5.5 Gradient Projection and Reduced Gradient Methods

Rosen's gradient projection method is based on projecting the search direction into the subspace tangent to the active constraints. Let us first examine the method for the case of linear constraints [7]. We define the constrained problem as

$$\begin{aligned} &\textbf{minimize} && f(\mathbf{x}) \\ &\textbf{such that} && g_j(\mathbf{x}) = \sum_{i=1}^{n} a_{ji}x_i - b_j \geq 0, \qquad j = 1,\ldots,n_g \; . \end{aligned} \tag{5.5.1}$$

In vector form

$$g_j = \mathbf{a}_j^T\mathbf{x} - b_j \geq 0 \; . \tag{5.5.2}$$

If we select only the r active constraints ($j \in I_A$), we may write the constraint equations as

$$\mathbf{g}_a = \mathbf{N}^T\mathbf{x} - \mathbf{b} = 0 \, , \tag{5.5.3}$$

where $\mathbf{g}_a$ is the vector of active constraints and the columns of the matrix $\mathbf{N}$ are the gradients of these constraints. The basic assumption of the gradient projection method is that $\mathbf{x}$ lies in the subspace tangent to the active constraints. If

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha\mathbf{s} \, , \tag{5.5.4}$$

and both $\mathbf{x}_i$ and $\mathbf{x}_{i+1}$ satisfy Eq. (5.5.3), then

$$\mathbf{N}^T\mathbf{s} = 0 \; . \tag{5.5.5}$$

If we want the steepest descent direction satisfying Eq. (5.5.5), we can pose the problem as

$$\begin{aligned} \textbf{minimize} \quad & \mathbf{s}^T\nabla f \\ \textbf{such that} \quad & \mathbf{N}^T\mathbf{s} = 0 \, , \\ \textbf{and} \quad & \mathbf{s}^T\mathbf{s} = 1 \; . \end{aligned} \tag{5.5.6}$$

That is, we want to find the direction with the most negative directional derivative which satisfies Eq. (5.5.5). We use Lagrange multipliers $\boldsymbol{\lambda}$ and μ to form the Lagrangian

$$\mathcal{L}(\mathbf{s},\boldsymbol{\lambda},\mu) = \mathbf{s}^T\nabla f - \mathbf{s}^T\mathbf{N}\boldsymbol{\lambda} - \mu(\mathbf{s}^T\mathbf{s} - 1) \; . \tag{5.5.7}$$

The condition for $\mathcal{L}$ to be stationary is

$$\frac{\partial\mathcal{L}}{\partial\mathbf{s}} = \nabla f - \mathbf{N}\boldsymbol{\lambda} - 2\mu\mathbf{s} = 0 \; . \tag{5.5.8}$$

Premultiplying Eq. (5.5.8) by $\mathbf{N}^T$ and using Eq. (5.5.5) we obtain

$$\mathbf{N}^T\nabla f - \mathbf{N}^T\mathbf{N}\boldsymbol{\lambda} = 0 \, , \tag{5.5.9}$$

or

$$\boldsymbol{\lambda} = (\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T\nabla f \; . \tag{5.5.10}$$

So that from Eq. (5.5.8)

$$\mathbf{s} = \frac{1}{2\mu}[I - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T]\nabla f = \frac{1}{2\mu}\mathbf{P}\nabla f \ . \tag{5.5.11}$$

$\mathbf{P}$ is the projection matrix defined in Eq. (5.3.8). The factor of $1/2\mu$ is not significant because $\mathbf{s}$ defines only the direction of search, so in general we use $\mathbf{s} = -\mathbf{P}\nabla f$. To show that $\mathbf{P}$ indeed has the projection property, we need to prove that if $\mathbf{w}$ is an arbitrary vector, then $\mathbf{Pw}$ is in the subspace tangent to the active constraints, that is $\mathbf{Pw}$ satisfies

$$\mathbf{N}^T\mathbf{Pw} = 0 \ . \tag{5.5.12}$$

We can easily verify this by using the definition of $\mathbf{P}$.

Equation (5.3.8) which defines the projection matrix $\mathbf{P}$ does not provide the most efficient way for calculating it. Instead it can be shown that

$$\mathbf{P} = \mathbf{Q}_2^T\mathbf{Q}_2 \, , \tag{5.5.13}$$

where the matrix $\mathbf{Q}_2$ consists of the last $n - r$ rows of the $\mathbf{Q}$ factor in the QR factorization of $\mathbf{N}$ (see Eq. (5.3.9)).

