Computer Aided Ship Design

Part I. Optimization Method

Ch. 7 Constrained Nonlinear Optimization Method

September, 2013
Prof. Myung-Il Roh

Department of Naval Architecture and Ocean Engineering,
Seoul National University of College of Engineering
Ch. 7 Constrained Nonlinear Optimization Method

7.1 Quadratic Programming (QP)
7.2 Sequential Linear Programming (SLP)
7.3 Sequential Quadratic Programming (SQP)
7.4 Determine the Search Direction of the Quadratic Programming Problem by Using the Simplex Method
7.5 Summary of the Sequential Quadratic Programming (SQP)
7.1 Quadratic Programming (QP)
Minimize  \( f(x) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 \)

Subject to  \( g_1(x) = -2x_1 - x_2 + 4 \leq 0 \)
\( g_2(x) = -x_1 - 2x_2 + 4 \leq 0 \)
\( x_1 \geq 0, x_2 \geq 0 \)

Minimum point:  \( x^* = (\frac{4}{3}, \frac{4}{3}), f(x^*) = \frac{2}{9} \)

Minimize  \( f(x) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 \)

Subject to  \( g_1(x) = -2x_1 - x_2 + 4 \leq 0 \)
\( g_2(x) = -x_1 - 2x_2 + 4 \leq 0 \)
\( -x_1 \leq 0, -x_2 \leq 0 \)

Inequality constraints are transformed to equality constraints by introducing the slack variable.

Quadratic programming problem
- Objective function: quadratic form
- Constraint: linear form

Minimum at Point A
\( x^* = (\frac{4}{3}, \frac{4}{3}), f(x^*) = \frac{2}{9} \)

Feasible region
[Review] 4.3: Finding the optimal solution for the quadratic objective function with linear inequality constraints problem by using the Kuhn-Tucker Necessary Condition, where $x_i$ are nonnegative (2/4)

**Minimize** \( f(x) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 \)

**Subject to** \( g_1(x) = -2x_1 - x_2 + 4 + s_1^2 = 0 \)

\[
g_2(x) = -x_1 - 2x_2 + 4 + s_2^2 = 0
\]

\[-x_1 + \delta_1^2 = 0, -x_2 + \delta_2^2 = 0\]

**Lagrange function**

\[
L(x, u, s, \zeta, \delta) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 + u_1(-2x_1 - x_2 + 4 + s_1^2)
+ u_2(-x_1 - 2x_2 + 4 + s_2^2) + \zeta_1(-x_1 + \delta_1^2) + \zeta_2(-x_2 + \delta_2^2)
\]

**Kuhn-Tucker necessary condition:** \( \nabla L(x, u, s, \zeta, \delta) = 0 \)

\[
\frac{\partial L}{\partial x_1} = 2x_1 - 2 - 2u_1 - u_2 - \zeta_1 = 0
\]

\[
\frac{\partial L}{\partial x_2} = -2x_1 - x_2 + 4 + s_1^2 = 0
\]

\[
\frac{\partial L}{\partial u_1} = 2u_1s_1 = 0
\]

\[
\frac{\partial L}{\partial u_2} = 2u_2s_2 = 0
\]

\[
\frac{\partial L}{\partial s_1} = \delta_1^2 - x_1 = 0
\]

\[
\frac{\partial L}{\partial s_2} = \delta_2^2 - x_2 = 0
\]

\[
\frac{\partial L}{\partial \zeta_1} = \delta_1^2 - x_1 = 0
\]

\[
\frac{\partial L}{\partial \zeta_2} = \delta_2^2 - x_2 = 0
\]

\[
\frac{\partial L}{\partial \delta_1} = 2\zeta_1\delta_1 = 0
\]

\[
\frac{\partial L}{\partial \delta_2} = 2\zeta_2\delta_2 = 0
\]

Minimum at Point A

\( x^* = \left(\frac{4}{3}, \frac{4}{3}\right), f(x^*) = \frac{2}{9} \)
Kuhn-Tucker necessary condition: \( \nabla L(x, u, s, \zeta, \delta) = 0 \)

\[
\begin{align*}
\frac{\partial L}{\partial x_1} &= 2x_1 - 2 - 2u_1 - u_2 - \zeta_1 = 0 \\
\frac{\partial L}{\partial u_1} &= -2x_1 - x_2 + 4 + s_1^2 = 0 \\
\frac{\partial L}{\partial s_1} &= 2u_1 s_1 = 0 \\
\frac{\partial L}{\partial \zeta_1} &= \delta_1^2 - x_1 = 0 \rightarrow \delta_1^2 = x_1 \\
\frac{\partial L}{\partial \delta_1} &= 2\zeta_1 \delta_1 = 0 \rightarrow 2\zeta_1 \delta_1^2 = 0 \\
\frac{\partial L}{\partial x_2} &= 2x_2 - 2 - u_1 - 2u_2 - \zeta_2 = 0 \\
\frac{\partial L}{\partial u_2} &= -x_1 - 2x_2 + 4 + s_2^2 = 0 \\
\frac{\partial L}{\partial s_2} &= 2u_2 s_2 = 0 \\
\frac{\partial L}{\partial \zeta_2} &= \delta_2^2 - x_2 = 0 \rightarrow \delta_2^2 = x_2 \\
\frac{\partial L}{\partial \delta_2} &= 2\zeta_2 \delta_2 = 0 \rightarrow 2\zeta_2 \delta_2^2 = 0
\end{align*}
\]

\( \zeta_1, \delta_1 \) and \( \zeta_2, \delta_2 \) are eliminated.

Reformulated Kuhn-Tucker necessary condition:

\[
\begin{align*}
\frac{\partial L}{\partial x_1} &= 2x_1 - 2 - 2u_1 - u_2 - \zeta_1 = 0 \\
\frac{\partial L}{\partial x_2} &= 2x_2 - 2 - u_1 - 2u_2 - \zeta_2 = 0 \\
\frac{\partial L}{\partial u_1} &= -2x_1 - x_2 + 4 + s_1^2 = 0 \\
\frac{\partial L}{\partial u_2} &= -x_1 - 2x_2 + 4 + s_2^2 = 0 \\
\frac{\partial L}{\partial s_1} &= 2u_1 s_1 = 0 \\
\frac{\partial L}{\partial s_2} &= 2u_2 s_2 = 0 \\
2\zeta_1 x_1 &= 0 \\
2\zeta_2 x_2 &= 0 \\
u_i, \zeta_i, \delta_i &\geq 0, \ i = 1, 2
\end{align*}
\]
Lagrangian function
\[ L(x, u, s, \zeta, \delta) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 + u_1(-2x_1 - x_2 + 4 + s_1^2) + u_2(-x_1 - 2x_2 + 4 + s_2^2) + \zeta_1(-x_1 + \delta_1^2) + \zeta_2(-x_2 + \delta_2^2) \]

Case #1: \( s_1 = s_2 = \zeta_1 = \zeta_2 = 0 \), (Point A)
\[ x_1 = x_2 = \frac{4}{3}, u_1 = u_2 = \frac{2}{3} \]

Case #2: \( u_1 = s_2 = \zeta_1 = \zeta_2 = 0 \), (Point B)
\[ x_1 = \frac{6}{5}, x_2 = \frac{2}{5}, u_2 = \frac{2}{5}, s_1^2 = -\frac{1}{5} \]
It has to be nonnegative.

Case #3: \( u_2 = s_1 = \zeta_1 = \zeta_2 = 0 \), (Point C)
\[ x_1 = \frac{7}{5}, x_2 = \frac{6}{5}, u_1 = \frac{2}{5}, s_2^2 = -\frac{1}{5} \]
It has to be nonnegative.

Case #4: \( u_1 = u_2 = \zeta_1 = \zeta_2 = 0 \), (Point D)
\[ x_1 = x_2 = 1, s_1^2 = s_2^2 = -1 \]
It has to be nonnegative.

Case #5: \( u_1 = u_2 = x_1 = x_2 = 0 \), (Point E)
\[ x_1 = x_2 = 0, s_1^2 = s_2^2 = -4, \zeta_1 = \zeta_2 = -2 \]
It has to be nonnegative.

Case #6: \( u_1 = s_2 = x_1 = x_2 = 0 \), (Point E)
\[ x_1 = x_2 = 0, s_1^2 = s_2^2 = -4, \zeta_1 = \zeta_2 = -2 \]
It has to be nonnegative.

Case #7: \( u_2 = s_1 = x_1 = x_2 = 0 \), (Point E)
\[ x_1 = x_2 = 0, s_1^2 = s_2^2 = -4, -2x_1 - x_2 + 4 + s_1^2 \neq 0 \]
The constraint is violated.

Case #8: \( s_1 = s_2 = x_1 = x_2 = 0 \), (Point E)
\[ x_1 = x_2 = 0, -2x_1 - x_2 + 4 + s_1^2 \neq 0, -x_1 - 2x_2 + 4 + s_2^2 \neq 0 \]
The constraint is violated.

Case #9: \( u_1 = s_2 = \zeta_2 = x_1 = 0 \), (Point F)
\[ x_1 = 0, x_2 = 2, u_1 = 1, s_1^2 = -2, \zeta_1 = -3 \]
It has to be nonnegative.

Case #10: \( u_2 = s_1 = \zeta_1 = x_2 = 0 \), (Point G)
\[ x_1 = 2, x_2 = 0, u_1 = 1, s_2^2 = -2, \zeta_2 = -3 \]
It has to be nonnegative.

Case #11: \( s_1 = s_2 = \zeta_1 = x_2 = 0 \), (Point G)
\[ x_1 = 2, x_2 = 0, \text{The constraint is violated.} \]

Case #12: \( u_2 = s_1 = \zeta_2 = x_1 = 0 \), (Point H)
\[ x_1 = 0, x_2 = 4, u_1 = 6, s_2^2 = 4, \zeta_1 = -14 \]
It has to be nonnegative.

Case #13: \( s_1 = s_2 = \zeta_2 = x_1 = 0 \), (Point H)
\[ x_1 = 0, x_2 = 4, \text{The constraint is violated.} \]

Case #14: \( u_1 = s_2 = \zeta_1 = x_2 = 0 \), (Point I)
\[ x_1 = 4, x_2 = 0, u_2 = 6, s_1^2 = 4, \zeta_2 = -14 \]
It has to be nonnegative.

Case #15: \( u_1 = u_2 = \zeta_1 = x_1 = 0 \), (Point J)
\[ x_1 = 0, x_2 = 1, s_1^2 = -3, s_2^2 = -2, \zeta_1 = -2 \]
It has to be nonnegative.

Case #16: \( u_1 = u_2 = \zeta_1 = x_2 = 0 \), (Point K)
\[ x_1 = 1, x_2 = 0, s_1^2 = -2, s_2^2 = -3, \zeta_2 = -2 \]
It has to be nonnegative.
### Summary for Solution of QP Problem Using the Kuhn-Tucker Necessary Condition

Minimize \[ f(x) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 \]

Subject to \[ \begin{align*}
-2x_1 - x_2 & \leq -4 \\
-x_1 - 2x_2 & \leq -4
\end{align*} \]
\[ x_1 \geq 0, x_2 \geq 0 \]

Subject to \[ \begin{align*}
-2x_1 - x_2 + 4 + s_1^2 &= 0 \\
-x_1 - 2x_2 + 4 + s_2^2 &= 0
\end{align*} \]
\[ x_1 + \delta_1^2 = 0, -x_2 + \delta_2^2 = 0 \]

**Lagrange function**
\[ L(x, u, s, \zeta, \delta) = x_1^2 + x_2^2 - 2x_1 - 2x_2 + 2 + u_1(-2x_1 - x_2 + 4 + s_1^2) + u_2(-x_1 - 2x_2 + 4 + s_2^2) + \zeta_1(-x_1 + \delta_1^2) + \zeta_2(-x_2 + \delta_2^2) \]
where, \( u_i, \zeta_i \geq 0 \)

**Kuhn-Tucker necessary condition:**
\[ \nabla L(x, u, s, \zeta, \delta) = 0 \]

\[ \begin{align*}
\frac{\partial L}{\partial x_1} &= -2 + 2x_1 - 2u_1 - u_2 - \zeta_1 = 0, \\
\frac{\partial L}{\partial x_2} &= -2 + 2x_2 - u_1 - 2u_2 - \zeta_2 = 0 \\
\frac{\partial L}{\partial u_1} &= -2x_1 - x_2 + 4 + s_1^2 = 0, \\
\frac{\partial L}{\partial u_2} &= -x_1 - 2x_2 + 4 + s_2^2 = 0 \\
\frac{\partial L}{\partial s_1} &= 2u_1s_1 = 0, \\
\frac{\partial L}{\partial s_2} &= 2u_2s_2 = 0 \\
\frac{\partial L}{\partial \zeta_1} &= 2\zeta_1\delta_1 = 0, \\
\frac{\partial L}{\partial \zeta_2} &= 2\zeta_2\delta_2 = 0 \\
\frac{\partial L}{\partial \delta_1} &= -x_1 + \delta_1^2 = 0, \\
\frac{\partial L}{\partial \delta_2} &= -x_2 + \delta_2^2 = 0
\end{align*} \]

where \( u_i, \zeta_i \geq 0 \)

Multiply both side of each equation \( ①, ② \) by \( s_1, s_2 \), respectively

Multiply both side of each equation \( ③, ④ \) by \( \delta_1, \delta_2 \), respectively
Solution Procedure of Quadratic Programming (QP) Problem
- Approximate the Original Problem as a Quadratic Programming Problem

\textbf{Minimize} \quad f(x + \Delta x) = f(x) + \nabla f^T(x) \Delta x + 0.5 \Delta x^T H \Delta x

The second-order Taylor series expansion of the objective function

\textbf{Subject to} \quad h_j(x + \Delta x) = h_j(x) + \nabla h_j^T(x) \Delta x = 0; \; j = 1 \text{ to } p

The first-order (linear) Taylor series expansion of the equality constraints

\quad g_j(x + \Delta x) = g_j(x) + \nabla g_j^T(x) \Delta x \leq 0; \; j = 1 \text{ to } m

The first-order (linear) Taylor series expansion of the inequality constraints

Define:
\[\bar{f} = f(x + \Delta x) - f(x), \quad e_j = -h_j(x), \quad b_j = -g_j(x),\]
\[c_i = \frac{\partial f(x)}{\partial x_i}, \quad n_{ij} = \frac{\partial h_j(x)}{\partial x_i}, \quad a_{ij} = \frac{\partial g_j(x)}{\partial x_i},\]
\[d_i = \Delta x_i\]

