if the value of c is made suitably large, the penalty term will exact such a heavy cost for any constraint violation that the minimization of the augmented objective function will yield a feasible solution.

The function $cP(\mathbf{x})$ is illustrated in Fig. 19 for the one-dimensional case with $g_1(\mathrm{x}) = b - x$ and $g_2(\mathrm{x}) = x - a$. For large c it is clear that the minimum point of problem (24) will be in a region where P is small. Thus for increasing c it is expected that the corresponding solution points will approach the feasible region S and, subject to being close, will minimize f. Ideally then, as $c \rightarrow \infty$ the solution point of the penalty problem will converge to a solution of the constrained problem.

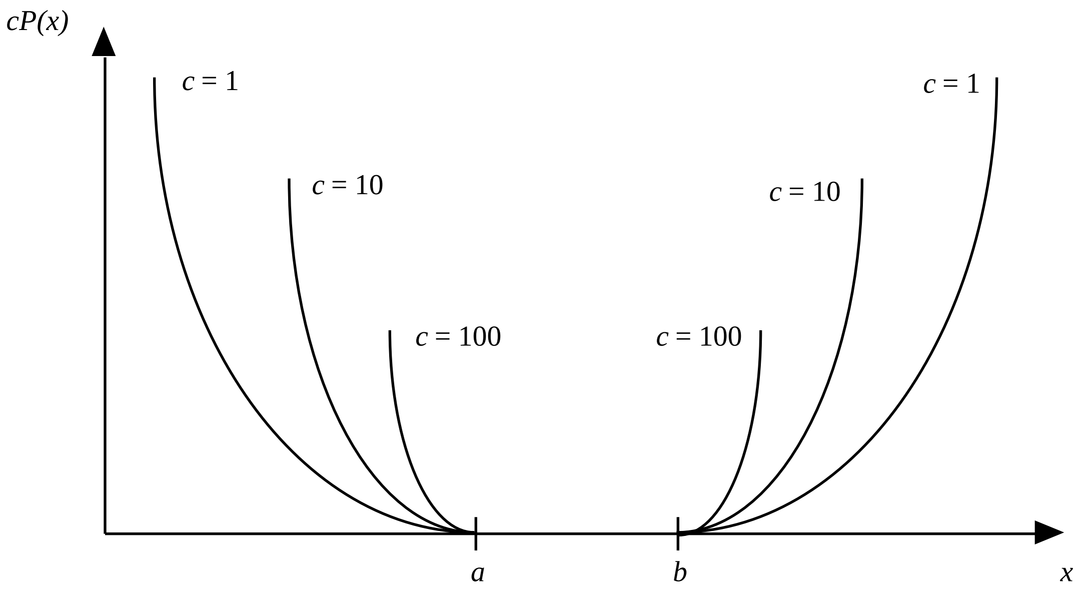

Figure 19. Illustration of penalty function

Example 17

To clarify these ideas and to get some understanding on how to select the penalty parameter *c*, let us consider the following problem.

$$\begin{aligned} \text{Minimize} \quad & f(\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 7)^2 \\ \text{subject to} \quad & g_1(\mathbf{x}) = -3x_1 - 2x_2 + 6 \le 0 \\ & g_2(\mathbf{x}) = -x_1 + x_2 - 3 \le 0 \\ & g_3(\mathbf{x}) = x_1 + x_2 - 7 \le 0 \\ & g_4(\mathbf{x}) = \tfrac{2}{3}x_1 - x_2 - \tfrac{4}{3} \le 0 \end{aligned}$$

The feasible region is shown graphically in Fig. 20 along with several isovalue contours for the objective function. The problem is a quadratic program and the isovalue contours are concentric circles centered at (6,7), the unconstrained minimum of $f(\mathbf{x})$.

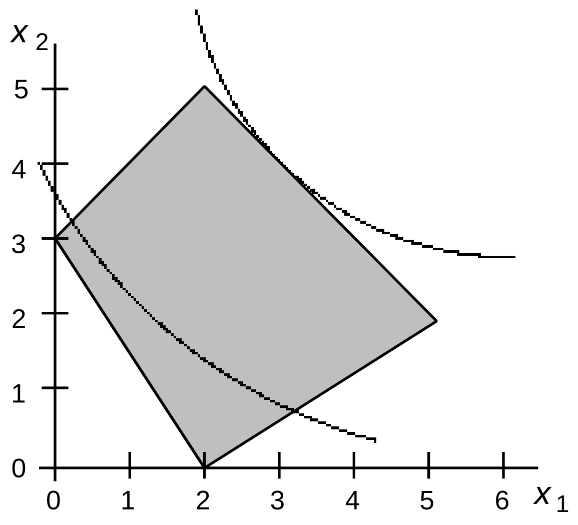

Figure 20. Feasible region for Example 17

Using the quadratic penalty function (25), the augmented objective function is

$$\theta(c,\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 7)^2 + c\big((\max\{0, -3x_1 - 2x_2 + 6\})^2$$
$$+ (\max\{0, -x_1 + x_2 - 3\})^2 + (\max\{0, x_1 + x_2 - 7\})^2$$
$$+ (\max\{0, \tfrac{2}{3}x_1 - x_2 - \tfrac{4}{3}\})^2\big).$$

The first step in the solution process is to select a starting point. A good rule of thumb is to start at an infeasible point. By design then, we will see that every trial point, except the *last* one, will be infeasible (exterior to the feasible region).