A version of the gradient projection method known as the *generalized reduced gradient* method was developed by Abadie and Carpentier [8]. As a first step we select r linearly independent rows of $\mathbf{N}$, denote their transpose as $\mathbf{N}_1$ and partition $\mathbf{N}^T$ as

$$\mathbf{N}^T = [\mathbf{N}_1 \quad \mathbf{N}_2] \, . \tag{5.5.14}$$

Next we consider Eq. (5.5.5) for the components s_i of the direction vector. The r equations corresponding to $\mathbf{N}_1$ are then used to eliminate r components of $\mathbf{s}$ and obtain a reduced order problem for the direction vector.

Once we have identified $\mathbf{N}_1$ we can easily obtain $\mathbf{Q}_2$ which is given as

$$\mathbf{Q}_2^T = \begin{bmatrix} -\mathbf{N}_1^{-1}\mathbf{N}_2 \\ \mathbf{I} \end{bmatrix} \ . \tag{5.5.15}$$

Equation (5.5.15) can be verified by checking that $\mathbf{N}^T\mathbf{Q}_2^T = 0$, so that $\mathbf{Q}_2\mathbf{N} = 0$, which is the requirement that $\mathbf{Q}_2$ has to satisfy (see discussion following Eq. (5.3.11)).

After obtaining $\mathbf{s}$ from Eq. (5.5.11) we can continue the search with a one dimensional minimization, Eq. (5.5.4), unless $\mathbf{s} = 0$. When $\mathbf{s} = 0$ Eq. (5.3.7) indicates that the Kuhn-Tucker conditions may be satisfied. We then calculate the Lagrange multipliers from Eq. (5.3.6) or Eq. (5.3.11). If all the components of $\boldsymbol{\lambda}$ are nonnegative, the Kuhn-Tucker conditions are indeed satisfied and the optimization can be terminated. If some of the Lagrange multipliers are negative, it is an indication that while no progress is possible with the current set of active constraints, it may be possible to proceed by removing some of the constraints associated with negative Lagrange multipliers. A common strategy is to remove the constraint associated with the most negative Lagrange multiplier and repeat the calculation of $\mathbf{P}$ and $\mathbf{s}$. If $\mathbf{s}$

is now non-zero, a one-dimensional search may be started. If $\mathbf{s}$ remains zero and there are still negative Lagrange multipliers, we remove another constraint until all Lagrange multipliers become positive and we satisfy the Kuhn-Tucker conditions.

After a search direction has been determined, a one dimensional search must be carried out to determine the value of α in Eq. (5.5.4). Unlike the unconstrained case, there is an upper limit on α set by the inactive constraints. As α increases, some of them may become active and then violated. Substituting $\mathbf{x} = \mathbf{x}_i + \alpha\mathbf{s}$ into Eq. (5.5.2) we obtain

$$g_j = \mathbf{a}_j^T(\mathbf{x}_i + \alpha\mathbf{s}) - b_j \geq 0\,, \tag{5.5.16}$$

or

$$\alpha \leq -(\mathbf{a}_j^T\mathbf{x}_i - b_j)/\mathbf{a}_j^T\mathbf{s} = -g_j(\mathbf{x}_i)/\mathbf{a}_j^T\mathbf{s}\ . \tag{5.5.17}$$

Equation (5.5.17) is valid if $\mathbf{a}_j^T\mathbf{s} < 0$. Otherwise, there is no upper limit on α due to the jth constraint. From Eq. (5.5.17) we get a different α, say α_j for each constraint. The upper limit on α is the minimum

$$\bar{\alpha} = \min_{\alpha_j > 0,\ j \ni I_A} \alpha_j\ . \tag{5.5.18}$$

At the end of the move, new constraints may become active, so that the set of active constraints may need to be updated before the next move is undertaken.

The version of the gradient projection method presented so far is an extension of the steepest descent method. Like the steepest descent method, it may have slow convergence. The method may be extended to correspond to Newton or quasi-Newton methods. In the unconstrained case, these methods use a search direction defined as

$$\mathbf{s} = -\mathbf{B}\nabla f\,, \tag{5.5.19}$$

where $\mathbf{B}$ is the inverse of the Hessian matrix of f or an approximation thereof. The direction that corresponds to such a method in the subspace tangent to the active constraints can be shown [4] to be

$$\mathbf{s} = -\mathbf{Q}_2^T(\mathbf{Q}_2^T\mathbf{A}_L\mathbf{Q}_2)^{-1}\mathbf{Q}_2\nabla f\,, \tag{5.5.20}$$

where $\mathbf{A}_L$ is the Hessian of the Lagrangian function or an approximation thereof.