Matrix form

\textbf{Minimize} \quad \bar{f} = c^T_{(1\times n)} d_{(n \times 1)} + \frac{1}{2} d^T_{(1 \times n)} H_{(n \times n)} d_{(n \times 1)} : \text{Quadratic objective function}

\textbf{Subject to} \quad N^T_{(p \times n)} d_{(n \times 1)} = e_{(p \times 1)} : \text{Linear equality constraints}

\quad A^T_{(m \times n)} d_{(n \times 1)} \leq b_{(m \times 1)} : \text{Linear inequality constraints}
Solution Procedure of Quadratic Programming (QP) Problem
- Construction of Lagrange Function

Minimize
\[ \bar{f} = \mathbf{c}^T (1 \times n) \mathbf{d}_{(n \times 1)} + \frac{1}{2} \mathbf{d}^T (1 \times n) \mathbf{H}_{(n \times n)} \mathbf{d}_{(n \times 1)} \]

Subject to
\[ \mathbf{N}^T (p \times n) \mathbf{d}_{(n \times 1)} = \mathbf{e}_{(p \times 1)} \]
\[ \mathbf{A}^T (m \times n) \mathbf{d}_{(n \times 1)} \leq \mathbf{b}_{(m \times 1)} \quad \Rightarrow \quad \mathbf{A}^T (m \times n) \mathbf{d}_{(n \times 1)} - \mathbf{b}_{(m \times 1)} + \mathbf{s}_{(m \times 1)}^2 = 0 \]

Lagrange Function
\[ L = \mathbf{c}^T (1 \times n) \mathbf{d}_{(n \times 1)} + \frac{1}{2} \mathbf{d}^T (1 \times n) \mathbf{H}_{(n \times n)} \mathbf{d}_{(n \times 1)} + \mathbf{u}^T (1 \times m) (\mathbf{A}^T (m \times n) \mathbf{d}_{(n \times 1)} + \mathbf{s}_{(m \times 1)}^2 - \mathbf{b}_{(m \times 1)}) + \mathbf{v}^T (1 \times p) (\mathbf{N}^T (p \times n) \mathbf{d}_{(n \times 1)} - \mathbf{e}_{(p \times 1)}) \]
Solution Procedure of Quadratic Programming (QP) Problem
- Apply the K-T Necessary Condition to the Lagrange Function

**Lagrange Function**

\[
L(d, v, s, u) = c^T_{(1×n)}d_{(n×1)} + \frac{1}{2}d^T_{(1×n)}H_{(n×n)}d_{(n×1)} + u^T_{(1×m)}(A^T_{(m×n)}d_{(n×1)} + s_{(m×1)}^2 - b_{(m×1)}) + v^T_{(1×p)}(N^T_{(p×n)}d_{(n×1)} - e_{(p×1)})
\]

**Kuhn-Tucker Necessary Condition:** \( \nabla L(d, v, s, u) = 0 \)

\[
\frac{\partial L(d, v, s, u)}{\partial d_{(n×1)}} = c_{(n×1)} + H_{(n×n)}d_{(n×1)} + A_{(n×m)}u_{(m×1)} + N_{(n×p)}v_{(p×1)} = 0
\]

\[
\frac{\partial L(d, v, s, u)}{\partial v_{(p×1)}} = N^T_{(p×n)}d_{(n×1)} - e_{(p×1)} = 0
\]

\[
\frac{\partial L(d, v, s, u)}{\partial u_{(n×1)}} = A^T_{(m×n)}d_{(n×1)} + s_{(m×1)}^2 - b_{(m×1)} = 0
\]

\[
\frac{\partial L(d, v, s, u)}{\partial s_i} = u_i s_i = 0, \quad i = 0 \text{ to } m
\]

\[
\frac{\partial L(d, v, s, u)}{\partial s_i} = u_i s_i = 0, \quad i = 0 \text{ to } m
\]
Solution Procedure of Quadratic Programming (QP) Problem
- Method 1: Direct Solving the Eqs. from the K-T Conditions

Optimization problem

\[ \text{Minimize } f(x) = f(x_1, x_2, \cdots, x_n) \]

\[ \text{Subject to } h_i(x) = 0, \quad i = 1, \ldots, p \quad \text{Equality constraint} \]

\[ g_i(x) \leq 0, \quad i = 1, \ldots, m \quad \text{Inequality constraint} \]

Definition of Lagrange function

\[ L(x, v, u, s) = f(x) + \sum_{i=1}^{p} v_i h_i(x) + \sum_{i=1}^{m} u_i (g_i(x) + s_i^2) \]

\( v_i \): Lagrange multiplier for the equality constraint (It is free in sign.)

\( u_i \): Lagrange multiplier for the inequality constraint (Nonnegative)

\( s_i \): Slack variable transforming an inequality constraint to an equality constraint

Kuhn-Tucker necessary condition: \( \nabla L(x, v, u, s) = 0 \)

\[ \frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{p} v_i^* \frac{\partial h_i}{\partial x_j} + \sum_{i=1}^{m} u_i^* \frac{\partial g_i}{\partial x_j} = 0, \quad j = 1, \ldots, n \]

\[ \frac{\partial L}{\partial v_i} = h_i(x^*) = 0, \quad i = 1, \ldots, p \]

\[ \frac{\partial L}{\partial u_i} = g_i(x^*) + s_i^* = 0, \quad i = 1, \ldots, m \]

Linear indeterminate equations

\[ \frac{\partial L}{\partial s_i} = u_i^* s_i^* = 0, \quad i = 1, \ldots, m \]

Nonlinear indeterminate equations

\[ u_i^* \geq 0, \quad i = 1, \ldots, m \]

Method 1:
- Find the solutions which satisfy the nonlinear indeterminate equations.
- Check whether the solutions satisfy the linear indeterminate equations and determine the solution of this problem.
- Human can find the solution of this problem easily by using this method.
Solution Procedure of Quadratic Programming (QP) Problem
- Method 2: Formulate the Problem of the K-T Condition as a LP Problem (1/3)

Kuhn-Tucker Necessary Condition: \( \nabla L(d, v, u, s) = 0 \)

\[
\frac{\partial L}{\partial d} = c_{(n\times1)} + H_{(n\times n)}d_{(n\times1)} + A_{(n\times m)}u_{(m\times1)} + N_{(n\times p)}v_{(p\times1)} = 0, \quad \frac{\partial L}{\partial v} = N^T_{(p\times n)}d_{(n\times1)} - e_{(p\times1)} = 0 \\
\frac{\partial L}{\partial s_i} = u_i \begin{bmatrix} s_i \end{bmatrix} = 0, \quad i = 0 \text{ to } m \quad \text{..... (1)} \quad \frac{\partial L}{\partial u} = A^T_{(m\times n)}d_{(n\times1)} + s_{(m\times1)}^2 - b_{(m\times1)} = 0
\]

Multiply \( s_i \) both side of the Equation (1)

\( u_is_i = 0 \Rightarrow u_is_i^2 = 0 \)

Transform Kuhn-Tucker Necessary Condition: \( \nabla L(d, v, u, s) = 0 \)

\[
\frac{\partial L}{\partial d} = c_{(n\times1)} + H_{(n\times n)}d_{(n\times1)} + A_{(n\times m)}u_{(m\times1)} + N_{(n\times p)}v_{(p\times1)} = 0, \quad \frac{\partial L}{\partial v} = N^T_{(p\times n)}d_{(n\times1)} - e_{(p\times1)} = 0 \\
\frac{\partial L}{\partial s_i} = u_i \begin{bmatrix} s_i \end{bmatrix} = 0, \quad i = 0 \text{ to } m \quad \text{..... (1)} \quad \frac{\partial L}{\partial u} = A^T_{(m\times n)}d_{(n\times1)} + s_{(m\times1)}^2 - b_{(m\times1)} = 0
\]
Solution Procedure of Quadratic Programming (QP) Problem
- Method 2: Formulate the Problem of the K-T Condition as a LP Problem (2/3)

**Kuhn-Tucker Necessary Condition:** \( \nabla L(d, v, u, s) = 0 \)
\[
\frac{\partial L}{\partial d} = c_{(n \times 1)} + H_{(n \times n)} d_{(n \times 1)} + A_{(n \times m)} u_{(m \times 1)} + N_{(n \times p)} v_{(p \times 1)} = 0, \quad \frac{\partial L}{\partial v} = N^T_{(p \times n)} d_{(n \times 1)} - e_{(p \times 1)} = 0
\]
\[
\frac{\partial L}{\partial s_i} = u_i s_i = 0, \quad i = 0 \text{ to } m \quad ①' \quad \frac{\partial L}{\partial u} = A^T_{(m \times n)} d_{(n \times 1)} + s'_{(m \times 1)} - b_{(m \times 1)} = 0
\]
Replace \( s_i^2 \) with \( s_i' \) (where \( s_i' \geq 0 \))

Nonlinear indeterminate equations

Linear indeterminate equations

Check whether the solution obtained from the linear indeterminate equations satisfies the nonlinear indeterminate equations and determine the solution.

Since these equations are linear in the variables \( d, s', u, v \), this problem is a linear programming problem only having the equality constraints.
Solution Procedure of Quadratic Programming (QP) Problem
- Method 2: Formulate the Problem of the K-T Condition as a LP Problem (3/3)

**Kuhn-Tucker Necessary Condition:** $\nabla L(d, v, u, s) = 0$

\[
\frac{\partial L}{\partial d} = c_{(n \times 1)} + H_{(n \times n)}d_{(n \times 1)} + A_{(n \times m)}u_{(m \times 1)} + N_{(n \times p)}v_{(p \times 1)} = 0, \\
\frac{\partial L}{\partial v} = N^T_{(p \times n)}d_{(n \times 1)} - e_{(p \times 1)} = 0
\]

Nonlinear indeterminate equations

Linear indeterminate equations

Check whether the solution obtained from the linear indeterminate equations satisfies the nonlinear indeterminate equations and determine the solution.

Where $u_i, s'_i \geq 0; i = 1 \text{ to } m$

Since the design variables $d_{(n \times 1)}$ are free in sign, we may decompose them as follows to use the Simplex method.

$d_{(n \times 1)} = d^+_{(n \times 1)} - d^-_{(n \times 1)}, \quad (d^+_i \geq 0, \quad d^-_i \geq 0; \quad i = 1 \text{ to } n)$

Also, the Lagrange multipliers $v_{(p \times 1)}$ for the equality constraints are free in sign, we may decompose them as follows to use the Simplex method.

$v_{(p \times 1)} = y_{(p \times 1)} - z_{(p \times 1)}, \quad (y_i \geq 0, \quad z_i \geq 0; \quad i = 1 \text{ to } p)$
Solution Procedure of Quadratic Programming (QP) Problem
- Method 2: Simplex Method for Solving Quadratic Programming Problem (1/2)

Kuhn-Tucker Necessary Condition: \( \nabla L(d, v, u, s) = 0 \)

\[
\frac{\partial L}{\partial d} = c_{(n \times 1)} + H_{(n \times n)}d_{(n \times 1)} + A_{(n \times m)}u_{(m \times 1)} + N_{(n \times p)}v_{(p \times 1)} = 0, \\
\frac{\partial L}{\partial v} = N_{(p \times n)}^Td_{(n \times 1)} - e_{(p \times 1)} = 0, \\
\frac{\partial L}{\partial s_i} = u_is'_i = 0, \quad i = 0 \text{ to } m \\
\frac{\partial L}{\partial u} = A_{(m \times n)}^Td_{(n \times 1)} + s'_{(m \times 1)} - b_{(m \times 1)} = 0.
\]

Nonlinear indeterminate equations

Check whether the solution obtained from the linear indeterminate equations satisfies the nonlinear indeterminate equations and determine the solution.

where \( u_i, s'_i \geq 0; \quad i = 1 \text{ to } m \)

Because \( d \) and \( v \) are free in sign.

\[
d_{(n \times 1)} = d^+_{(n \times 1)} - d^-_{(n \times 1)}, \quad (d^+_i \geq 0, \quad d^-_i \geq 0; \quad i = 1 \text{ to } n) \\
v_{(p \times 1)} = y_{(p \times 1)} - z_{(p \times 1)}, \quad (y_i \geq 0, \quad z_i \geq 0; \quad i = 1 \text{ to } p)
\]

Since these equations are linear in the variables \( d \), \( s' \), \( u \), \( v \), this problem is a linear programming problem only having the equality constraints.

Matrix From

\[
\begin{bmatrix}
H_{(n \times n)} & -H_{(n \times n)} & A_{(n \times m)} & 0_{(n \times m)} & N_{(n \times p)} & -N_{(n \times p)} \\
A^T_{(m \times n)} & -A^T_{(m \times n)} & 0_{(m \times m)} & I_{(m \times m)} & 0_{(m \times p)} & 0_{(m \times p)} \\
N^T_{(p \times n)} & -N^T_{(p \times n)} & 0_{(p \times m)} & 0_{(p \times m)} & 0_{(p \times p)} & 0_{(p \times p)}
\end{bmatrix}
\begin{bmatrix}
d^+_{(n \times 1)} \\
d^-_{(n \times 1)} \\
u_{(m \times 1)} \\
s'_{(m \times 1)} \\
y_{(p \times 1)} \\
z_{(p \times 1)}
\end{bmatrix}
= 
\begin{bmatrix}
-c_{(n \times 1)} \\
b_{(m \times 1)} \\
e_{(p \times 1)}
\end{bmatrix}
\]

Introduce the artificial variables, define the artificial objective function, and solve the linear programming problem by using the Simplex method.
Solution Procedure of Quadratic Programming (QP) Problem
- Method 2: Simplex Method for Solving Quadratic Programming Problem (2/2)

Matrix Form

Introduce the artificial variables, define the artificial objective function, and solve the linear programming problem by using the Simplex method.

\[
\begin{bmatrix}
H_{(n\times n)} & -H_{(n\times n)} & A_{(n\times m)} & 0_{(n\times m)} & N_{(n\times p)} & -N_{(n\times p)} \\
A^T_{(m\times n)} & -A^T_{(m\times n)} & 0_{(m\times m)} & I_{(m\times m)} & 0_{(m\times p)} & 0_{(m\times p)} \\
N^T_{(p\times n)} & -N^T_{(p\times n)} & 0_{(p\times m)} & 0_{(p\times m)} & 0_{(p\times p)} & 0_{(p\times p)} \\
\end{bmatrix}
\begin{bmatrix}
d^+_{(n\times 1)} \\
d^-_{(n\times 1)} \\
u_{(m\times 1)} \\
s'_{(m\times 1)} \\
y_{(p\times 1)} \\
z_{(p\times 1)} \\
\end{bmatrix}
+\begin{bmatrix}
Y_1 \\
Y_2 \\
\vdots \\
Y_{n+m+p} \\
\end{bmatrix}
=\begin{bmatrix}
-c_{(n\times 1)} \\
b_{(m\times 1)} \\
e_{(p\times 1)} \\
\end{bmatrix}
\]

Artificial variables

How to define the artificial objective function

1. Define an one equation by sum of all the equations from the 1st row to (n+m+p)th row.
2. Define the sum of the all artificial variables (Y_1 + Y_2 + \ldots + Y_{n+m+p}) as an objective function (w).