A reasonable place to start is at the unconstrained minimum so we set $\mathbf{x}^0 = (6,7)$. Since only constraint 3 is violated at this point, we have

$$\theta(c,\mathbf{x}) = (x_1 - 6)^2 + (x_2 - 7)^2 + c(\max\{0, x_1 + x_2 - 7\})^2.$$

Assuming that in the neighborhood of $\mathbf{x}^0$ the "max" operator returns the constraint, the gradient with respect to $\mathbf{x}$ is

$${}_{\mathbf{x}}\theta(c,\mathbf{x}) = \begin{pmatrix} 2x_1 - 12 + 2c(x_1 + x_2 - 7) \\ 2x_2 - 14 + 2c(x_1 + x_2 - 7) \end{pmatrix}.$$

Setting the elements of ${}_{\mathbf{x}}\theta(c,\mathbf{x})$ to zero and solving yields the stationary point

$$x_1^*(c) = \frac{6(1+c)}{1+2c} \text{ and } x_2^*(c) = 7 - \frac{6c}{1+2c}$$

as a function of c. For any positive value of c, $\theta(c,\mathbf{x})$ is a strictly convex function (the Hessian of $\theta(c,\mathbf{x})$ is positive definite for all $c > 0$), so $x_1^*(c)$ and $x_2^*(c)$ determine a global minimum. It turns out for this example that the minima will continue to satisfy all but the third constraint for all positive values of c. If we take the limit of $x_1^*(c)$ and $x_2^*(c)$ as $c \rightarrow \infty$, we obtain $x_1^* = 3$ and $x_2^* = 4$, the constrained global minimum for the original problem.

Selecting the Penalty Parameter

Because the above approach seems to work so well, it is natural to conjecture that all we have to do is set c to a very large number and then optimize the resulting augmented objective function $\theta(c,\mathbf{x})$ to obtain the solution to the original problem. Unfortunately, this conjecture is not correct. First, "large" depends on the particular model. It is almost always impossible to tell how large c must be to provide a solution to the problem without creating numerical difficulties in the computations. Second, in a very real sense, the problem is dynamically changing with the relative position of the current value of $\mathbf{x}$ and the subset of the constraints that are violated.

The third reason why the conjecture is not correct is associated with the fact that large values of c create enormously steep valleys at the constraint boundaries. Steep valleys will often present formidable if not insurmountable convergence difficulties for all preferred search methods unless the algorithm starts at a point extremely close to the minimum being sought.

Fortunately, there is a direct and sound strategy that will overcome each of the difficulties mentioned above. All that needs to be done is to start with a relatively small value of c and an infeasible (exterior) point. This will assure that no steep valleys are present in the initial optimization of $\theta(c,\mathbf{x})$. Subsequently, we will solve a sequence of unconstrained problems with monotonically increasing values of c chosen so that the solution to each new problem is "close" to the previous one. This will preclude any major difficulties in finding the minimum of $\theta(c,\mathbf{x})$ from one iteration to the next.

Algorithm

To implement this strategy, let $\{c_k\}$, $k = 1,2,...$ be a sequence tending to infinity such that $c_k > 0$ and $c_{k+1} > c_k$. Now for each k we solve the problem

$$\text{Minimize}\{\theta(c_k,\mathbf{x}) : \mathbf{x} \qquad {}^{n}\} \tag{26}$$

to obtain $\mathbf{x}^k$, the optimum   It is assumed that problem (26) has a solution for all positive values of c_k. This will be true, for example, if $\theta(c,\mathbf{x})$ increases without bounds as $||\mathbf{x}|| \to \infty$.

A simple implementation known as the sequential unconstrained minimization technique (SUMT), is given below.

Initialization Step: Select a growth parameter $\eta > 1$, a stopping parameter $\varepsilon > 0$, and an initial value of the penalty parameter c_0. Choose a starting point $\mathbf{x}^0$ that violates at least one constraint and formulate the augmented objective function $\theta(c_0,\mathbf{x})$. Let $k = 1$.

Iterative Step: Starting from $\mathbf{x}^{k-1}$, use an unconstrained search technique to find the point that minimizes $\theta(c_{k-1},\mathbf{x})$. Call it $\mathbf{x}^k$ and determine which constraints are violated at this point.

Stopping Rule: If the distance between $\mathbf{x}^{k-1}$ and $\mathbf{x}^k$ is smaller than ε (i.e., $||\mathbf{x}^{k-1} - \mathbf{x}^k|| < \varepsilon$) or the difference between two successive objective functions values is smaller than ε (i.e., $|f(\mathbf{x}^{k-1}) - f(\mathbf{x}^k)| < \varepsilon$), stop with $\mathbf{x}^k$ an estimate of the optimal solution. Otherwise, put $c^k \leftarrow \eta c^{k-1}$, formulate the new $\theta(c_k,\mathbf{x})$ based on which constraints are violated at $\mathbf{x}^k$, put $k \leftarrow k+1$ and return to the iterative step.

Applying the algorithm to Example 17 with $\eta = 2$ and $c_0 = 0.5$ for eight iterations yields the sequence of solutions given in Table A1. The iterates $\mathbf{x}^k$ are seen to approach the true minimum point $\mathbf{x}^* = (3,4)$.