The gradient projection method has been generalized by Rosen to nonlinear constraints [9]. The method is based on linearizing the constraints about $\mathbf{x}_i$ so that

$$\mathbf{N} = [\nabla g_1(\mathbf{x}_i),\ \nabla g_2(\mathbf{x}_i), \ldots, \nabla g_r(\mathbf{x}_i)]\ . \tag{5.5.21}$$

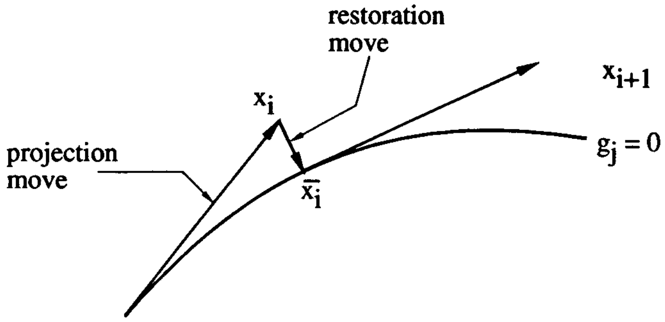

Figure 5.5.1 Projection and restoration moves.

The main difficulty caused by the nonlinearity of the constraints is that the one-dimensional search typically moves away from the constraint boundary. This is because we move in the tangent subspace which no longer follows exactly the constraint boundaries. After the one-dimensional search is over, Rosen prescribes a restoration move to bring $\mathbf{x}$ back to the constraint boundaries, see Figure 5.5.1.

To obtain the equation for the restoration move, we note that instead of Eq. (5.5.2) we now use the linear approximation

$$g_j \approx g_j(\mathbf{x}_i) + \nabla g_j^T(\bar{\mathbf{x}}_i - \mathbf{x}_i) \ . \tag{5.5.22}$$

We want to find a correction $\bar{\mathbf{x}}_i - \mathbf{x}_i$ in the tangent subspace (i.e. $\mathbf{P}(\bar{\mathbf{x}}_i - \mathbf{x}_i) = 0$) that would reduce g_j to zero. It is easy to check that

$$\bar{\mathbf{x}}_i - \mathbf{x}_i = -\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a(\mathbf{x}_i) \,, \tag{5.5.23}$$

is the desired correction, where $\mathbf{g}_a$ is the vector of active constraints. Equation (5.5.23) is based on a linear approximation, and may therefore have to be applied repeatedly until $\mathbf{g}_a$ is small enough.

In addition to the need for a restoration move, the nonlinearity of the constraints requires the re-evaluation of $\mathbf{N}$ at each point. It also complicates the choice of an upper limit for α which guarantees that we will not violate the presently inactive constraints. Haug and Arora [10] suggest a procedure which is better suited for the nonlinear case. The first advantage of their procedure is that it does not require a one-dimensional search. Instead, α in Eq. (5.5.4) is determined by specifying a desired specified reduction γ in the objective function. That is, we specify

$$f(\mathbf{x}_i) - f(\mathbf{x}_{i+1}) \approx \gamma f(\mathbf{x}_i) \ . \tag{5.5.24}$$

Using a linear approximation with Eq. (5.5.4) we get

$$\alpha^* = -\frac{\gamma f(\mathbf{x}_i)}{\mathbf{s}^T\nabla f} \ . \tag{5.5.25}$$

The second feature of Haug and Arora's procedure is the combination of the projection and the restoration moves as

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \alpha^*\mathbf{s} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a \,, \tag{5.5.26}$$

where Eqs. (5.5.4), (5.5.23) and (5.5.25) are used.

Example 5.5.1

Use the gradient projection method to solve the following problem

$$\begin{aligned} \textbf{minimize} \quad & f = x_1^2 + x_2^2 + x_3^2 + x_4^2 - 2x_1 - 3x_4 \\ \textbf{subject to} \quad & g_1 = 2x_1 + x_2 + x_3 + 4x_4 - 7 \geq 0\,, \\ & g_2 = x_1 + x_2 + x_3^2 + x_4 - 5.1 \geq 0\,, \\ & x_i \geq 0, \qquad i = 1, \ldots, 4\,. \end{aligned}$$