- Determine an initial basic feasible solution (to satisfy the artificial objective function (w) to be zero) for the linear programming problem by using the Simplex method (Phase 1).

- Check whether the initial basic feasible solutions satisfy the following nonlinear indeterminate equations and determine that as a solution.

\[
\frac{\partial L}{\partial s_i} = u_i s'_i = 0, \quad i = 0 \text{ to } m
\]
Solution Procedure of Quadratic Programming (QP) Problem
- Summary of Method 2 of Simplex Method for Solving Quadratic Programming Problem

Kuhn-Tucker Necessary Condition (Matrix form)

\[ B_{((n+m+p) \times (2n+2m+2p))} X_{((2n+2m+2p) \times 1)} = D_{((n+m+p) \times 1)} \]

Simplex Method for Solving Quadratic Programming Problem

1. The problem to solve the Kuhn-Tucker necessary condition is same with the problem having only the equality constraints (linear programming problem).

2. To solve the linear indeterminate equations, we introduce the artificial variables, define the artificial objective function, and determine the initial basic feasible solution by using the Simplex method.

\[ B_{((n+m+p) \times (2n+2m+2p))} X_{((2n+2m+2p) \times 1)} + Y_{((n+m+p) \times 1)} = D_{((n+m+p) \times 1)} \]

If any of the elements in \( D \) is (are) negative, the corresponding equation must be multiplied by -1 to have a nonnegative element on the right side.

3. The artificial objective function is defined as follows.

\[ w = \sum_{i=1}^{n+m+p} Y_i - \sum_{i=1}^{n+m+p} D_i - \sum_{j=1}^{2(n+m+p)} \left( \sum_{i=1}^{n+m+p} B_{ij} X_j \right) = w_0 + \sum_{j=1}^{2(n+m+p)} C_j X_j \]

where \( C_j = - \sum_{i=1}^{n+m+p} B_{ij} \), \( w_0 = \sum_{i=1}^{n+m+p} D_i \) Initial value of the artificial objective function

Add the elements of the \( j \) th column of the matrix \( B \) and change its sign. (Initial relative objective coefficient).

4. Solve the linear programming problem by using the Simplex and check whether the solution satisfies the following equation.

\[ u_i s_i' = 0; \quad i = 1 \text{ to } m \] : This equation is used to check whether the solution satisfies this equation.
**Solution Procedure of Quadratic Programming (QP) Problem**

- Comparison between Method 1 and 2

**Optimization problem**

\[ \text{Minimize } f(x) = f(x_1, x_2, \cdots, x_n) \]

\[ \text{Subject to } h_i(x) = 0, \quad i = 1, \ldots, p \quad \text{Equality constraint} \]

\[ g_i(x) = 0, \quad i = 1, \ldots, m \quad \text{Inequality constraint} \]

**Definition of Lagrange function**

\[ L(x, v, u, s) = f(x) + \sum_{i=1}^{p} v_i h_i(x) + \sum_{i=1}^{m} u_i (g_i(x) + s_i^2) \]

- \( v_i \): Lagrange multiplier for the equality constraint (it is free in sign.)
- \( u_i \): Lagrange multiplier for the inequality constraint (Nonnegative)
- \( s_i \): Slack variable transforming an inequality constraint to an equality constraint

**Kuhn-Tucker necessary condition:** \( \nabla L(x, v, u, s) = 0 \)

\[
\frac{\partial L}{\partial x_j} = \frac{\partial f}{\partial x_j} + \sum_{i=1}^{p} v_i \frac{\partial h_i}{\partial x_j} + \sum_{i=1}^{m} u_i \frac{\partial g_i}{\partial x_j} = 0, \quad j = 1, \ldots, n
\]

\[
\frac{\partial L}{\partial v_i} = h_i(x^*) = 0, \quad i = 1, \ldots, p
\]

\[
\frac{\partial L}{\partial u_i} = g_i(x^*) + s_i^2 = 0, \quad i = 1, \ldots, m
\]

**Linear indeterminate equations**

\[
\frac{\partial L}{\partial s_i} = u_i s_i^* = 0, \quad i = 1, \ldots, m
\]

**Nonlinear indeterminate equations**

\[ u_i^* \geq 0, \quad i = 1, \ldots, m \]

**Method 1:**
- Find the solutions to satisfy the **nonlinear indeterminate equations**.
- Check whether the solutions satisfy the **linear indeterminate equations** and determine the solution of this problem.
- **Human** can find the solution of this problem easily by using this method.

**Method 2:**
- Find the solutions to satisfy the **linear indeterminate equations** by using the Simplex method.
- Check whether the solutions satisfy the **nonlinear indeterminate equations** and determine the solution of this problem.
- Since the algorithm of this method is more systematical, this method is useful for the **computational approach**.
7.2 Sequential Linear Programming (SLP)
Sequential Linear Programming (SLP)

- Define the linear programming (LP) problem by linearizing the objective function and the constraints in the current design point.
- Compute the design change by solving the linear programming problem and obtain the improved design point.

\[
x^{(k+1)} = x^{(k)} + d^{(k)}
\]

This method is to find the optimal solution by solving the linear programming problem **sequentially**.
Sequential Linear Programming (SLP)
- [Example] Problem with Inequality Constraints (1)

Minimize \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

Subject to \( g_1(x) = \frac{1}{6} x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \)
\( g_2(x) = -x_1 \leq 0 \)
\( g_3(x) = -x_1 \leq 0 \)

The starting design point: \( x^{(0)} = (1,1), \)
\( \varepsilon_1 = \varepsilon_2 = 0.001 \)

Choose move limits such that a 15% design change is permissible.

The optimal solution:
\( x^* = (\sqrt{3}, \sqrt{3}), f(x^*) = -3 \)
Sequential Linear Programming (SLP)
- [Example] Problem with Inequality Constraints (2)

Minimize \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

Subject to \( g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0 \)
\( g_2(x) = -x_1 \leq 0 \)
\( g_3(x) = -x_1 \leq 0 \)

(1) Iteration 1 \((k = 0)\)

(i) Step 1

From the given point (starting point), the current design point is as follows.
\( x^{(0)} = (1, 1), \quad \varepsilon_1 = \varepsilon_2 = 0.001 \)

(ii) Step 2: Evaluate the objective and constraint function at the current design point.

\( f(1, 1) = -1 \)
\( g_1(1, 1) = -\frac{2}{3} \leq 0 \quad \Rightarrow \) Constraint is satisfied.
\( g_2(1, 1) = -1 \leq 0 \quad \Rightarrow \) Constraint is satisfied.
\( g_3(1, 1) = -1 \leq 0 \quad \Rightarrow \) Constraint is satisfied.
Sequential Linear Programming (SLP)  
- [Example] Problem with Inequality Constraints (3)

**Minimize** \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

**Subject to** \( g_1(x) = -x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \)
\( g_2(x) = -x_1 \leq 0 \)
\( g_3(x) = -x_2 \leq 0 \)

(1) Iteration 1 \((k = 0)\) \( x^{(0)} = (1, 1), \varepsilon_1 = \varepsilon_2 = 0.001 \)

(iii) Step 3: Define the LP problem (linearize the objective function).

Minimize: \( f(x^{(0)} + \Delta x^{(0)}) \approx f(x^{(0)}) + \nabla f^T (x^{(0)}) \Delta x^{(0)} \)

Minimize: \( f(x^{(0)} + \Delta x^{(0)}) - f(x^{(0)}) \approx \nabla f^T (x^{(0)}) \Delta x^{(0)} \)

Minimize: \( f(x^{(0)} + d^{(0)}) - f(x^{(0)}) \approx \begin{bmatrix} 2x_1 - 3x_2 & 2x_2 - 3x_1 \end{bmatrix}_{x^{(0)}} \begin{bmatrix} d_1^{(0)} \\ d_2^{(0)} \end{bmatrix} \)

\( \bar{f}(d^{(0)}) \approx (2x_1^{(0)} - 3x_2^{(0)})d_1^{(0)} + (2x_2^{(0)} - 3x_1^{(0)})d_2^{(0)} \)

\( f(d^{(0)}) \approx -d_1^{(0)} - d_2^{(0)} \)

The first-order (linear) Taylor series expansion of the objective function

\[ x^{(k+1)} = x^{(k)} + d^{(k)} \]
\[ x^{(k)} = \begin{bmatrix} x_1^{(k)} \\ x_2^{(k)} \end{bmatrix}^T \]
\[ d^{(k)} = \begin{bmatrix} d_1^{(k)} \\ d_2^{(k)} \end{bmatrix}^T = \begin{bmatrix} \Delta x_1^{(k)} \\ \Delta x_2^{(k)} \end{bmatrix}^T \]

The linearized objective function

\[ x^{(0)} = (1, 1) \]
Sequential Linear Programming (SLP)  
- [Example] Problem with Inequality Constraints (4)

**Minimize** \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

**Subject to**  
\( g_1(x) = -x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \)
\( g_2(x) = -x_1 \leq 0 \)
\( g_3(x) = -x_2 \leq 0 \)

---

(1) **Iteration 1** (\( k = 0 \))  
\( x^{(0)} = (1,1), \varepsilon_1 = \varepsilon_2 = 0.001 \)

(iii) **Step 3: Define the LP problem (linearize the constraints).**

**Subject to:**  
\( g_j(x^{(0)} + \Delta x^{(0)}) \Rightarrow g_j(x^{(0)}) + \nabla g_j^T(x^{(0)}) \Delta x^{(0)} \leq 0; j = 1 \text{ to } m \)

\[
\nabla g_j^T(x^{(0)}) \Delta x^{(0)} \leq -g_j(x^{(0)}); j = 1 \text{ to } m
\]

**Subject to:**  
\( \bar{g}_1(d^{(0)}) \Rightarrow \left[ \frac{1}{3} x_1^{(0)} \quad \frac{1}{3} x_2^{(0)} \right] \left[ \begin{array}{c} d_1^{(0)} \\ d_2^{(0)} \end{array} \right] \leq - \left( \frac{1}{6} (x_1^{(0)})^2 + \frac{1}{6} (x_2^{(0)})^2 - 1.0 \right) \)

\( \bar{g}_2(d^{(0)}) \Rightarrow \left[ -x_1^{(0)} \quad 0 \right] \left[ \begin{array}{c} d_1^{(0)} \\ d_2^{(0)} \end{array} \right] \leq -(-x_1^{(0)}) \)

\( \bar{g}_3(d^{(0)}) \Rightarrow \left[ 0 \quad -x_2^{(0)} \right] \left[ \begin{array}{c} d_1^{(0)} \\ d_2^{(0)} \end{array} \right] \leq -(-x_2^{(0)}) \)

Substitute \( x^{(0)} = (1,1) \)

\( g_1(1,1) = -\frac{2}{3} \)
\( g_2(1,1) = -1 \)
\( g_3(1,1) = -1 \)

The linearized constraints

\( g_1(d^{(0)}) = \frac{1}{3} d_1^{(0)} + \frac{1}{3} d_2^{(0)} \leq \frac{2}{3} \)
\( g_2(d^{(0)}) = -d_1^{(0)} \leq 1 \)
\( g_3(d^{(0)}) = -d_2^{(0)} \leq 1 \)
Sequential Linear Programming (SLP) - [Example] Problem with Inequality Constraints (5)

(iv) Step 4: Solve LP problem for the design change \(d(0)\).

Minimize \( \tilde{f} = -d_1 - d_2 \)
Subject to \( \frac{1}{3} d_1 + \frac{1}{3} d_2 \leq \frac{2}{3} \)
\(-d_1 \leq 1\)
\(-d_2 \leq 1\)
\(-0.15 \leq d_1 \leq 0.15\)
\(-0.15 \leq d_2 \leq 0.15\)

limits must be imposed on changes in design called move limit

Linearize the objective function and constraints.

Minimize \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)
Subject to \( g_1(x) = \frac{1}{6} x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \)
\( g_2(x) = -x_1 \leq 0 \)
\( g_3(x) = -x_1 \leq 0 \)

To solve the problem, the Simplex method can be used.
Sequential Linear Programming (SLP)
- [Example] Problem with Inequality Constraints (6)

(v) Step 5: Check for convergence by using the obtained design change $d^{(0)}$.

\[ d^{(0)} = (d_1, d_2) = (0.15, 0.15) \]

Since \( \|d^{(0)}\| = \sqrt{0.15^2 + 0.15^2} = 0.212 > \varepsilon_2 (= 0.001) \), the criterion for convergence is not satisfied.

(vi) Step 6: Update the design point as \( x^{(k+1)} = x + d^{(k)} \). Set \( k = k+1 \) and go to Step 2.

\[ x^{(1)} = x^{(1,1)} = x^{(0)} + d^{(0)} = (1, 1) + (0.15, 0.15) = (1.15, 1.15) \]

\[ k = k + 1 = 1 \]
Summary of Sequential Linear Programming (SLP)

Minimize \( f(x^{(k)} + \Delta x^{(k)}) \approx f(x^{(k)}) + \nabla f^T(x^{(k)})\Delta x^{(k)} \)

Subject to \( h_j(x^{(k)} + \Delta x^{(k)}) \approx h_j(x^{(k)}) + \nabla h_j^T(x^{(k)})\Delta x^{(k)} = 0; j = 1 \text{ to } p \)

\( g_j(x^{(k)} + \Delta x^{(k)}) \approx g_j(x^{(k)}) + \nabla g_j^T(x^{(k)})\Delta x^{(k)} \leq 0; j = 1 \text{ to } m \)

Define:
\[
\bar{f} = f(x + \Delta x) - f(x), \quad e_j = -h_j(x), \quad b_j = -g_j(x), \\
c_i = \frac{\partial f(x)}{\partial x_i}, \quad n_{ij} = \frac{\partial h_j(x)}{\partial x_i}, \quad a_{ij} = \frac{\partial g_j(x)}{\partial x_i}, \\
d_i = \Delta x_i
\]

Minimize \( \bar{f} = \sum_{i=1}^{n} c_i d_i \)

Subject to \( \sum_{i=1}^{n} n_{ij} d_i = e_j; j = 1 \text{ to } p \)

\( \sum_{i=1}^{n} a_{ij} d_i \leq b_j; j = 1 \text{ to } m \)

where \( d_{il} \leq d_i \leq d_{iu} (\Delta x_{il}^{(k)} \leq \Delta x_i^{(k)} \leq \Delta x_{iu}^{(k)}) \)

Subject to \( N^T_{(p \times n)}d_{(n \times 1)} = e_{(p \times 1)} \)

\( A^T_{(m \times n)}d_{(n \times 1)} \leq b_{(m \times 1)} \)

Matrix form

Minimize \( \bar{f} = c^T_{(1 \times n)}d_{(n \times 1)} \)

Linear Programming Problem

It can be solved by using the Simplex method.
Summary of the SLP Algorithm (1)

- **Step 1:** Estimate a starting design point as \( x^{(0)} \). Set \( k = 0 \). Specify two small numbers, \( \varepsilon_1, \varepsilon_2 \) (criterion for violating the constraints and convergence).