Table A1. Sequence of Solutions Using the Penalty Method

k	c	x_1	x_2	g_3
0	—	6.00	7.00	6.00
1	0.5	4.50	5.50	3.00
2	1	4.00	5.00	2.00
3	2	3.60	4.60	1.20
4	4	3.33	4.33	0.66
5	8	3.18	4.18	0.35
6	16	3.09	4.09	0.18
7	32	3.05	4.05	0.09
8	64	3.02	4.02	0.04

Implementation Issues

Much of the success of SUMT depends on the approach used to solve the intermediate problems, which in turn depends on their complexity. One thing that should be done prior to attempting to solve a nonlinear program using a penalty function method is to scale the constraints so that the penalty generated by each is about the same magnitude. This scaling operation is intended to ensure that no subset of the constraints has an undue influence on the search process. If some constraints are dominant, the algorithm will steer towards a solution that satisfies those constraints at the expense of searching for the minimum.

In a like manner, the initial value of the penalty parameter should be fixed so that the magnitude of the penalty term is not much smaller than the magnitude of objective function. If an imbalance exists, the influence of the objective function could direct the algorithm to head towards an unbounded minimum even in the presence of unsatisfied constraints. In either case, convergence may be exceedingly slow.

Convergence

Although SUMT has an intuitive appeal, it is still necessary to prove that it will converge to the optimum of the original problem (23). The following lemma is the first component of the proof, and gives a set of inequalities that follows directly from the definition of $\mathbf{x}^k$ and the inequality $c_{k+1} > c_k$.

Lemma 1:

$$\theta(c_k, \mathbf{x}^k) \leq \theta(c_{k+1}, \mathbf{x}^{k+1})$$

$$P(\mathbf{x}^k) \geq P(\mathbf{x}^{k+1})$$

$$f(\mathbf{x}^k) \leq f(\mathbf{x}^{k+1})$$

Proof:

$$\theta(c_{k+1}, \mathbf{x}^{k+1}) = f(\mathbf{x}^{k+1}) + c_{k+1}P(\mathbf{x}^{k+1}) \geq f(\mathbf{x}^{k+1}) + c_kP(\mathbf{x}^{k+1})$$

$$\geq f(\mathbf{x}^k) + c_kP(\mathbf{x}^k) = \theta(c_k, \mathbf{x}^k),$$

which proves the first inequality. From this we also have

$$f(\mathbf{x}^k) + c_kP(\mathbf{x}^k) \leq f(\mathbf{x}^{k+1}) + c_kP(\mathbf{x}^{k+1})$$

$$f(\mathbf{x}^{k+1}) + c_{k+1}\mathrm{P}(\mathbf{x}^{k+1}) \leq f(\mathbf{x}^k) + c_{k+1}P(\mathbf{x}^k)$$

Adding these two expressions and rearranging terms yields $(c_{k+1} - c_k)P(\mathbf{x}^{k+1}) \leq (c_{k+1} - c_k)\mathrm{P}(\mathbf{x}^k)$ which proves the second inequality. Finally, $f(\mathbf{x}^{k+1}) + c_kP(\mathbf{x}^{k+1}) \geq f(\mathbf{x}^k) + c_kP(\mathbf{x}^k)$ which, in conjunction with the second inequality, leads to the third. ▮

Lemma 2: Let $\mathbf{x}^*$ be a solution to problem (23). Then for each k

$$f(\mathbf{x}^*) \geq \theta(c_k, \mathbf{x}^k) \geq f(\mathbf{x}^k)$$

Proof: $f(\mathbf{x}^*) = f(\mathbf{x}^*) + c_kP(\mathbf{x}^*) \geq f(\mathbf{x}^k) + c_kP(\mathbf{x}^k) \geq f(\mathbf{x}^k)$. ▮

Global convergence of the penalty method, or more precisely verification that any limit point of the sequence is a solution, follows easily from the two lemmas above.

Theorem 10: Let $\{\mathbf{x}^k\}$ be a sequence generated by the penalty method, where $\mathbf{x}^k$ is the global minimum of $\theta(c_{k-1}, \mathbf{x})$ at the iterative step of the algorithm. Then any limit point of the sequence is a solution to (23).

Proof: see Luenberger [1984].

Barrier Function

It should be apparent that the quadratic penalty function in (25) is only one of an endless set of possibilities. While the "sum of the squared violations" is probably the best type of penalty function for an *exterior method*, it is not at all suitable if one desires to conduct a search through a sequence of feasible or *interior* points. Let us consider the following augmented objective function

$$B(r,\mathbf{x}) = f(\mathbf{x}) + r \sum_{i=1}^{m} \frac{-1}{g_i(\mathbf{x})} \tag{27}$$

where $r > 0$ is the barrier parameter. This function is valid only for interior points such that all constraints are strictly satisfied: $g_i(\mathbf{x}) < 0$ for all i.

Equation (27) indicates that the closer one gets to a constraint boundary, the larger $B(r,\mathbf{x})$ becomes. Indeed, points precisely on any boundary are not defined. Hence $B(r,\mathbf{x})$ is often called a *barrier function* and, in a sense, is opposite to the kind of exterior penalty function introduced in (25).