Assume that as a result of previous moves we start at the point $\mathbf{x}_0^T = (2, 2, 1, 0)$, $f(\mathbf{x}_0) = 5.0$, where the nonlinear constraint g_2 is slightly violated. The first constraint is active as well as the constraint on x_4. We start with a combined projection and restoration move, with a target improvement of 10% in the objective function. At $\mathbf{x}_0$

$$\mathbf{N} = \begin{bmatrix} 2 & 1 & 0 \\ 1 & 1 & 0 \\ 1 & 2 & 0 \\ 4 & 1 & 1 \end{bmatrix}, \qquad \mathbf{N}^T\mathbf{N} = \begin{bmatrix} 22 & 9 & 4 \\ 9 & 7 & 1 \\ 4 & 1 & 1 \end{bmatrix},$$

$$(\mathbf{N}^T\mathbf{N})^{-1} = \frac{1}{11}\begin{bmatrix} 6 & -5 & -19 \\ -5 & 6 & 14 \\ -19 & 14 & 73 \end{bmatrix},$$

$$\mathbf{P} = \mathbf{I} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{N}^T = \frac{1}{11}\begin{bmatrix} 1 & -3 & 1 & 0 \\ -3 & 9 & -3 & 0 \\ 1 & -3 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \qquad \nabla f = \begin{Bmatrix} 2 \\ 4 \\ 2 \\ -3 \end{Bmatrix}.$$

The projection move direction is $\mathbf{s} = -\mathbf{P}\nabla f = [8/11, -24/11, 8/11, 0]^T$. Since the magnitude of a direction vector is unimportant we scale $\mathbf{s}$ to $\mathbf{s}^T = [1, -3, 1, 0]$. For a 10% improvement in the objective function $\gamma = 0.1$ and from Eq. (5.5.25)

$$\alpha^* = -\frac{0.1f}{\mathbf{s}^T\nabla f} = -\frac{0.1 \times 5}{-8} = 0.0625\,.$$

For the correction move we need the vector $\mathbf{g}_a$ of constraint values , $\mathbf{g}_a^T = (0, -0.1, 0)$, so the correction is

$$-\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \frac{-1}{110}\begin{Bmatrix} 4 \\ -1 \\ -7 \\ 0 \end{Bmatrix}.$$

Combining the projection and restoration moves, Eq. (5.5.26)

$$\mathbf{x}_1 = \begin{Bmatrix} 2 \\ 2 \\ 1 \\ 0 \end{Bmatrix} + 0.0625\begin{Bmatrix} 1 \\ -3 \\ 1 \\ 0 \end{Bmatrix} - \frac{1}{110}\begin{Bmatrix} 4 \\ -1 \\ -7 \\ 0 \end{Bmatrix} = \begin{Bmatrix} 2.026 \\ 1.822 \\ 1.126 \\ 0 \end{Bmatrix},$$

we get $f(\mathbf{x}_1) = 4.64$, $g_1(\mathbf{x}_1) = 0$, $g_2(\mathbf{x}_1) = 0.016$. Note that instead of 10% reduction we got only 7% due to the nonlinearity of the objective function. However, we did satisfy the nonlinear constraint.• • •

Example 5.5.2

Consider the four bar truss of Example 5.1.2. The problem of finding the minimum weight design subject to stress and displacement constraints was formulated as

$$\begin{aligned} \textbf{minimize} \quad & f = 3x_1 + \sqrt{3}x_2 \\ \textbf{subject to} \quad & g_1 = 3 - \frac{18}{x_1} - \frac{6\sqrt{3}}{x_2} \geq 0\,, \\ & g_2 = x_1 - 5.73 \geq 0\,, \\ & g_3 = x_2 - 7.17 \geq 0\,, \end{aligned}$$

where the x_i are non-dimensional areas

$$x_i = \frac{A_i E}{1000P}, \qquad i = 1, 2\ .$$

The first constraint represents a limit on the vertical displacement, and the other two represent stress constraints.

Assume that we start the search at the intersection of $g_1 = 0$ and $g_3 = 0$, where $x_1 = 11.61$, $x_2 = 7.17$, and $f = 47.25$. The gradients of the objective function and two active constraints are

$$\nabla f = \begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}, \qquad \nabla g_1 = \begin{Bmatrix} 0.1335 \\ 0.2021 \end{Bmatrix}, \qquad \nabla g_3 = \begin{Bmatrix} 0 \\ 1 \end{Bmatrix}, \qquad \mathbf{N} = \begin{bmatrix} 0.1335 & 0 \\ 0.2021 & 1 \end{bmatrix}.$$