- **Step 2:** Evaluate objective and constraint function at current design point \( x^{(k)} \). Also evaluate the objective and constraint function gradients at the current design point.

- **Step 3:** Select the proper move limits \( \Delta x_{il}^{(k)} \) and \( \Delta x_{iu}^{(k)} \) as some fraction of the current design point. Define the linear programming problem.

\[
\Delta x_{il}^{(k)} \leq \Delta x_{i}^{(k)} \leq \Delta x_{iu}^{(k)}
\]
Summary of the SLP Algorithm (2)

- Step 4: Solve the linear programming problem for $d^{(k)}$ by using the Simplex method.

- Step 5: Check for convergence. If, $g_i \leq \varepsilon_1 (i = 1 \text{ to } m)$, $|h_i| \leq \varepsilon_1 (i = 1 \text{ to } p)$, and $|d^{(k)}| \leq \varepsilon_2$, then stop and the current design point $x^{(k)}$ is the optimal solution. Otherwise, continue.

- Step 6: Update the design point as $x^{(k+1)} = x + \Delta x^{(k)}$, Set $k = k+1$ and go to Step 2.
Limitations of SLP Method

- The move limits of the design variables should be defined by the user.
- If the move limits are too small, it take much time to find the optimal solution.
- If the move limits are too large, it can cause oscillations in the design point during iterations.
- Thus performance of the method depends heavily on selection of move limits.

\[ f(x) = f(x^{(n)}) + \nabla f(x^{(n)})^T (x - x^{(n)}) + \frac{1}{2} (x - x^{(n)})^T \nabla^2 f(x^{(n)}) (x - x^{(n)}) \]

Original objective function

Linearized objective function

\[ f(x^{(n+1)}) = f(x^{(n)}) + \nabla f(x^{(n)})^T (x^{(n+1)} - x^{(n)}) + \frac{1}{2} (x^{(n+1)} - x^{(n)})^T \nabla^2 f(x^{(n)}) (x^{(n+1)} - x^{(n)}) \]

The optimal solution cannot be obtained, because of the oscillations in the design point during iterations.
7.3 Sequential Quadratic Programming (SQP)
Sequential Quadratic Programming (SQP)

- Define the Quadratic programming (QP) problem by approximating quadratic form of the objective function and linear form of the constraints at the current design point.

- Compute the design change by solving the quadratic programming problem and obtain the improved design point.

\[
\mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \mathbf{d}^{(k)}
\]

- This method is to find the optimal solution by solving the Quadratic programming problem **sequentially**.
**Formulation of the Quadratic Programming Problem to Determine the Search Direction**

\[ \text{Minimize} \quad f(x + \Delta x) \cong f(x) + \nabla f^T(x)\Delta x + 0.5\Delta x^T H \Delta x \]

The second-order Taylor series expansion of the objective function

\[ \text{Subject to} \quad h_j(x + \Delta x) \cong h_j(x) + \nabla h_j^T(x)\Delta x = 0; \quad j = 1 \text{to } p \]

The first-order (linear) Taylor series expansion of the equality constraints

\[ g_j(x + \Delta x) \cong g_j(x) + \nabla g_j^T(x)\Delta x \leq 0; \quad j = 1 \text{to } m \]

The first-order (linear) Taylor series expansion of the inequality constraints

Define:

\[ \bar{f} = f(x + \Delta x) - f(x), \quad e_j = -h_j(x), \quad b_j = -g_j(x), \]

\[ c_i = \frac{\partial f(x)}{\partial x_i}, \quad n_{ij} = \frac{\partial h_j(x)}{\partial x_i}, \quad a_{ij} = \frac{\partial g_j(x)}{\partial x_i}, \]

\[ d_i = \Delta x_i \]

**Matrix form**

\[ \text{Minimize} \quad \bar{f} = c^T_{(1 \times n)}d_{(n \times 1)} + \frac{1}{2}d^T_{(1 \times n)}H_{(n \times n)}d_{(n \times 1)} \quad : \text{Quadratic objective function} \]

\[ \text{Subject to} \quad N^T_{(p \times n)}d_{(n \times 1)} = e_{(p \times 1)} \quad : \text{Linear equality constraints} \]

\[ A^T_{(m \times n)}d_{(n \times 1)} \leq b_{(m \times 1)} \quad : \text{Linear inequality constraints} \]
Algorithm of SQP

**Step 1**
Define the quadratic programming problem at the current point.

**Step 2**
Calculate the search direction $d^{(0)}$ by solving the quadratic programming problem.

**Step 3**
Transform the constrained optimal design problem to the unconstrained problem by modifying the objective function which has added penalty for possible constraint violations to the current value of the objective function. Then calculate the step size using the one dimensional search method, e.g., Golden section search method.

Stopping criteria
If the magnitude of the search direction $|d^{(0)}|$ is smaller than a small value (epsilon), then stop.

Objective function is approximated to the quadratic form.

Quadratic programming problem
- Objective function: quadratic form
- Constraint: linear form
Difference between Sequential Quadratic Programming (SQP) and CSD (Constrained Steepest Descent) Method

☑ Sequential Quadratic Programming (SQP)

1. First, we define a quadratic programming problem for the objective function and constraints at the current design point, and find the search direction \( d^{(k)} \).

2. We define the penalty function by adding a penalty for possible constraint violations to the current value of the objective function, and calculate the step size \( \alpha_k \) to minimize the penalty function. For determination of the step size, one dimensional search method, e.g., Golden section search method can be used. And we determine the improved design point.

3. At the improved design point, we go to 1.

The method is to find the optimal solution by solving the quadratic programming problem sequentially.

☑ CSD (Constrained Steepest Descent) method

- This method is a kind of the SQP method.
- When defining the quadratic programming problem, the Hessian matrix is assumed to be equal to the identity Matrix.
- This method uses the Pshenichny’s penalty function.
Define the QP problem

To find the search direction $d^{(0)}$, we define the QP problem at current design point.

Minimize: $f(x^{(0)} + \Delta x^{(0)}) \approx f(x^{(0)}) + \nabla f^T(x^{(0)})\Delta x^{(0)} + 0.5\Delta x^{(0)^T}H\Delta x^{(0)}$

The second-order Taylor series expansion of the objective function

Minimize: $f(x^{(0)} + \Delta x^{(0)}) - f(x^{(0)}) \approx \nabla f^T(x^{(0)})\Delta x^{(0)} + 0.5\Delta x^{(0)^T}H\Delta x^{(0)}$

$\Delta x^{(0)} = d^{(0)}, \nabla f^T = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}, H = I$ (In the CSD method, the Hessian matrix is assumed to be equal to the identity matrix.)

Minimize: $f(x^{(0)} + d^{(0)}) - f(x^{(0)}) \approx \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}_{x^{(0)}} \begin{bmatrix} d_1^{(0)} \\ d_2^{(0)} \end{bmatrix} + 0.5(d_1^{(0)^2} + d_2^{(0)^2})$

$\bar{f}(d^{(0)}) \approx \frac{\partial f(x^{(0)})}{\partial x_1}d_1^{(0)} + \frac{\partial f(x^{(0)})}{\partial x_2}d_2^{(0)} + 0.5(d_1^{(0)^2} + d_2^{(0)^2})$

It has the same form of the equation of circle.

Form of the equation of circle: $x_1^2 + x_2^2 + c_1x_1 + c_2x_2 + c_3 = 0$
Solution Procedure of SQP Using the Example
- Determination of the Search Direction (1/5) [Iteration 1]

Minimize \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

Subject to \( g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0 \)
\( g_2(x) = -x_1 \leq 0 \)
\( g_3(x) = -x_2 \leq 0 \)

Assume the starting point is \( x^{(0)} = (1, 1) \).

(1) Iteration 1\((k = 0)\)

(i) Step 1: Evaluate the objective function and constraints at the current design point.

\( f(1, 1) = -1 \)
\( g_1(1, 1) = -\frac{2}{3} < 0 \quad \Rightarrow \text{Constraint is satisfied.} \)
\( g_2(1, 1) = -1 < 0 \quad \Rightarrow \text{Constraint is satisfied.} \)
\( g_3(1, 1) = -1 < 0 \quad \Rightarrow \text{Constraint is satisfied.} \)
Solution Procedure of SQP Using the Example - Determination of the Search Direction (2/5)

Minimize \[ f(x) = x_1^2 + x_2^2 - 3x_1x_2 \]

Subject to \[ g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0 \]
\[ g_2(x) = -x_1 \leq 0 \]
\[ g_3(x) = -x_2 \leq 0 \]

(1) Iteration 1 \((k = 0)\) \(x^{(0)} = (1, 1)\)

(ii) Step 2: Define a QP problem (The objective function is approximated to the quadratic form.)

\[
\text{Minimize:} \quad f(x^{(0)} + \Delta x^{(0)}) \approx f(x^{(0)}) + \nabla f^T(x^{(0)})\Delta x^{(0)} + 0.5\Delta x^{(0)T}H\Delta x^{(0)}
\]

\[
\text{Minimize:} \quad f(x^{(0)} + \Delta x^{(0)}) - f(x^{(0)}) \approx \nabla f^T(x^{(0)})\Delta x^{(0)} + 0.5\Delta x^{(0)T}H\Delta x^{(0)}
\]

\[
\Delta x^{(0)} = d^{(0)}, \quad \nabla f^T = \begin{bmatrix} \frac{\partial f}{\partial x_1} & \frac{\partial f}{\partial x_2} \end{bmatrix}, \quad H = I
\]

\[
\text{Minimize:} \quad f(x^{(0)} + d^{(0)}) - f(x^{(0)}) \approx \begin{bmatrix} 2x_1 - 3x_2 & 2x_2 - 3x_1 \end{bmatrix}_x^{(0)} \begin{bmatrix} d_1^{(0)} \\ d_2^{(0)} \end{bmatrix} + 0.5(d_1^{(0)2} + d_2^{(0)2})
\]

\[
\bar{f}(d^{(0)}) \approx (2x_1^{(0)} - 3x_2^{(0)})d_1^{(0)} + (2x_2^{(0)} - 3x_1^{(0)})d_2^{(0)} + 0.5(d_1^{(0)2} + d_2^{(0)2})
\]

Objective function is approximated to the first order term.

\[
\bar{f}(d^{(0)}) \approx -d_1^{(0)} - d_2^{(0)} + 0.5(d_1^{(0)2} + d_2^{(0)2})
\]

Objective function is approximated to the second order term.
Solution Procedure of SQP Using the Example - Determination of the Search Direction (3/5)

Minimize  \[ f(x) = x_1^2 + x_2^2 - 3x_1x_2 \]

Subject to  \[ g_1(x) = \frac{1}{6} x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \]
   \[ g_2(x) = -x_1 \leq 0 \]
   \[ g_3(x) = -x_2 \leq 0 \]

(1) Iteration 1 \((k = 0)\)  \[ x^{(0)} = (1, 1) \]

Subject to: \(g_j(x^{(0)} + \Delta x^{(0)}) \leq g_j(x^{(0)}) + \nabla g_j^T(x^{(0)})\Delta x^{(0)} \leq 0; \ j = 1 \) to \(m\)

\[ \nabla g_j^T(x^{(0)})\Delta x^{(0)} \leq -g_j(x^{(0)}); \ j = 1 \) to \(m\)

\[ \Delta x^{(0)} = d^{(0)}, \nabla g_j^T = \left[ \frac{\partial g_j}{\partial x_1} \frac{\partial g_j}{\partial x_2} \right] \]

Subject to:

\[ g_1(d^{(0)}) = \left[ \frac{1}{3} x_1^{(0)} \frac{1}{3} x_2^{(0)} \right] \left[ \begin{array}{c} d_1^{(0)} \\ d_2^{(0)} \end{array} \right] \leq -\left( \frac{1}{6} (x_1^{(0)})^2 + \frac{1}{6} (x_2^{(0)})^2 - 1.0 \right) \]

\[ g_2(d^{(0)}) = \left[ -x_1^{(0)} 0 \right] \left[ \begin{array}{c} d_1^{(0)} \\ d_2^{(0)} \end{array} \right] \leq -(-x_1^{(0)}) \]

\[ g_3(d^{(0)}) = \left[ 0 -x_2^{(0)} \right] \left[ \begin{array}{c} d_1^{(0)} \\ d_2^{(0)} \end{array} \right] \leq -(-x_2^{(0)}) \]

The constraints are linearized

\[ \frac{1}{3} d_1^{(0)} + \frac{1}{3} d_2^{(0)} \leq \frac{2}{3} \]

\[ -d_1^{(0)} \leq 1 \]

\[ -d_2^{(0)} \leq 1 \]

Substitute \( x^{(0)} = (1, 1) \)
(iii) Step 3: Solve the QP problem to find the search direction \(d^{(0)}\).