As discussed in Chapter 4, the basic idea of interior point methods is to start with a feasible point and a relatively *large* value of the parameter r. This will prevent the algorithm from approaching the boundary of the feasible region. At each subsequent iteration, the value of r is monotonically *decreased* in such a way that the resultant problem is relatively easy to solve if the optimal solution of its immediate predecessor is used as the starting point. Mathematically, the sequence of solutions $\{\mathbf{x}^k\}$ can be shown to converge to a local minimum in much the same way that the exterior penalty method was shown to converge earlier.

Example 18

To clarify these ideas consider the problem

$$\text{Minimize}\left\{f(\mathbf{x}) = 2x_1^2 + 9x_2 \;\text{ subject to }\; x_1 + x_2 \geq 4\right\}.$$

Using (27) the augmented objective function is

$$B(r,\mathbf{x}) = 2x_1^2 + 9x_2 + r\left(\frac{-1}{-\mathrm{x}_1 - \mathrm{x}_2 + 4}\right)$$

with gradient

$${}_{\mathbf{x}}B(r,\mathbf{x}) = \begin{pmatrix} 4x_1 - r(-x_1 - x_2 + 4)^{-2} \\ 9 - r(-x_1 - x_2 + 4)^{-2} \end{pmatrix}.$$

Setting the gradient vector to zero and solving for the stationary point yields

$$x_1(r) = 2.25 \text{ and } x_2(r) = 0.333\sqrt{r} + 1.75 \text{ for all } r > 0.$$

In the limit as r approaches 0, these values become $x_1 = 2.25$ and $x_2 = 1.75$ with $f(\mathbf{x}) = 25.875$ which is the optimal solution to the original problem. Figure 21 depicts the feasible region and several isovalue contours of $f(\mathbf{x})$. Also shown is the locus of optimal solutions for $B(r,\mathbf{x})$ starting with $r_0 = 20$ and decreasing to 0.

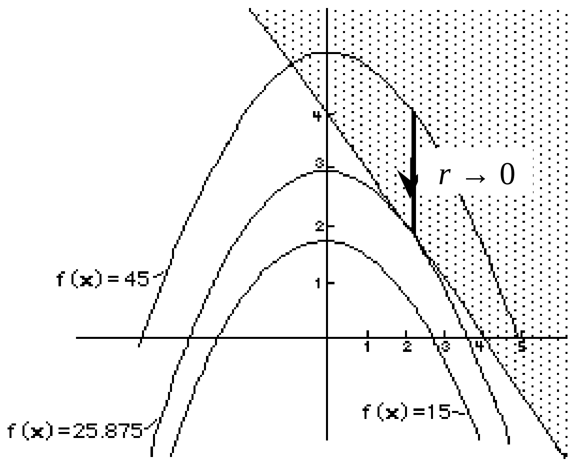

Figure 21. Graphical illustration of the barrier search procedure

A Mixed Barrier-Penalty Function

Equality constraints, though not discussed up until now, can be handled efficiently with a penalty function. Let us consider the following problem.

$$\begin{aligned} &\text{Minimize } f(\mathbf{x}) \\ &\text{subject to } h_i(\mathbf{x}) = 0, \; i = 1,\dots,p \\ &\qquad g_i(\mathbf{x}) \le 0, \; i = 1,\dots,m \end{aligned}$$

The most common approach to implementing the sequential unconstrained minimization technique for this model is to form the *logarithmic-quadratic loss function*

$$LQ(r_k,\mathbf{x}) = f(\mathbf{x}) - r_k \sum_{i=1}^{m} \ln(-g_i(\mathbf{x})) + \frac{1}{r_k} \sum_{i=1}^{p} h_i(\mathbf{x})^2 .$$

The algorithm finds the unconstrained minimizer of $LQ(r_k,\mathbf{x})$ over the set $\{\mathbf{x} : g_i(\mathbf{x}) < 0, i = 1,\dots,m\}$ for a sequence of scalar parameters $\{r_k\}$ strictly decreasing to zero. All convergence, duality, and convexity results for $LQ(r_k,\mathbf{x})$ are similar to those for the pure penalty and barrier functions.

Note that for a given r, the stationarity condition is

$$f(\mathbf{x}(r)) - \sum_{i=1}^{m} \frac{r}{-g_i(\mathbf{x}(r))} \quad g_i(\mathbf{x}(r)) + \sum_{i=1}^{p} \frac{2h_i(\mathbf{x}(r))}{r} \quad h_i(\mathbf{x}(r)) = 0$$

Fiacco and McCormick were able to show that as $r \to 0$, $r / g_i(\mathbf{x}(r)) \to \mu_i^*$, the optimal Lagrange multiplier for the *i*th inequality constraint, and $2h_i(\mathbf{x}(r))/r \to \lambda_i^*$, the optimal Lagrange multiplier for the *i*th equality constraint. This result is suggested by the fractions in the above summations.

Summary

Penalty and barrier methods are among the most powerful class of algorithms available for attacking general nonlinear optimization problems. This statement is supported by the fact that these techniques will converge to at least a local minimum in most cases, regardless of the convexity characteristics of the objective function and constraints. They work well even in the presence of cusps and similar anomalies that can stymie other approaches.

Of the two classes, the exterior methods must be considered preferable. The primary reasons are as follows.

1. Interior methods cannot deal with equality constraints without cumbersome modifications to the basic approach.
2. Interior methods demand a feasible starting point. Finding such a point often presents formidable difficulties in and of itself.
3. Interior methods require that the search never leave the feasible region. This significantly increases the computational effort associated with the line search segment of the algorithm.