Because $\mathbf{N}$ is nonsingular, Eq. (5.3.8) shows that $\mathbf{P} = 0$. Also since the number of linearly independent active constraints is equal to the number of design variables the tangent subspace is a single point, so that there is no more room for progress. Using Eqs. (5.3.6) or (5.3.11) we obtain

$$\boldsymbol{\lambda} = \begin{Bmatrix} 22.47 \\ -2.798 \end{Bmatrix}.$$

The negative multiplier associated with g_3 indicates that this constraint can be dropped from the active set. Now

$$\mathbf{N} = \begin{bmatrix} 0.1335 \\ 0.2021 \end{bmatrix}.$$

The projection matrix is calculated from Eq. (5.3.8)

$$\mathbf{P} = \begin{bmatrix} 0.6962 & -0.4600 \\ -0.4600 & 0.3036 \end{bmatrix}, \qquad \mathbf{s} = -\mathbf{P}\nabla f = \begin{Bmatrix} -1.29 \\ 0.854 \end{Bmatrix}.$$

We attempt a 5% reduction in the objective function, and from Eq. (5.5.25)

$$\alpha^* = \frac{0.05 \times 47.25}{[-1.29\ 0.854]\begin{Bmatrix} 3 \\ \sqrt{3} \end{Bmatrix}} = 0.988\ .$$

Since there was no constraint violation at $\mathbf{x}_0$ we do not need a combined projection and correction step, and

$$\mathbf{x}_1 = \mathbf{x}_0 + \alpha^* \mathbf{s} = \left\{ \begin{matrix} 11.61 \\ 7.17 \end{matrix} \right\} + 0.988 \left\{ \begin{matrix} -1.29 \\ 0.854 \end{matrix} \right\} = \left\{ \begin{matrix} 10.34 \\ 8.01 \end{matrix} \right\} .$$

At $\mathbf{x}_1$ we have $f(\mathbf{x}_1) = 44.89$, $g_1(\mathbf{x}_1) = -0.0382$. Obviously g_2 is not violated. If there were a danger of that we would have to limit α^* using Eq. (5.5.17). The violation of the nonlinear constraint is not surprising, and its size indicates that we should reduce the attempted reduction in f in the next move. At x_1, only g_1 is active so

$$\mathbf{N} = \nabla \mathbf{g}_1 = \left\{ \begin{matrix} 0.1684 \\ 0.1620 \end{matrix} \right\} .$$

The projection matrix is calculated to be

$$\mathbf{P} = \begin{bmatrix} 0.4806 & -0.4996 \\ -0.4996 & 0.5194 \end{bmatrix}, \qquad \mathbf{s} = -\mathbf{P}\nabla f = \left\{ \begin{matrix} -0.5764 \\ 0.5991 \end{matrix} \right\} .$$

Because of the violation we reduce the attempted reduction in f to 2.5%, so

$$\alpha^* = -\frac{0.025 \times 44.89}{[-0.567 \; 0.599] \left\{ \begin{matrix} 3 \\ \sqrt{3} \end{matrix} \right\}} = 1.62 .$$

We need also a correction due to the constraint violation ($\mathbf{g}_a = -0.0382$)

$$-\mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \left\{ \begin{matrix} 0.118 \\ 0.113 \end{matrix} \right\} .$$

Altogether

$$\mathbf{x}_2 = \mathbf{x}_1 + \alpha^* \mathbf{s} - \mathbf{N}(\mathbf{N}^T\mathbf{N})^{-1}\mathbf{g}_a = \left\{ \begin{matrix} 10.34 \\ 8.01 \end{matrix} \right\} - 1.62 \left\{ \begin{matrix} 0.576 \\ -0.599 \end{matrix} \right\} + \left\{ \begin{matrix} 0.118 \\ 0.113 \end{matrix} \right\} = \left\{ \begin{matrix} 9.52 \\ 9.10 \end{matrix} \right\} .$$

We obtain $f(\mathbf{x}_2) = 44.32$, $g_1(\mathbf{x}_2) = -0.0328$.

The optimum design is actually $\mathbf{x}^T = (9.464, 9.464)$, $f(\mathbf{x}) = 44.78$, so after two iterations we are quite close to the optimum design.• • •

5.6 The Feasible Directions Method

The feasible directions method [11] has the opposite philosophy to that of the gradient projection method. Instead of following the constraint boundaries, we try to stay as far away as possible from them. The typical iteration of the feasible direction method starts at the boundary of the feasible domain (unconstrained minimization techniques are used to generate a direction if no constraint is active).