### Constrained Optimal Design Problem (Original problem)

**Minimize** \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

**Subject to**

\[
g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0
\]

\[
g_2(x) = -x_1 \leq 0
\]

\[
g_3(x) = -x_2 \leq 0
\]

### Quadratic Programming Problem

**Minimize**

\[
\bar{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)
\]

**Subject to**

\[
\frac{1}{3}d_1 + \frac{1}{3}d_2 \leq \frac{2}{3}
\]

\[-d_1 \leq 1
\]

\[-d_2 \leq 1
\]

Kuhn-Tucker necessary condition: \(\nabla L(d, u, s) = 0\)

- \(\frac{\partial L}{\partial d_1} = -1 + d_1 + \frac{1}{3}u_1 - u_2 = 0\)
- \(\frac{\partial L}{\partial d_2} = -1 + d_2 + \frac{1}{3}u_1 - u_3 = 0\)
- \(\frac{\partial L}{\partial u_1} = \frac{1}{3}(d_1 + d_2 - 2) + s_1^2 = 0\)
- \(\frac{\partial L}{\partial u_2} = -d_1 - 1 + s_2^2 = 0\)
- \(\frac{\partial L}{\partial u_3} = -d_2 - 1 + s_3^2 = 0\)
- \(\frac{\partial L}{\partial s_i} = u_is_i = 0, \; u \geq 0, \; i = 1, 2, 3\)

The search direction is

\[
(0, 0, 0), \quad (0, 1.414, 1.414), \quad (1, 1)
\]

*The search direction also can be determined using the Simplex method.*
Solution Procedure of SQP Using the Example - Determination of the Search Direction (5/5)

\[\mathbf{d}^{(0)} = (d_1, d_2) = (1, 1)\]

The search direction is determined.

(iv) Step 4: After the search direction \((d^{(0)})\) is determined, calculate the step size.

Calculate step size minimizing the value of the objective function along the search direction

\[
\mathbf{X}^{(1)} = \mathbf{X}^{(0)} + \alpha_0 \mathbf{d}^{(0)}
\]

Improved design point - Current design point - Search direction obtained from the QP problem

Find \(\alpha_k\):

\[
\text{Minimize } f(\mathbf{x}^{(1)}) = f(\mathbf{x}^{(0)} + \alpha_0 \mathbf{d}^{(0)}) = f(\alpha_0)
\]

The improved design point can be found along the search direction by minimizing the objective function. However, it may violate the original constraints.

Therefore, a penalty function should be constructed by adding the penalty for possible constraint violations to the current value of the objective function.

By property of the nature, the objective function is decreased when the constraints is violated.
Solution Procedure of SQP Using the Example
- Definition of **Penalty Function** (Pshenichny’s Descent Function) (1/2)

**Penalty function** (Pshenichny’s descent function, \( \Phi(x^{(k)}) \))

By adding a penalty for possible constraint violations to the current value of the objective function, the constrained optimal design problem is transformed into the unconstrained optimal design problem:

\[
\Phi(x^{(k)}) = f(x^{(k)}) + R_k \cdot V(x^{(k)})
\]

where,

- \( k \): iteration number how many times the QP problem is defined approximately
- \( f(x^{(k)}) \): current \((k^{th} \text{ iteration})\) value of the objective function
- \( V(x^{(k)}) \) is either the maximum constraint violation among all the constraints or zero.
- \( V(x^{(k)}) \) is nonnegative. If all the constraints are satisfied, the value of the \( V(x^{(k)}) \) is zero.

\[
V(x^{(k)}) = \max \{0; |h_1|, |h_2|, \ldots, |h_p|, g_1, g_2, \ldots, g_m\}
\]

where,

- \( h_p \): value of the equality constraint function at the design point \( x^{(k)} \)
- \( g_p \): value of the inequality constraint function at the design point \( x^{(k)} \)

\( R_k \) is a positive number called the penalty parameter.

\[
R_k = \max \{R_0, r_k\}
\]

Initial value of \( R_k \) is specified by the user.

Summation of all the Lagrange multipliers

\[
r_k = \sum_{i=1}^{p} |\lambda_i^{(k)}| + \sum_{i=1}^{m} u_i^{(k)}
\]

\( \lambda_i^{(k)} \): Lagrange multipliers for the equality constraints (free in sign)

\( u_i^{(k)} \): Lagrange multiplier for the inequality constraints (nonnegative)
Solution Procedure of SQP Using the Example
- Definition of Penalty Function (Pshenichny’s Descent Function) (2/2)

\( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

\( g_1(x) = \frac{1}{6}x_1^3 + \frac{1}{6}x_2^3 - 1.0 \leq 0 \)

\( g_2(x) = -x_1 \leq 0 \)

\( g_3(x) = -x_2 \leq 0 \)

(v) Step 5: Calculate the penalty parameter \( R_k \) (In this example, the initial penalty parameter is assumed as \( R_0 = 10 \)).

\[
\mathbf{u}^{(0)} = (u_1, u_2, u_3) = (0, 0, 0) \quad \text{and} \quad r_k = \sum_{i=1}^{p} v_i^{(k)} + \sum_{i=1}^{m} u_i^{(k)}
\]

\[
r_0 = \sum_{i=1}^{m} u_i^{(0)} = 0
\]

Therefore, \( R_0 = \max \{ R_0, r_0 \} = \max \{ 10, 0 \} = 10 \)

\[
\Phi(x^{(k)}) = f(x^{(k)}) + R_k \cdot V(x^{(k)})
\]

\[
= x_1^2 + x_2^2 - 3x_1x_2 + 10 \cdot V(x^{(k)}), \quad V(x^{(k)}) = \max \{ 0, g_1(x^{(k)}), g_2(x^{(k)}), g_3(x^{(k)}) \}, \quad (k=0)
\]

Since this problem does not have the equality constraints, we do not consider the \( v_i \).
Solution Procedure of SQP Using the Example
- Determination of the Step Size

(vi) Step 6:
By using the one dimensional search method, e.g., Golden section search method, calculate the step size to minimize the penalty function along the search direction \(d^{(0)}\), and determine the improved design point.

\[
\Phi(x^{(k)}) = f(x^{(k)}) + R_k \cdot V(x^{(k)})
\]
\[
= x_1^2 + x_2^2 - 3x_1x_2 + 10 \cdot V(x^{(k)})
\]
\[
V(x^{(k)}) = \max\{0, g_1(x^{(k)}), g_2(x^{(k)}), g_3(x^{(k)})\}, \quad (k=0)
\]

After the k-th search direction is found, one dimensional search for the step size is started.

\[
x^{(k,j)} = x^{(k)} + \alpha^{(k,j)} d^{(k)}
\]

The iteration number \(k\) does not change during the one dimensional search for the step size.

\[
\Phi(x^{(k,j)}) = f(x^{(k,j)}) + R_k \cdot V(x^{(k,j)}) \quad \text{and} \quad V(x^{(k,j)}) = \max\{0, g_1(x^{(k,j)}), g_2(x^{(k,j)}), g_3(x^{(k,j)})\}
\]

The iteration number \(k\) does not change during the one dimensional search method.

\[
f(x) = x_1^2 + x_2^2 - 3x_1x_2
\]
\[
g_1(x) = -x_1^2 + \frac{1}{x_2} - 1.0 \leq 0
\]
\[
g_2(x) = -x_1 \leq 0
\]
\[
g_3(x) = -x_2 \leq 0
\]

After completing the one dimensional search, \(k\) is changed to \(k+1\):

\[x^{(k,j)}\text{ is changed to } x^{(k+1)}\]
Solution Procedure of SQP Using the Example
- Determination of the Step Size Using the Golden Section Search Method (1/6)

(vi) Step 6:

Search direction: $d_0 = (1,1)$, $k = 0$, $j = 0$

When $\alpha_{(0,j)} = 0.0$

$x^{(0,j)} = x^{(0)} + \alpha_{(0,j)} \cdot d^{(0)} = (1,1) + 0 \cdot (1,1) = (1,1)$

$\Phi(x^{(0,j)}) = f(x^{(0,j)}) + R_0 \cdot V(x^{(0,j)}) = -1 + 10 \times 0 = -1$

where, $V(x^{(0,j)}) = \max \{ 0, g_1(x^{(0,j)}), g_2(x^{(0,j)}), g_3(x^{(0,j)}) \}$

$= \max \{ 0, -\frac{2}{3}, -1, -1 \} = 0$
Solution Procedure of SQP Using the Example
- Determination of the Step Size Using the Golden Section Search Method (2/6)

(vi) Step 6:

Search direction: \( d_0 = (1,1) \), \( k = 0 \), \( j = 1 \)

Assume \( \alpha_{(0,j)} = 0.1^* \),

\[
\begin{align*}
  x^{(0,j)} &= x^{(0)} + \alpha_{(0,j)} \cdot d^{(0)} = (1,1) + 0.1 \cdot (1,1) = (1.1, 1.1) \\
  \Phi\left( x^{(0,j)} \right) &= f\left( x^{(0,j)} \right) + R_0 \cdot V\left( x^{(0,j)} \right) = -1.21 + 10 \times 0 = -1.21 \\
  \text{where, } V\left( x^{(0,j)} \right) &= \max \{0, g_1(x^{(0,j)}), g_2(x^{(0,j)}), g_3(x^{(0,j)})\} \\
  &= \max \{0, -0.57, -1.1, -1.1\} = 0
\end{align*}
\]

* The initial value of \( \alpha_{(k,j)} (0.1) \) is given by the user. It can be given as another value, e.g., 0.5.
Solution Procedure of SQP Using the Example
-Determination of the Step Size Using the Golden Section Search Method (3/6)

(vi) Step 6:

Search direction: $\mathbf{d}_0 = (1,1)$, $k = 0$, $j = 2$

When $\alpha_{(0,j)} = 0.1 + 1.618(0.1) = 0.2618$

$$\mathbf{x}^{(0,j)} = \mathbf{x}^{(0)} + \alpha_{(0,j)} \cdot \mathbf{d}^{(0)} = (1,1) + 0.262 \cdot (1,1) = (1.262,1.262)$$

$$\Phi\left(\mathbf{x}^{(0,j)}\right) = f\left(\mathbf{x}^{(0,j)}\right) + R_0 \cdot V\left(\mathbf{x}^{(0,j)}\right) = -1.592 + 10 \times 0 = -1.592$$

where, $V\left(\mathbf{x}^{(0,2)}\right) = \max\{0, g_1(\mathbf{x}^{(0,2)}), g_2(\mathbf{x}^{(0,2)}), g_3(\mathbf{x}^{(0,2)})\}$

$$= \max\{0, -0.469, -1.262, -1.262\} = 0$$
Solution Procedure of SQP Using the Example
- Determination of the Step Size Using the Golden Section Search Method (4/6)

(vi) Step 6:

Search direction: \( d_0 = (1,1), \quad k = 0, \quad j = 3 \)

When \( \alpha_{(0,j)} = 0.1 + 1.618(0.1) + 1.618^2 (0.1) = 0.5236 \)

\[
\mathbf{x}^{(0,j)} = \mathbf{x}^{(0)} + \alpha_{(0,j)} \cdot \mathbf{d}^{(0)} = (1,1) + 0.524 \cdot (1,1) = (1.524,1.524)
\]

\[
\Phi \left( \mathbf{x}^{(0,j)} \right) = f \left( \mathbf{x}^{(0,j)} \right) + R_0 \cdot V \left( \mathbf{x}^{(0,j)} \right) = -2.321 + 10 \times 0 = -2.321
\]

where, \( V (\mathbf{x}^{(0,j)}) = \max \{0, g_1 (\mathbf{x}^{(0,j)}), g_2 (\mathbf{x}^{(0,j)}), g_3 (\mathbf{x}^{(0,j)})\} \)

\[
= \max \{0, -0.226, -1.524, -1.524\} = 0
\]
Solution Procedure of SQP Using the Example
- Determination of the Step Size Using
the Golden Section Search Method (5/6)

(vi) Step 6:

Search direction: \( d_0 = (1,1), \ k = 0, j = 4 \)

When \( \alpha_{(0,j)} = 0.1 + 1.618(0.1) + 1.618^2 (0.1) + 1.618^3 (0.1) = 0.9472 \)
\( x^{(0,j)} = x^{(0)} + \alpha_{(0,j)} \cdot d^{(0)} = (1,1) + 0.947 \cdot (1,1) = (1.947,1.947) \)
\( \Phi(x^{(0,j)}) = f(x^{(0,j)}) + R_0 \cdot V(x^{(0,j)}) = -3.792 + 10 \times 0.2638 = -1.154 \)

where, \( V(x^{(0,4)}) = \max \{0, g_1(x^{(0,4)}), g_2(x^{(0,4)}), g_3(x^{(0,4)})\} \)
\[ = \max \{0, 0.2638, -1.947, -1.947\} = 0.2638 \]
(vi) Step 6:

The value of the $\alpha_0 = 0.732$ is found at which the penalty function is minimized in the interval between $x^{(0,2)}$ and $x^{(0,4)}$.

$x^{(1)} = x^{(0)} + \alpha_0 \cdot d^{(0)} = (1,1) + 0.732 \cdot (1,1) = (1.732, 1.732)$

$f(x^{(1)}) = f(1.732, 1.732) = -3$
Solution Procedure of SQP Using the Example
- Determination of the Search Direction (1/3) [Iteration 2]

(2) Iteration 2\((k = 1)\)

(i) Step 1: Calculate maximum constraint violation of all the constraints.

From the previous stage,
\[ x^{(1)} = (1.732, 1.732) \]

\[ f\left(x^{(1)}\right) = f\left(1.732, 1.732\right) = -2.999824 \]

\[ g_1\left(x^{(1)}\right) = g_1\left(1.732, 1.732\right) = -5.866 \times 10^{-5} \]  \(\Rightarrow\) Constraint is satisfied.

\[ g_2\left(x^{(1)}\right) = -1.732 \]  \(\Rightarrow\) Constraint is satisfied.

\[ g_3\left(x^{(1)}\right) = -1.732 \]  \(\Rightarrow\) Constraint is satisfied.

\[ V_1 = V\left(x^{(1)}\right) = \max\{0; -5.866 \times 10^{-5}, -1.732, -1.732\} = 0 \]

And,

\[ \nabla f\left(x^{(1)}\right) = (2x_1 - 3x_2, 2x_2 - 3x_1) = (-1.732, -1.732) \]

\[ \nabla g_1\left(x^{(1)}\right) = \left(\frac{1}{3}x_1, \frac{1}{3}x_2\right) = (0.577, 0.577), \nabla g_2 = (-1, 0), \nabla g_3 = (0, -1) \]
Solution Procedure of SQP Using the Example
- Determination of the Search Direction (2/3)

(ii&iii) Step 2&3: Define and solve the QP problem to determine the search direction \(d^{(1)}\).