Although penalty and barrier methods met with great initial success, their slow rates of convergence due to ill-conditioning of the associated Hessian led researchers to pursue other approaches. With the advent of interior point methods for linear programming, algorithm designers have taken a fresh look at penalty methods and have been able to achieve much greater efficiency than previously thought possible (e.g., see Nash and Sofer [1993]).

A.2 Primal Methods

In solving a nonlinear program, primal methods work on the original problem directly by searching the feasible region for an optimal solution. Each point generated in the process is feasible and the value of the objective function constantly decreases. These methods have three significant advantages: (1) if they terminate before confirming optimality (which is very often the case with all procedures), the current point is feasible; (2) if they generate a convergent sequence, it can usually be shown that the limit point of that sequence must be at least a local minimum; (3) they do not rely on special structure, such as convexity, so they are quite general. Notable disadvantages are that they require a phase 1 procedure to obtain an initial feasible point and that they are all plagued, particularly when the problem constraints are nonlinear, with computational difficulties arising from the need to remain within the feasible region from one iteration to the next. The convergence rates of primal methods are competitive with those of other procedures, and for problems with linear constraints, they are often among the most efficient.

Primal methods, often called *feasible direction methods*, embody the same philosophy as the techniques of unconstrained minimization but are designed to deal with inequality constraints. Briefly, the idea is to pick a starting point satisfying the constraints and to find a direction such that (i) a small move in that direction remains feasible, and (ii) the objective function improves. One then moves a finite distance in the determined direction, obtaining a new and better point. The process is repeated until no direction satisfying both (i) and (ii) can be found. In general, the terminal point is a constrained local (but not necessarily global) minimum of the problem. A direction satisfying both (i) and (ii) is called a *usable feasible direction*. There are many ways of choosing such directions, hence many different primal methods. We now present a popular one based on linear programming.

Zoutendijk's Method

Once again, we consider problem (23) with constraint set is $\boldsymbol{S} = \{\mathbf{x} : g_i(\mathbf{x}) \leq 0,\ i = 1,\ldots,m\}$. Assume that a starting point $\mathbf{x}^0 \quad \boldsymbol{S}$ is available. The problem is to choose a vector $\mathbf{d}$ whose direction is both usable and feasible. Let $g_i(\mathbf{x}^0) = 0,\ i \quad I$, where the indices in I correspond to the binding constraints at $\mathbf{x}^0$. For feasible direction $\mathbf{d}$, a small move along this vector beginning at the point $\mathbf{x}^0$ makes no binding constraints negative, i.e.,

$$\frac{d}{dt} g_i(\mathbf{x}^0 + t\mathbf{d})\Big|_{t=0} = \quad g_i(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d} \leq 0, \quad i \quad I$$

For a minimization objective, a usable feasible vector has the additional property that

$$\frac{d}{dt} f(\mathbf{x}^0+t\mathbf{d})\Big|_{t=0} = \quad f(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d} < 0$$

Therefore, the function initially decreases along the vector. In searching for a "best" vector **d** along which to move, one could choose that feasible vector minimizing $\quad f(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d}$. If some of the binding constraints were nonlinear, however, this could lead to certain difficulties. In particular, starting at $\mathbf{x}^0$ the feasible direction $\mathbf{d}^0$ that minimizes $\quad f(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d}$ is the projection of $- \quad f(\mathbf{x}^0)$ onto the tangent plane generated by the binding constraints at $\mathbf{x}^0$. Because the constraint surface is curved, movement along $\mathbf{d}^0$ for any finite distance violates the constraint. Thus a recovery move must be made to return to the feasible region. Repetitions of the procedure lead to inefficient zigzagging. As a consequence, when looking for a locally best direction it is wise to choose one that, in addition to decreasing *f*, also moves away from the boundaries of the nonlinear constraints. The expectation is that this will avoid zigzagging. Such a direction is the solution of the following problem.

$$\text{Minimize } \ \xi \tag{28a}$$

$$\text{subject to } \quad g_i(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d} - \phi_i\xi \le 0, \quad i \quad I \tag{28b}$$

$$f(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d} - \xi \ \le \ 0 \tag{28c}$$

$$\mathbf{d}^{\mathrm{T}}\mathbf{d} \ = \ 1 \tag{28d}$$

where $0 \le \phi_i \le 1$ is selected by the user. If all $\phi_i = 1$, then any vector $(\mathbf{d}, \xi)$ satisfying (28b) - (28c) with $\xi < 0$ is a usable feasible direction. That with minimum ξ value is a best direction which simultaneously makes $\quad f(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d}$ and $\quad g_i(\mathbf{x}^0)^{\mathrm{T}}\mathbf{d}$ as negative as possible; i.e., steers away from the nonlinear constraint boundaries. Other values of ϕ_i enable one to emphasize certain constraint boundaries relative to others. Equation (28d) is a normalization requirement ensuring that ξ is finite. If it were not included and a vector $(\mathbf{d}, \xi)$ existed satisfying Eqs. (28b) - (28c) with ξ negative, then ξ could be made to approach $-\infty$, since (28b) - (28c) are not homogeneous. Other normalizations, such as $|d_j| \le 1$ for all *j*, are also possible.