### Constrained Optimal Design Problem
(Original problem)

**Minimize** \(f(x) = x_1^2 + x_2^2 - 3x_1x_2\)

*Subject to*
- \(g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0\)
- \(g_2(x) = -x_1 \leq 0\)
- \(g_3(x) = -x_1 \leq 0\)

### Quadratic Programming Problem

**Minimize** \(\tilde{f} = (-1.732d_1 - 1.732d_2) + 0.5(d_1^2 + d_2^2)\)

*Subject to* \(0.577d_1 + 0.577d_2 \leq 5.866 \times 10^{-5}\)

- \(-d_1 \leq 1.732\)
- \(-d_2 \leq 1.732\)

where,
- \(d_1 = x_1 - 1.732\)
- \(d_2 = x_2 - 1.732\)

### Lagrange function

\[ L = (-1.732d_1 - 1.732d_2) + 0.5(d_1^2 + d_2^2) + u_1[0.577(d_1 + d_2) - 5.866 \times 10^{-5} + s_1^2] + u_2(-d_1 - 1.732 + s_2^2) + u_3(-d_2 - 1.732 + s_3^2) \]

### Kuhn-Tucker necessary condition:

\[ \nabla L(d, u, s) = 0 \]

### The search direction is

\[
\begin{align*}
\frac{\partial L}{\partial d_1} &= -1.732 + d_1 + 0.577u_1 - u_2 = 0 \\
\frac{\partial L}{\partial d_2} &= -1.732 + d_2 + 0.577u_1 - u_3 = 0 \\
\frac{\partial L}{\partial u_1} &= 0.577(d_1 + d_2) - 5.866 \times 10^{-6} + s_1^2 = 0 \\
\frac{\partial L}{\partial u_2} &= -d_1 - 1.732 + s_2^2 = 0 \\
\frac{\partial L}{\partial u_3} &= -d_2 - 1.732 + s_3^2 = 0 \\
\frac{\partial L}{\partial s_i} &= u_is_i = 0, \ u \geq 0, \ i = 1, 2, 3
\end{align*}
\]

\[
\begin{align*}
\mathbf{d}^{(1)} &= (d_1, d_2) \\
&= (5.081 \times 10^{-5}, 5.081 \times 10^{-5}) \\
\mathbf{u}^{(1)} &= (u_1, u_2, u_3) \\
&= (3, 0, 0) \\
\mathbf{s}^{(1)} &= (s_1, s_2, s_3) \\
&= (0, 1.316, 1.316)
\end{align*}
\]

* The search direction also can be determined using the Simplex method.
(iv) Step 4: Check for the following stopping criteria.

\[ d^{(1)} = (d_1, d_2) = (5.081 \times 10^{-5}, 5.081 \times 10^{-5}) \]

\[ \left\| d^{(1)} \right\| = \sqrt{(5.081 \times 10^{-5})^2 + (5.081 \times 10^{-5})^2} = 7.186 \times 10^{-5} < \varepsilon_2 = 0.001 \]

The stopping criteria is satisfied.

(v) Step 5: Stop the iteration.

The candidate minimum solution: \[ x^* = (\sqrt{3}, \sqrt{3}), f(x^*) = -3 \]

where the Lagrange multiplier are:

\[ u^* = (3, 0, 0), s^* = (0, 1.316, 1.316) \]
Summary of Sequential Quadratic Programming (SQP)

Optimization Problem

Minimize \( f(x) = f(x_1, x_2, \cdots, x_n) \)

Subject to \( h_i(x) = 0, \quad i = 1, \ldots, p \) Equality constraints

\( g_i(x) = 0, \quad i = 1, \ldots, m \) Inequality constraints

Pshenichny’s descent function: the penalty function is constructed by adding a penalty for possible constraint violations to the current value of the objective function

\[
\Phi(x^{(k)}) = f(x^{(k)}) + R_k \cdot V(x^{(k)})
\]

\((k\) is the iteration number how many times the QP problem is defined.\)

\(V(x^{(k)})\) is either the maximum constraint violation of all the constraints or zero.

\(V(x^{(k)})\) is nonnegative. If all the constraints are satisfied, the value of the \(V(x^{(k)})\) is zero.

\[
V(x^{(k)}) = \max \{0; |h_1|, |h_2|, \cdots, |h_p|; g_1, g_2, \cdots, g_m\}
\]

\(R_k\) is a positive number called the penalty parameter (initially specified by the user).

\[
R_k = \max \left\{ R_0, \quad r_k = \left( \sum_{i=1}^{p} \left| v_i^{(k)} \right| + \sum_{i=1}^{m} u_i^{(k)} \right) \right\}
\]

\(: Summation of all the Lagrange multipliers\)

The improved design point is determined as follows:

\[
x^{(k+1)} = x^{(k)} + \alpha_k \cdot d^{(k)}
\]

Improved design point \( \quad \) Current design point \( \quad \) Search direction obtained from the QP problem \( \quad \) Step size calculated by one dimensional search method (ex. Golden section search method)
7.4 Determine the Search Direction of the Quadratic Programming Problem by Using the Simplex Method
Formulation of the Quadratic Programming Problem (1/5)

Solve the QP problem to determine the search direction ($d^{(0)}$)

<table>
<thead>
<tr>
<th>Constrained Optimal Design Problem (Original problem)</th>
<th>Quadratic Programming Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Minimize</strong> $f(x) = x_1^2 + x_2^2 - 3x_1x_2$</td>
<td><strong>Minimize</strong> $\tilde{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)$</td>
</tr>
<tr>
<td><strong>Subject to</strong> $g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0$</td>
<td><strong>Subject to</strong> $\frac{1}{3}d_1 + \frac{1}{3}d_2 \leq \frac{2}{3}$</td>
</tr>
<tr>
<td>$g_2(x) = -x_1 \leq 0$</td>
<td>$-d_1 \leq 1$</td>
</tr>
<tr>
<td>$g_3(x) = -x_2 \leq 0$</td>
<td>$-d_2 \leq 1$</td>
</tr>
</tbody>
</table>

**Kuhn-Tucker necessary condition**

1. $\frac{\partial L}{\partial d_1} = -1 + d_1 + \frac{1}{3}u_1 - u_2 = 0$
2. $\frac{\partial L}{\partial d_2} = -1 + d_2 + \frac{1}{3}u_1 - u_3 = 0$
3. $\frac{\partial L}{\partial u_1} = \frac{1}{3} (d_1 + d_2 - 2) + s_1^2 = 0$
4. $\frac{\partial L}{\partial u_2} = -d_1 - 1 + s_2^2 = 0$
5. $\frac{\partial L}{\partial u_3} = -d_2 - 1 + s_3^2 = 0$
6. $\frac{\partial L}{\partial s_i} = u_is_i = 0, \ u_i \geq 0, \ i = 1, 2, 3$

**Graphical Representation**

- Objective function: quadratic form
- Constraint: linear form

$d_i = x_i - 1, \ d_2 = x_2 - 1$
Formulation of the Quadratic Programming Problem (2/5)

Quadratic Programming Problem

Minimize \( \tilde{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2) \)

Subject to \( \frac{1}{3} d_1 + \frac{1}{3} d_2 \leq \frac{2}{3} \)

\[-d_1 \leq 1 \]
\[-d_2 \leq 1 \]

Kuhn-Tucker necessary condition

\[
\begin{align*}
\frac{\partial L}{\partial d_1} &= -1 + d_1 + \frac{1}{3} u_1 - u_2 = 0 \\
\frac{\partial L}{\partial d_2} &= -1 + d_2 + \frac{1}{3} u_1 - u_3 = 0 \\
\frac{\partial L}{\partial u_1} &= \frac{1}{3} (d_1 + d_2 - 2) + s_1^2 = 0 \\
\frac{\partial L}{\partial u_2} &= -d_1 - 1 + s_2^2 = 0 \\
\frac{\partial L}{\partial u_3} &= -d_2 - 1 + s_3^2 = 0 \\
\frac{\partial L}{\partial s_i} &= u_i s_i = 0, \; u_i \geq 0, \; i = 1, 2, 3 \quad u_i s_i^2 = 0, \; u_i \geq 0, \; i = 1, 2, 3
\end{align*}
\]

Multiply the both side of equations by \( s_i \)

Replace \( s_i^2 \) with \( s_i' \)

\( s_i' \geq 0 \)

\[
\begin{align*}
\frac{\partial L}{\partial d_1} &= -1 + d_1 + \frac{1}{3} u_1 - u_2 = 0 \\
\frac{\partial L}{\partial d_2} &= -1 + d_2 + \frac{1}{3} u_1 - u_3 = 0 \\
\frac{\partial L}{\partial u_1} &= \frac{1}{3} (d_1 + d_2 - 2) + s_1' = 0 \\
\frac{\partial L}{\partial u_2} &= -d_1 - 1 + s_2' = 0 \\
\frac{\partial L}{\partial u_3} &= -d_2 - 1 + s_3' = 0 \\
\frac{\partial L}{\partial s_i} &= u_i s_i' = 0 \\
u_i, s_i' \geq 0, \; i = 1, 2, 3
\end{align*}
\]

Represent \( s_i' \) to \( s_i \) for the convenience

\[
\begin{align*}
\frac{\partial L}{\partial d_1} &= -1 + d_1 + \frac{1}{3} u_1 - u_2 = 0 \\
\frac{\partial L}{\partial d_2} &= -1 + d_2 + \frac{1}{3} u_1 - u_3 = 0 \\
\frac{\partial L}{\partial u_1} &= \frac{1}{3} (d_1 + d_2 - 2) + s_1 = 0 \\
\frac{\partial L}{\partial u_2} &= -d_1 - 1 + s_2 = 0 \\
\frac{\partial L}{\partial u_3} &= -d_2 - 1 + s_3 = 0 \\
\frac{\partial L}{\partial s_i} &= u_i s_i = 0 \\
u_i, s_i \geq 0, \; i = 1, 2, 3
\end{align*}
\]
Formulation of the Quadratic Programming Problem (3/5)

**Quadratic Programming Problem**

Minimize \( f = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2) \)

Subject to \( \frac{1}{3} d_1 + \frac{1}{3} d_2 \leq \frac{2}{3} \)
\(- d_1 \leq 1 \)
\(- d_2 \leq 1 \)

Matrix form

Minimize \( \tilde{f} = c^T (1 \times 2) d_{(2 \times 1)} + \frac{1}{2} d^T (1 \times 2) H_{(2 \times 2)} d_{(2 \times 1)} \)

Subject to \( A^T (3 \times 2) d_{(2 \times 1)} \leq b_{(3 \times 1)} \)

Assume that \( H_{(2 \times 2)} \) is equal to \( I_{(2 \times 2)} \).

where, \( d_{(2 \times 1)} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \)
\( c_{(2 \times 1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix} \)
\( H_{(2 \times 2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \)

\( A_{(2 \times 3)} = \begin{bmatrix} \frac{1}{3} & -1 & 0 \\ \frac{1}{3} & 0 & -1 \end{bmatrix} \)
\( b_{(3 \times 1)} = \begin{bmatrix} 2/3 \\ 1 \\ 1 \end{bmatrix} \)

**Kuhn-Tucker necessary condition**

\( \frac{\partial L}{\partial d_1} = -1 + d_1 + \frac{1}{3} u_1 - u_2 = 0 \)
\( \frac{\partial L}{\partial d_2} = -1 + d_2 + \frac{1}{3} u_1 - u_3 = 0 \)
\( \frac{\partial L}{\partial u_1} = \frac{1}{3} (d_1 + d_2 - 2) + s_1 = 0 \)
\( \frac{\partial L}{\partial u_2} = -d_1 - 1 + s_2 = 0 \)
\( \frac{\partial L}{\partial u_3} = -d_2 - 1 + s_3 = 0 \)
\( \frac{\partial L}{\partial s_i} = u_i s_i = 0 \)

\( u_i, s_i \geq 0, i = 1, 2, 3 \)

How can we express the Kuhn-Tucker necessary condition in matrix form \((d, c, H, A, b)\)?
Kuhn-Tucker necessary condition

\[
\frac{\partial L}{\partial d_1} = -1 + d_1 + \frac{1}{3}u_1 - u_2 = 0
\]
\[
\frac{\partial L}{\partial d_2} = -1 + d_2 + \frac{1}{3}u_1 - u_3 = 0
\]
\[
\frac{\partial L}{\partial u_1} = \frac{1}{3}(d_1 + d_2 - 2) + s_1 = 0
\]
\[
\frac{\partial L}{\partial u_2} = -d_1 - 1 + s_2 = 0
\]
\[
\frac{\partial L}{\partial u_3} = -d_2 - 1 + s_3 = 0
\]
\[
\frac{\partial L}{\partial s_i} = u_is_i = 0
\]

\(u_i, s_i \geq 0, i = 1, 2, 3\)

Matrix form

\[
\begin{bmatrix}
\frac{1}{3}(d_1 + d_2 - 2) + s_1 \\
-d_1 - 1 + s_2 \\
-d_2 - 1 + s_3
\end{bmatrix}
= \begin{bmatrix}
\frac{1}{3} & \frac{1}{3} \\
0 & -1 \\
0 & -1
\end{bmatrix}
\begin{bmatrix}
d_1 \\
d_2 \\
s_1
\end{bmatrix}
+ \begin{bmatrix}
s_1 \\
s_2 \\
s_3
\end{bmatrix}
- \begin{bmatrix}
\frac{2}{3} \\
1 \\
1
\end{bmatrix}
= A^T_{(3x2)}d_{(2x1)} + s_{(3x1)} - b_{(3x1)} = 0
\]
Formulation of the Quadratic Programming Problem (5/5)

In order to use the Simplex method, we decompose into two variables, because the design variables \(d_{(n \times 1)}\) are free in sign:

\[
d_{(2 \times 1)} = d^+_{(2 \times 1)} - d^-_{(2 \times 1)}
\]

Kuhn-Tucker necessary condition in Matrix form:

\[
\nabla L(d^+, d^-, u, s) = 0
\]

\[
\begin{bmatrix}
H_{(2 \times 2)} & -H_{(2 \times 2)} & A_{(2 \times 3)} & 0_{(2 \times 3)} \\
A^T_{(3 \times 2)} & -A^T_{(3 \times 2)} & 0_{(3 \times 3)} & I_{(3 \times 3)}
\end{bmatrix}
\begin{bmatrix}
d^+_{(2 \times 1)} \\
d^-_{(2 \times 1)} \\
u_{(3 \times 1)} \\
s_{(3 \times 1)}
\end{bmatrix}
= \begin{bmatrix}
-c_{(2 \times 1)} \\
b_{(3 \times 1)}
\end{bmatrix} = X_{(10 \times 1)}
\]

where

\[
d^+_{(2 \times 1)} = \begin{bmatrix} d^+_1 \\ d^+_2 \end{bmatrix},
d^-_{(2 \times 1)} = \begin{bmatrix} d^-_1 \\ d^-_2 \end{bmatrix},
c_{(2 \times 1)} = \begin{bmatrix} -1 \\ -1 \end{bmatrix},
H_{(2 \times 2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix},
A_{(2 \times 3)} = \begin{bmatrix} \frac{1}{3} & -1 & 0 \\ \frac{1}{3} & 0 & -1 \end{bmatrix},
b_{(3 \times 1)} = \begin{bmatrix} \frac{2}{3} \\ 1 \end{bmatrix}
\]

\[
B_{(5 \times 10)} = \begin{bmatrix}
1 & 0 & -1 & 0 & \frac{1}{3} & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & \frac{1}{3} & 0 & -1 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
X^T_{(1 \times 10)} = \begin{bmatrix}
d^+_1 & d^+_2 & d^-_1 & d^-_2 & u_1 & u_2 & u_3 & s_1 & s_2 & s_3
\end{bmatrix}
\]

\[
D^T_{(1 \times 5)} = \begin{bmatrix} 1 & 1 & \frac{2}{3} & 1 & 1 \end{bmatrix}
\]
Kuhn-Tucker necessary condition(matrix form)

\[
B_{(5\times 10)} X_{(10\times 1)} = D_{(5\times 1)} = \begin{bmatrix}
1 & 0 & -1 & 0 & \frac{1}{3} & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & \frac{1}{3} & 0 & -1 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & -\frac{1}{3} & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
d^+ \\
d^+_1 \\
d^- \\
d^-_2 \\
u_1 \\
u_2 \\
u_3 \\
s_1 \\
s_2 \\
s_3 \\
\end{bmatrix} = \begin{bmatrix}
1 \\
1 \\
\frac{2}{3} \\
1 \\
1 \\
\end{bmatrix}
\]

We want to find.