Because the vectors $\quad f$ and $\quad g_i$ are evaluated at a fixed point $\mathbf{x}^0$, the above direction-finding problem is almost linear, the only nonlinearity being (28d). Zoutendijk showed that this constraint can be handled by a modified version of the simplex method so problem (28) may be solved with reasonable efficiency. Note that if some of the constraints in the original nonlinear program (1) were given as equalities, the algorithm would have to be modified slightly.

Of course, once a direction has been determined, the step size must still be found. This problem may be dealt with in almost the same manner as in the unconstrained case. It is still desirable to minimize the objective function along the vector **d**, but now no constraint may be violated. Thus t is determined to minimize $f(\mathbf{x}^k + t\mathbf{d}^k)$ subject to the constraint $\mathbf{x}^k + t\mathbf{d}^k \quad S$. Any of the techniques discussed in Section 10.6 can be used. A new point is thus determined and the direction-finding problem is re-solved. If at some point the minimum $\xi \geq 0$, then there is no feasible direction satisfying $\quad f(\mathbf{x}^0)^\mathrm{T}\mathbf{d} < 0$ and the procedure terminates. The final point will generally be a local minimum of the problem. Zoutendijk showed that for convex programs the procedure converges to the global minimum.

A.3 Sequential Quadratic Programming

Successive linear programming (SLP) methods solve a sequence of linear approximations to the original nonlinear program. In this respect they are similar to Zoutendijk's method but they do not require that feasibility be maintained at each iteration. Recall that if $f(\mathbf{x})$ is a nonlinear function and $\mathbf{x}^c$ is the current value for $\mathbf{x}$, then the first order Taylor series expansion of $f(\mathbf{x})$ around $\mathbf{x}^c$ is

$$f(\mathbf{x}) = f(\mathbf{x}^c + \Delta\mathbf{x}) \quad f(\mathbf{x}^c) + \quad f(\mathbf{x}^c)(\Delta\mathbf{x}) \tag{29}$$

where $\Delta\mathbf{x}$ is the direction of movement. Given initial values for the variables, in SLP all nonlinear functions are replaced by their linear approximations as in Eq. (29). The variables in the resulting LP are the Δx_j's representing changes from the current values. It is common to place upper and lower bounds on each Δx_j, given that the linear approximation is reasonably accurate only in some neighborhood of the initial point.

The resulting linear program is solved and if the new point provides an improvement it becomes the incumbent and the process is repeated. If the new point does not yield an improvement, the step bounds may need to be reduced or we may be close enough to an optimum to stop. Successive points generated by this procedure need not be feasible even if the initial point is. However, the amount of infeasibility generally is reduced as the iterations proceed.

Successive quadratic programming (SQP) methods solve a sequence of quadratic programming approximations to the original nonlinear program (Fan et al. [1988]). By definition, QPs have a quadratic objective function, linear constraints, and bounds on the variables. A number of efficient procedures are available for solving them. As in SLP, the linear constraints are first order approximations of the actual constraints about the current point. The quadratic objective function used, however, is not just the second order Taylor series approximation to the original objective function but a variation based on the Lagrangian.

We will derive the procedure for the equality constrained version of a nonlinear program.

$$\text{Minimize } \; f(\mathbf{x}) \tag{30a}$$

$$\text{subject to } \; h_i(\mathbf{x}) = 0, \; i = 1,\dots,m \tag{30b}$$

The Lagrangian for this problem is $L(\mathbf{x}, \lambda) = f(\mathbf{x}) - \lambda^T h(\mathbf{x})$. Recall that first order necessary conditions for the point $\mathbf{x}^c$ to be a local minimum of Problem (30) are that there exist Lagrange multiplies λ^c such that

$$L_x(\mathbf{x}^c, \lambda^c) = \quad f(\mathbf{x}^c) - (\lambda^c)^T \quad \mathbf{h}(\mathbf{x}^c) = \mathbf{0} \tag{31}$$

and $\qquad \mathbf{h}(\mathbf{x}^c) = \mathbf{0}$

Applying Newton's method to solve the system of equations (31), requires their linearization at the current point yielding the linear system

$$\begin{bmatrix} L_{\mathrm{x}}^{2}(\mathbf{x}^{c},\lambda^{c}) & -\,{}^{2}\mathbf{h}(\mathbf{x}^{c}) \\ -\,{}^{2}\mathbf{h}(\mathbf{x}^{c})^{T} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \Delta\mathbf{x} \\ \Delta\lambda \end{pmatrix} = \begin{pmatrix} L_{\mathrm{x}}(\mathbf{x}^{c},\lambda^{c}) \\ \mathbf{h}(\mathbf{x}^{c}) \end{pmatrix} \tag{32}$$

It is easy to show that if ($\Delta\mathbf{x}, \Delta\lambda$)) satisfies Eq. (32) then ($\Delta\mathbf{x}, \lambda^c + \Delta\lambda$)) will satisfy the necessary conditions for the optimality of the following QP.

$$\text{Minimize } \ f(\mathbf{x}^{c})(\Delta\mathbf{x}) + \tfrac{1}{2}\Delta\mathbf{x}^{T} \ L_{\mathrm{x}}^{2}(\mathbf{x}^{c},\lambda^{c})\Delta\mathbf{x} \tag{33a}$$

$$\text{subject to } \ \mathbf{h}(\mathbf{x}^{c}) + \ \mathbf{h}(\mathbf{x}^{c})\Delta\mathbf{x} = \mathbf{0} \tag{33b}$$

On the other hand, if $\Delta\mathbf{x}^* = \mathbf{0}$ is the solution to this problem, we can show that $\mathbf{x}^c$ satisfies the necessary conditions (31) for a local minimum of the original problem. First, since $\Delta\mathbf{x}^* = \mathbf{0}$, $\mathbf{h}(\mathbf{x}^c) = \mathbf{0}$ and $\mathbf{x}^c$ is feasible to Problem (30). Now, because $\Delta\mathbf{x}^*$ solves Problem (33), there exists a λ^* such that the gradient of the Lagrangian function for (33) evaluated at $\Delta\mathbf{x}^* = \mathbf{0}$ is also equal to zero; i.e., $f(\mathbf{x}^c) - (\lambda^*)^T \ \mathbf{h}(\mathbf{x}^c) = \mathbf{0}$. The Lagrange multipliers λ^* can serve as the Lagrange multipliers for the original problem and hence the necessary conditions (31) are satisfied by ($\mathbf{x}^c, \lambda^*$).

The extension to inequality constraints is straightforward; they are linearized and included in the Lagrangian when computing the Hessian matrix, **L**, of the Lagrangian. Linear constraints and variable bounds contained in the original problem are included directly in the constraint region of Eq. (33b). Of course, the matrix **L** need not be positive definite, even at the optimal solution of the NLP, so the QP may not have a minimum. Fortunately, positive definite approximations of **L** can be used so the QP will have an optimal solution if it is feasible. Such approximations can be obtained by a slight modification of the popular BFGS updating formula used in unconstrained minimization. This formula requires only the gradient of the Lagrangian function so second derivatives of the problem functions need not be computed.

Because the QP can be derived from Newton's method applied to the necessary conditions for the optimum of the NLP, if one simply accepts the solution of the QP as defining the next point, the algorithm behaves like Newton's method; i.e., it converges rapidly near an optimum but may not converge from a poor initial point. If $\Delta\mathbf{x}$ is viewed as a search direction, the convergence properties can be improved. However, since both objective function improvement and reduction of the constraint infeasibilities need to be taken into accounted, the function to be minimized in the line search process must incorporate both. Two possibilities that have been suggested are the exact penalty function and the Lagrangian function. The Lagrangian is suitable for the following reasons:

1. On the tangent plane to the active constraints, it has a minimum at the optimal solution to the NLP.
2. It initially decreases along the direction $\Delta\mathbf{x}$.

If the penalty weight is large enough, the exact penalty function also has property (2) and is minimized at the optimal solution of the NLP.

Relative advantages and disadvantages:

Table A2, taken from Lasdon et al. [1996], summarizes the relative merits of SLP, SQP, and GRG algorithms, focusing on their application to problems with many nonlinear equality constraints. One feature appears as both an advantage and a disadvantage — whether or not the algorithm can violate the nonlinear constraints of the problem by relatively large amounts during the solution process.

SLP and SQP usually generate points yielding large violations of the constraints. This can cause difficulties, especially in models with log or fractional power expressions, since negative arguments for these functions are possible. Such problems have been documented in reference to complex chemical process examples in which SLP and some exterior penalty-type algorithms failed, whereas an implementation of the GRG method succeeded and was quite efficient. On the other hand, algorithms that do not attempt to satisfy the equalities at each step can be faster than those that do. The fact that SLP and SQP satisfy all linear constraints at each iteration should ease the aforementioned difficulties but do not eliminate them.

There are situations in which the optimization process must be interrupted before the algorithm has reached optimality and the current point must be used or discarded. Such cases are common in on-line process control where temporal constraints force immediate decisions. In these situations, maintaining feasibility during the optimization process may be a requirement for the optimizer inasmuch as constraint violations make a solution unusable. Clearly, all three algorithms have advantages that will dictate their use in certain situations. For large problems, SLP software is used most widely because it is relatively easy to implement given a good LP system. Nevertheless, large-scale versions of GRG and SQP have become increasingly popular.

Table A2. Relative Merits of SLP, SQP, and GRG Algorithms

Algorithm	Relative advantages	Relative disadvantage
SLP	· Easy to implement · Widely used in practice · Rapid convergence when optimum is at a vertex · Can handle very large problems · Does not attempt to satisfy equalities at each iteration · Can benefit from improvements in LP solvers	· May converge slowly on problems with nonvertex optima · Will usually violate nonlinear constraints until convergence, often by large amounts
SQP	· Usually requires fewest functions and gradient evaluations of all three algorithms (by far) · Does not attempt to satisfy equalities at each iteration	· Will usually violate nonlinear constraints until convergence, often by large amounts · Harder than SLP to implement · Requires a good QP solver
GRG	· Probably most robust of all three methods · Versatile--especially good for unconstrained or linearly constrained problems but also works well for nonlinear constraints · Can utilize existing process simulators employing Newton’s method · Once it reaches a feasible solution it remains feasible and then can be stopped at any stage with an improved solution	· Hardest to implement · Needs to satisfy equalities at each step of the algorithm

A.3 Exercises

33. Solve the problem given below with an exterior penalty function method, and then repeat the calculations using a barrier function method.

$$\begin{aligned} \text{Minimize } & x_1^2 + 4x_2^2 - 8x_1 - 16x_2 \\ \text{subject to } & x_1 + x_2 \le 5 \\ & 0 \le x_1 \le 3,\ x_2 \ge 0 \end{aligned}$$