- This problem is to find X in the linear programming problem only having the equality constraints.
- \( u_i s_i = 0; i = 1 to 3 \): Check whether the solution obtained from the linear indeterminate equation satisfies the nonlinear indeterminate equation and determine the solution.
Determine the Search Direction by Using the Simplex Method
- Iteration 1 (1/6)

Simplex method to solve the quadratic programming problem

1. The problem to solve the Kuhn-Tucker necessary condition is the same with the problem having only the equality constraints (Linear Programming problem).

2. To solve the linear indeterminate equation, we introduce the artificial variables, define the artificial objective function, and then determine the initial basic feasible solution by using the Simplex method.

   \[ \mathbf{B}_{(5 \times 10)} \mathbf{X}_{(10 \times 1)} + \mathbf{Y}_{(5 \times 1)} = \mathbf{D}_{(5 \times 1)} \]

   Artificial variables

3. The artificial objective function is defined as follows.

   \[ w = \sum_{i=1}^{5} Y_i = \sum_{i=1}^{5} D_i - \sum_{j=1}^{10} \sum_{i=1}^{5} B_{ij} X_j = w_0 + \sum_{j=1}^{10} C_j X_j \]

   where \( C_j = - \sum_{i=1}^{5} B_{ij} \) : Add the elements of the \( j \) th column of the matrix \( B \) and change the its sign. (Initial relative objective coefficient).

   \[ w_0 = \sum_{i=1}^{5} D_i = 1 + 1 + \frac{2}{3} + 1 + 1 = \frac{14}{3} \]

   : Initial value of the artificial objective function (summation of the all elements of the matrix \( D \))

4. Solve the linear programming problem by using the Simplex and check whether the solution satisfies the following nonlinear equation.

   \[ u_i s_i = 0; i = 1 \text{ to } 3 \]

   : Check whether the solution obtained from the linear indeterminate equation satisfies the nonlinear indeterminate equation and determine the solution.
Define the artificial objective function for using the Simplex method:

Sum all the rows(1~5):

\[
\frac{1}{3} X_1 + \frac{1}{3} X_2 - \frac{1}{3} X_3 - \frac{1}{3} X_4 + \frac{2}{3} X_5 - X_6 - X_7 + X_8 + X_9 + X_{10} + Y_1 + Y_2 + Y_3 + Y_4 + Y_5 = \frac{14}{3}
\]

Express the artificial function as \( w \) and rearrange:

\[
-\frac{1}{3} X_1 - \frac{1}{3} X_2 + \frac{1}{3} X_3 + \frac{1}{3} X_4 - \frac{2}{3} X_5 + X_6 + X_7 - X_8 - X_9 - X_{10} = w - \frac{14}{3}
\]

---

**Determine the Search Direction by Using the Simplex Method**

- **Iteration 1 (2/6)**

\[
B_{(5 \times 10)} X_{(10 \times 1)} + Y_{(5 \times 1)} = D_{(5 \times 1)} \Rightarrow
\]

Artificial variables

\[
\begin{bmatrix}
1 & 0 & -1 & 0 & \frac{1}{3} & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & \frac{1}{3} & 0 & -1 & 0 & 0 & 0 \\
\frac{1}{3} & \frac{1}{3} & -\frac{1}{3} & \frac{1}{3} & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
\begin{bmatrix}
d^+_1 (= X_1) \\
d^+_2 (= X_2) \\
d^-_3 (= X_3) \\
d^-_4 (= X_4) \\
u_5 (= X_5) \\
u_6 (= X_6) \\
u_7 (= X_7) \\
u_8 (= X_8) \\
u_9 (= X_9) \\
u_{10} (= X_{10})
\end{bmatrix}
\]

\[
\begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3 \\
Y_4 \\
Y_5
\end{bmatrix}
\]

\[
+ \begin{bmatrix}
1 \\
1 \\
\frac{2}{3} \\
1 \\
1
\end{bmatrix}
\]

 artificial objective function

Artificial variables

Sum all the elements of the row and change the its sign (ex. Row 1: \(-(1+0+1/3-1+0)=-1/3\)
### Determine the Search Direction by Using the Simplex Method - Iteration 1 (3/6)

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Y1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1/3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Y2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>1/3</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>X8</td>
<td>1/3</td>
<td>1/3</td>
<td>-1/3</td>
<td>-1/3</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2/3</td>
</tr>
<tr>
<td>Y4</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Y5</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-2/3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>w-4</td>
</tr>
</tbody>
</table>

### Results

- **Iteration 1:**
  - Objective Function Values:
    - Y4: 1
    - Y5: 0
    - A. Obj.: w-4

- **Search Direction:**
  - X9: -1
  - X10: -1
  - X1: 0
  - X2: 1
  - X3: 1
  - X4: 0
  - X5: 0
  - X6: 0
  - X7: 0
  - X8: 0

- **Objective Function Values:**
  - Y4: 1
  - Y5: 0
  - A. Obj.: w-4
Determine the Search Direction by Using the Simplex Method - Iteration 1 (4/6)

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1/3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Y2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>1/3</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>X8</td>
<td>0</td>
<td>1/3</td>
<td>0</td>
<td>-1/3</td>
<td>-1/9</td>
<td>1/3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>-1/3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1/3</td>
<td>1</td>
</tr>
<tr>
<td>X9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/3</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>X10</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>-1/3</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>w-1</td>
</tr>
</tbody>
</table>

Since the value of the objective function becomes zero, the initial basic feasible solution is obtained.
Determine the Search Direction by Using the Simplex Method - Iteration 1 (5/6)

\[
X^T_{(1\times 10)} = \begin{bmatrix} d_1^+ & d_2^+ & d_1^- & d_2^- & u_1 & u_2 & u_3 & s_1 & s_2 & s_3 \end{bmatrix}
\]

Basic solution: \( X_1 = 1, \ X_2 = 1, \ X_8 = 0, \ X_9 = 2, \ X_{10} = 2 \)

Nonbasic solution: \( X_3 = X_4 = X_5 = X_6 = X_7 = 0 \)

This solution satisfies the nonlinear indeterminate equation \( X_iX_{3+i} = 0; i = 5 \ to \ 7, \ X_i \geq 0; i = 1 \ to \ 10 \)

So, the optimal solution is \( d_1 = d_2 = 1, u_1 = u_2 = u_3 = 0, s_1 = 0, s_2 = s_3 = 2 \).

**Caution:** In the Pivot step, if the smallest (i.e., the most negative) coefficient of the artificial objective function or the smallest positive ratio “bi/ai” appears more than one time, the initial basic feasible solution can be changed depending on the selection of the pivot element in the pivot procedure.

**We have to check the solution until the nonlinear indeterminate equation** \( u_i^*s_i = 0 \) are satisfied.
The optimal solution for this problem is \( d_1 = d_2 = 1, u_1 = u_2 = u_3 = 0, s_1 = 0, s_2 = s_3 = 2 \).

Why are the values of \( u_1 \) and \( s_1 \) are zero at the same time?

Quadratic Programming Problem

\[
\text{Minimize} \quad \tilde{f} = (-d_1 - d_2) + 0.5(d_1^2 + d_2^2)
\]

\[
\text{Subject to} \quad \frac{1}{3} d_1 + \frac{1}{3} d_2 \leq 2 \quad \frac{1}{3} \quad \frac{1}{3} \\
- d_1 \leq 1 \\
- d_2 \leq 1
\]

This example is graphically represented in the right side.

However, fortunately, the optimal solution is on the linearized inequality constraint \((g_1(x), d_1 + d_2 = 2)\).

\( u_2 = u_3 = 0 \)

The optimal solution is in the region satisfying the inequality constraints of \( d_1 = -1 > 0, d_2 = -1 > 0 \) which are inactive.

\( u_1 = 0 \)

The optimal solution is on the inequality constraint \((g_1(x))\) and is equal to the optimal solution of the objective function to be approximated to the second order. Therefore, although we do not consider the inequality constraint \( g_1(x) \), the optimal solution of QP problem is not changed. \((g_1(x) \) does not affect the optimal solution of this problem.\)
The value of the $\alpha_0 = 0.732$ is found at which the penalty function is minimized in the interval between $x^{(0, 2)}$ and $x^{(0, 4)}$.

$x^{(1)} = x^{(0)} + \alpha_0 \cdot d^{(0)} = (1, 1) + 0.732 \cdot (1, 1) = (1.732, 1.732)$

$f(x^{(1)}) = f(1.732, 1.732) = -3$
Solve the QP problem to determine the search direction \(d^{(0)}\)

**Constrained Optimal Design Problem (Original problem)**

\[ \text{Minimize} \quad f(x) = x_1^2 + x_2^2 - 3x_1x_2 \]

\[ \text{Subject to} \quad g_1(x) = \frac{1}{6} x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \]
\[ g_2(x) = -x_1 \leq 0 \]
\[ g_3(x) = -x_1 \leq 0 \]

**Quadratic Programming Problem**

\[ \text{Minimize} \quad \bar{f} = (-1.732d_1 - 1.732d_2) + 0.5(d_1^2 + d_2^2) \]

\[ \text{Subject to} \quad 0.577d_1 + 0.577d_2 \leq 5.866 \times 10^{-5} \]
\[ -d_1 \leq 1.732 \]
\[ -d_2 \leq 1.732 \]

**Lagrange function**

\[ L = (-1.732d_1 - 1.732d_2) + 0.5(d_1^2 + d_2^2) + u_1[0.577(d_1 + d_2) - 5.866 \times 10^{-5} + s_1^2] \]
\[ + u_2(-d_1 - 1.732 + s_1^2) \]
\[ + u_3(-d_2 - 1.732 + s_2^3) \]

**Kuhn-Tucker necessary condition:**

\[ \nabla L(d,u,s) = 0 \]

\[ \frac{\partial L}{\partial d_1} = -1.732 + d_1 + 0.577u_1 - u_2 = 0 \]
\[ \frac{\partial L}{\partial d_2} = -1.732 + d_2 + 0.577u_1 - u_3 = 0 \]
\[ \frac{\partial L}{\partial u_1} = 0.577(d_1 + d_2) - 5.866 \times 10^{-6} + s_1^2 = 0 \]
\[ \frac{\partial L}{\partial u_2} = -d_1 - 1.732 + s_1^2 = 0 \]
\[ \frac{\partial L}{\partial u_3} = -d_2 - 1.732 + s_2^3 = 0 \]
\[ \frac{\partial L}{\partial s_i} = u_is_i = 0, u \geq 0, i = 1, 2, 3 \]

1. Multiply the both side by \(s_i\) and replace \(s_i^2\) with \(s_i'\).
2. Represent \(s_i'\) to \(s_i\) for the convenience.
Determine the Search Direction by Using the Simplex Method
- Iteration 2 (2/10)

**Quadratic Programming Problem**

Minimize \( \bar{f} = (-1.732d_1 - 1.732d_2) + 0.5(d_1^2 + d_2^2) \)

Subject to \( 0.577d_1 + 0.577d_2 \leq 0 \)

\(-d_1 \leq 1.732 \)

\(-d_2 \leq 1.732 \)

**Matrix form**

Minimize \( \bar{f} = c^T (1 \times 2) d (2 \times 1) + \frac{1}{2} d^T (1 \times 2) H (2 \times 2) d (2 \times 1) \)

Subject to \( A^T (3 \times 2) d (2 \times 1) \leq b (3 \times 1) \)

where

\[ d (2 \times 1) = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}, \quad c (2 \times 1) = \begin{bmatrix} -1.732 \\ -1.732 \end{bmatrix}, \quad H (2 \times 2) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \]

\[ A (2 \times 3) = \begin{bmatrix} 0.577 & -1 & 0 \\ 0.577 & 0 & -1 \end{bmatrix}, \quad b (3 \times 1) = \begin{bmatrix} 1.732 \\ 1.732 \end{bmatrix} \]

Assume that \( H (2 \times 2) \) is equal to \( I (2 \times 2) \).