34. Perform 5 iterations of the sequential unconstrained minimization technique using the logarithmic-quadratic loss function on the problem below. Let $\mathbf{x}^0 = (0, 0)$, $r_0 = 2$ and put $r_{k+1} \leftarrow r_k/2$ after each iteration.

$$\begin{aligned} \text{Minimize } & x_1^2 + 2x_2^2 \\ \text{subject to } & 4x_1 + x_2 \le 6 \\ & x_1 + x_2 = 3 \\ & x_1 \ge 0,\ x_2 \ge 0 \end{aligned}$$

35. Repeat the preceding exercise using Zoutendijk's procedure. Use the normalization $-1 \le d_j \le 1$, $j = 1, 2$, to permit solution by linear programming.

36. Consider the following separable nonlinear program.

$$\begin{aligned} \text{Minimize } & 5x_1^2 - 10x_1 - 10x_2\log_{10}x_2 \\ \text{subject to } & x_1^2 + 2x_2^2 \le 4,\ x_1 \ge 0,\ x_2 \ge 0 \end{aligned}$$

a. Approximate the separable functions with piecewise linear functions and solve the resultant model using linear programming. If necessary assume $0\log_{10}0 = 0$.

b. Solve the original problem using an penalty function approach.

c. Perform at least 4 iterations of Zoutendijk's procedure.

37. Solve the following problem using Zoutendijk's procedure. Start with $\mathbf{x}^0 = (0, 3/4)$.

$$\begin{aligned}
\text{Minimize}\quad & 2x_1^2 + 2x_2^2 - 2x_1x_2 - 4x_1 - 6x_2 \\
\text{subject to}\quad & x_1 + 5x_2 \le 5 \\
& 2x_1^2 + x_2 \le 0 \\
& x_1 \ge 0,\ x_2 \ge 0
\end{aligned}$$

38. Solve the relaxation of the redundancy problem when the budget for components is \$500 (the value of C). Use the data in the table below.

$$\text{Maximize}\quad \prod_{j=1}^{n} \left(1 - (1 - r_j)^{1+x_j}\right)$$

$$\text{subject to}\quad \sum_{j=1}^{n} c_j x_j \le C,\ \ x_j \ge 0,\ \ j = 1,\ldots,n$$

Item, j	1	2	3	4
Reliability, r_j	0.9	0.8	0.95	0.75
Cost per item, c_j	100	50	40	200

39. Consider the following quadratic programming.

$$\begin{aligned}
\text{Minimize}\quad f(\mathbf{x}) = {} & 2x_1^2 + 20x_2^2 + 43x_3^2 + 12x_1x_2 - 16x_1x_3 \\
& - 56x_2x_3 + 8x_1 + 20x_2 + 6x_3 \\
\text{subject to}\quad & 3x_1 + 2x_2 + 5x_3 \le 35 \\
& x_1 + 2x_2 + 3x_3 \ge 5 \\
& -x_1 + 2x_2 - 5x_3 \le 3 \\
& 5x_1 - 3x_2 + 2x_3 \le 30 \\
& x_1 \ge 0,\ x_2 \ge 0,\ x_3 \ge 0
\end{aligned}$$

a. Write out the KKT conditions then set up the appropriate linear programming model and solve with a restricted basis entry rule. Is the solution a global optimum? Explain.

b. Use an NLP code to find the optimum.

40. Use an NLP code to solve the problem in the preceding exercise but this time maximize rather than minimize. Because the maximization problem may have local solutions, try different starting points. Confirm that you have found the global maximum by solving the problem by hand.

Bibliography

Bazaraa, M.S., H.D. Sherali and C. M. Shetty, *Nonlinear Programming: Theory and Algorithms,* Second Edition, John Wiley & Sons, New York, 1993.

Fan, Y, S. Sarkar and L. Lasdon, "Experiments with Successive Quadratic Programming Algorithms," *Journal of Optimization Theory and Applications*, Vol. 56, No. 3, pp. 359-383, 1988.

Fiacco, A.V. and G.P. McCormick, *Nonlinear Programming: Sequential Unconstrained Minimization Techniques*, John Wiley & Sons, New York, 1968.

Fletcher, R. *Practical Methods of Optimization*, Second Edition, John Wiley & Sons, New York, 1987.

R. Horst and H. Tuy, *Global Optimization: Deterministic Approaches*, Third Edition, Springer-Verlag, Berlin, 1995.

Lasdon, L., J. Plummer and A. Warren, "Nonlinear Programming," in M. Avriel and B. Golany (eds.), *Mathematical Programming for Industrial Engineers*, Chapter 6, pp. 385-485, Marcel Dekker, New York, 1996.

Luenberger, D.G., *Linear and Nonlinear Programming*, Second Edition, Addison Wesley, Reading, MA, 1984.

Nash, S.G. and A. Sofer, “A Barrier Method for Large-Scale Constrained Optimization,” *ORSA Journal on Computing*, Vol. 5, No. 1, pp. 40-53, 1993.

Nash, S.G. and A. Sofer, *Linear and Nonlinear Programming*, McGraw Hill, New York, 1996.

Zoutendijk, G., *Methods of Feasible Directions*, Elsevier, Amsterdam, 1960.