**Kuhn-Tucker necessary condition**

\( \frac{\partial L}{\partial d_1} = -1.732 + d_1 + 0.577u_1 - u_2 = 0 \)

\( \frac{\partial L}{\partial d_2} = -1.732 + d_2 + 0.577u_1 - u_3 = 0 \)

\( \frac{\partial L}{\partial u_i} = 0.577(d_1 + d_2) - 5.866 \times 10^{-6} + s_i = 0 \)

\( \frac{\partial L}{\partial u_2} = -d_1 - 1.732 + s_2 = 0 \)

\( \frac{\partial L}{\partial u_3} = -d_2 - 1.732 + s_3 = 0 \)

\( \frac{\partial L}{\partial s_i} = u_is_i = 0, \quad u_i, s_i \geq 0, \quad i = 1, 2, 3 \)
Determine the Search Direction by Using the Simplex Method
- Iteration 2 (3/10)

Kuhn-Tucker necessary condition

\[
\begin{align*}
\frac{\partial L}{\partial d_1} &= -1.732 + d_1 + 0.577u_1 - u_2 = 0 \\
\frac{\partial L}{\partial d_2} &= -1.732 + d_2 + 0.577u_1 - u_3 = 0 \\
\frac{\partial L}{\partial u_1} &= 0.577(d_1 + d_2) - 5.866 \times 10^{-6} + s_1 = 0 \\
\frac{\partial L}{\partial u_2} &= -d_1 - 1.732 + s_2 = 0 \\
\frac{\partial L}{\partial u_3} &= -d_2 - 1.732 + s_3 = 0 \\
\frac{\partial L}{\partial s_i} &= u_i s_i = 0, u_i, s_i \geq 0, i = 1, 2, 3
\end{align*}
\]

Since the design variables \(d_{(n \times 1)}\) are free in sign, we may decompose them as follows for using the Simplex method.

\[
d_{(2 \times 1)} = d^+_{(2 \times 1)} - d^-_{(2 \times 1)}
\]

Matrix form

\[
\begin{bmatrix}
H_{(2 \times 2)} & -H_{(2 \times 2)} & A_{(2 \times 3)} & 0_{(2 \times 3)} \\
A^T_{(3 \times 2)} & -A^T_{(3 \times 2)} & 0_{(3 \times 3)} & I_{(3 \times 3)}
\end{bmatrix}
= B_{(5 \times 10)}
\begin{bmatrix}
d^+_{(2 \times 1)} \\
d^-_{(2 \times 1)} \\
u_{(3 \times 1)} \\
s_{(3 \times 1)}
\end{bmatrix}
= D_{(5 \times 1)}
\begin{bmatrix}
-c_{(2 \times 1)} \\
b_{(3 \times 1)}
\end{bmatrix}
= X_{(10 \times 1)}
\]
Determine the Search Direction by Using the Simplex Method - Iteration 2 (4/10)

Kuhn-Tucker necessary condition: \( \nabla L(d^+, d^-, u, s) = 0 \)

\[
\begin{bmatrix}
H_{(2 \times 2)} & -H_{(2 \times 2)} & A_{(2 \times 3)} & 0_{(2 \times 3)} \\
A^T_{(3 \times 2)} & -A^T_{(3 \times 2)} & 0_{(3 \times 3)} & I_{(3 \times 3)}
\end{bmatrix}
\begin{bmatrix}
d^+_{(2 \times 1)} \\
d^-_{(2 \times 1)} \\
u_{(3 \times 1)} \\
s_{(3 \times 1)}
\end{bmatrix}
= B_{(5 \times 10)}
\]

\[
= D_{(5 \times 1)} = X_{(10 \times 1)}
\]

where, \( d^+_{(2 \times 1)} = \begin{bmatrix} d^+_1 \\ d^+_2 \end{bmatrix}, d^-_{(2 \times 1)} = \begin{bmatrix} d^-_1 \\ d^-_2 \end{bmatrix}, c_{(2 \times 1)} = \begin{bmatrix} -1.732 \\ -1.732 \end{bmatrix}, H_{(2 \times 2)} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, A_{(2 \times 3)} = \begin{bmatrix} 0.577 & -1 & 0 \\ 0.577 & 0 & -1 \end{bmatrix}, b_{(3 \times 1)} = \begin{bmatrix} 0 \\ 1.732 \end{bmatrix} \)

\[
B_{(5 \times 10)} = \begin{bmatrix}
1 & 0 & -1 & 0 & 0.577 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0.577 & 0 & -1 & 0 & 0 & 0 \\
0.577 & 0.577 & -0.577 & -0.577 & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1
\end{bmatrix}
\]

\[
X^T_{(1 \times 10)} = \begin{bmatrix} d^+_1 & d^+_2 & d^-_1 & d^-_2 & u_1 & u_2 & u_3 & s_1 & s_2 & s_3 \end{bmatrix}, D^T_{(1 \times 5)} = \begin{bmatrix} 1.732 & 1.732 & 0 & 1.732 & 1.732 \end{bmatrix}
\]
Determine the Search Direction by Using the Simplex Method - Iteration 2 (5/10)

Kuhn-Tucker necessary condition (matrix form)

\[ B_{(5 \times 10)} X_{(10 \times 1)} = D_{(5 \times 1)} \]

\[
\begin{bmatrix}
1 & 0 & -1 & 0 & 0.577 & -1 & 0 & 0 & 0 & 0 \\
0 & 1 & 0 & -1 & 0.577 & 0 & -1 & 0 & 0 & 0 \\
0.577 & 0.577 & -0.577 & -0.577 & 0 & 0 & 0 & 1 & 0 & 0 \\
-1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & -1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
d_1^+ \\
d_2^+ \\
d_1^- \\
d_2^- \\
u_1 \\
u_2 \\
u_3 \\
s_1 \\
s_2 \\
s_3 \\
\end{bmatrix}
= \begin{bmatrix}
1.732 \\
1.732 \\
0 \\
1.732 \\
1.732 \\
\end{bmatrix}

We want to find.

⇒ This problem is to find X in the linear programming problem only having the equality constraints.
⇒ \(u_i s_i = 0; i = 1 \text{ to } 3\) : Check whether the solution obtained from the linear indeterminate equation satisfies the nonlinear indeterminate equation and determine the solution.
Determine the Search Direction by Using the Simplex Method
- Iteration 2 (6/10)

Simplex method to solve the quadratic programming problem

1. The problem to solve the Kuhn-Tucker necessary condition is the same with the problem having only the equality constraints (linear programming problem).
2. To solve the linear indeterminate equation, we introduce the artificial variables, define the artificial objective function and determine the initial basic feasible solution by using the Simplex method.

\[
B_{(5\times10)} X_{(10\times1)} + Y_{(5\times1)} = D_{(5\times1)}
\]

Artificial variables

3. The artificial objective function is defined as follows.

\[
w = \sum_{i=1}^{5} Y_i = \sum_{i=1}^{5} D_i - \sum_{j=1}^{10} \sum_{i=1}^{5} B_{ij} X_j = w_0 + \sum_{j=1}^{10} C_j X_j
\]

where \( C_j = -\sum_{i=1}^{5} B_{ij} \) : Add the elements of the \( j \) th column of the matrix B and change the its sign. (Initial relative objective coefficient).

\[
w_0 = \sum_{i=1}^{5} D_i = 1 + 1 + \frac{2}{3} + 1 + 1 = \frac{14}{3}
\] : Initial value of the artificial objective function (summation of the all elements of the matrix D)

4. Solve the linear programming problem by using the Simplex and check whether the solution satisfies the following equation.

\[
u_i s_i = 0; i = 1 \text{ to } 3
\] : Check whether the solution obtained from the linear indeterminate equation satisfies the nonlinear indeterminate equation and determine the solution.
Determine the Search Direction by Using the Simplex Method
- Iteration 2 (7/10)

\[
B_{(5\times10)}X_{(10\times1)} + Y_{(5\times1)} = D_{(5\times1)}
\]

Artificial variables

Define the artificial objective function for using the Simplex method

Sum the all rows(1~5): \(0.577X_1 + 0.577X_2 - 0.577X_3 - 0.577X_4 + 1.154X_5 - X_6 - X_7 + X_8 + X_9 + X_{10} + Y_1 + Y_2 + Y_3 + Y_4 + Y_5 = 6.928\)

Replace the summation of the all artificial to \(w\) and rearrange:

\[-0.577X_1 - 0.577X_2 + 0.577X_3 + 0.577X_4 - 1.154X_5 + X_6 + X_7 - X_8 - X_9 - X_{10} = w - 6.928\]

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0.577</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.732</td>
<td>3</td>
</tr>
<tr>
<td>Y2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0.577</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.732</td>
</tr>
<tr>
<td>Y3</td>
<td>0.577</td>
<td>0.577</td>
<td>-0.577</td>
<td>-0.577</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1.732</td>
</tr>
<tr>
<td>Y4</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1.732</td>
<td>-</td>
</tr>
<tr>
<td>Y5</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1.732</td>
<td>-</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>-0.577</td>
<td>-0.577</td>
<td>0.577</td>
<td>0.577</td>
<td>-1.154</td>
<td>1</td>
<td>1</td>
<td>-1</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>w-6.928</td>
<td>-</td>
</tr>
</tbody>
</table>

Artificial objective function

Sum all the elements of the row and change the its sign (ex. 1 row: \(-(1+0+1/3-1+0)=-1/3\)
## Determine the Search Direction by Using the Simplex Method

### - Iteration 2 (8/10)

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>1.732</td>
<td>0.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>3.000</td>
<td>-1.732</td>
<td></td>
</tr>
<tr>
<td>Y2</td>
<td>-1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Y3</td>
<td>0.577</td>
<td>0.577</td>
<td>-0.577</td>
<td>-0.577</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.732</td>
<td>1.732</td>
<td></td>
</tr>
<tr>
<td>Y5</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.732</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>1.423</td>
<td>-0.577</td>
<td>-1.423</td>
<td>0.577</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>-1.000</td>
<td>2.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>w-3.464</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>-1.732</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>3.000</td>
<td>-</td>
</tr>
<tr>
<td>X3</td>
<td>-1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Y3</td>
<td>0.000</td>
<td>1.155</td>
<td>0.000</td>
<td>-1.155</td>
<td>0.000</td>
<td>0.577</td>
<td>-0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.577</td>
<td>0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.732</td>
<td>-</td>
<td></td>
</tr>
<tr>
<td>Y5</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.732</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0.000</td>
<td>0.845</td>
<td>0.000</td>
<td>-0.845</td>
<td>0.000</td>
<td>0.423</td>
<td>-0.423</td>
<td>-1.000</td>
<td>-1.000</td>
<td>-1.000</td>
<td>0.577</td>
<td>1.423</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>w-3.464</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>-1.732</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>3.000</td>
<td>-</td>
</tr>
<tr>
<td>X3</td>
<td>-1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Y3</td>
<td>0.000</td>
<td>1.155</td>
<td>0.000</td>
<td>-1.155</td>
<td>0.000</td>
<td>0.577</td>
<td>-0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.577</td>
<td>0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.732</td>
<td>1.732</td>
<td></td>
</tr>
<tr>
<td>X10</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.732</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0.000</td>
<td>-0.155</td>
<td>0.000</td>
<td>0.155</td>
<td>0.000</td>
<td>0.423</td>
<td>-0.423</td>
<td>-1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.577</td>
<td>1.423</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>w-1.732</td>
<td></td>
</tr>
</tbody>
</table>
Determine the Search Direction by Using the Simplex Method - Iteration 2 (9/10)

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>-1.732</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>3.000</td>
<td>1.732</td>
</tr>
<tr>
<td>X3</td>
<td>-1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Y3</td>
<td>0.000</td>
<td>1.155</td>
<td>0.000</td>
<td>-1.155</td>
<td>0.000</td>
<td>0.577</td>
<td>-0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.577</td>
<td>0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>X9</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.732</td>
<td>-1.732</td>
</tr>
<tr>
<td>X10</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>-1.732</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0.000</td>
<td>-1.155</td>
<td>0.000</td>
<td>1.155</td>
<td>0.000</td>
<td>-0.577</td>
<td>-0.577</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.577</td>
<td>0.423</td>
<td>0.000</td>
<td>1.000</td>
<td>1.000</td>
<td>w-0.000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>1.732</td>
<td>0.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>3.000</td>
<td>1.732</td>
</tr>
<tr>
<td>X2</td>
<td>-1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>Y3</td>
<td>1.155</td>
<td>0.000</td>
<td>-1.155</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.577</td>
<td>0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.577</td>
<td>-0.577</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
</tr>
<tr>
<td>X9</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>1.732</td>
<td>-1.732</td>
</tr>
<tr>
<td>X10</td>
<td>-1.000</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-1.000</td>
<td>1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.732</td>
<td>-1.732</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>-1.155</td>
<td>0.000</td>
<td>1.155</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.577</td>
<td>-0.577</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.423</td>
<td>1.577</td>
<td>0.000</td>
<td>1.000</td>
<td>1.000</td>
<td>w-0.000</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>X9</th>
<th>X10</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>Y5</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>X5</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-0.866</td>
<td>-0.866</td>
<td>-1.500</td>
<td>0.000</td>
<td>0.000</td>
<td>0.866</td>
<td>0.866</td>
<td>0.866</td>
<td>-1.500</td>
<td>0.000</td>
<td>0.000</td>
<td>3.000</td>
</tr>
<tr>
<td>X2</td>
<td>0.000</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.500</td>
<td>-0.500</td>
<td>0.866</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.500</td>
<td>0.500</td>
<td>0.866</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>X1</td>
<td>1.000</td>
<td>0.000</td>
<td>-1.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.500</td>
<td>0.500</td>
<td>0.866</td>
<td>0.000</td>
<td>0.000</td>
<td>0.500</td>
<td>-0.500</td>
<td>0.866</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td></td>
</tr>
<tr>
<td>X9</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>-0.500</td>
<td>0.500</td>
<td>0.866</td>
<td>1.000</td>
<td>0.000</td>
<td>0.500</td>
<td>-0.500</td>
<td>0.866</td>
<td>1.000</td>
<td>0.000</td>
<td>1.732</td>
<td></td>
</tr>
<tr>
<td>X10</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.500</td>
<td>-0.500</td>
<td>0.866</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>-0.500</td>
<td>0.500</td>
<td>0.866</td>
<td>0.000</td>
<td>1.732</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>0.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>1.000</td>
<td>w-0.000</td>
<td></td>
</tr>
</tbody>
</table>
Determine the Search Direction by Using the Simplex Method  
- Iteration 2 (10/10)

\[
X^T_{(1\times10)} = \begin{bmatrix}
    d_1^+ & d_2^+ & d_1^- & d_2^- & u_1 & u_2 & u_3 & s_1 & s_2 & s_3
\end{bmatrix}
\]

Basic solution:  
\[X_5 = 3, \quad X_2 = 0, \quad X_1 = 0, \quad X_9 = 1.732, \quad X_{10} = 1.732\]

Nonbasic solution:  
\[X_3 = X_4 = X_6 = X_7 = X_8 = 0\]

This solution satisfy the nonlinear indeterminate equation (\(X_iX_{3+i} = 0; i = 5 \text{ to } 7, X_i \geq 0; i = 1 \text{ to } 10\)).

So, the optimal solution is \(d_1 = d_2 = 0, \quad u_1 = 3, \quad u_2 = u_3 = 0, \quad s_1 = 0, \quad s_2 = s_3 = 1.732\).

- In the Pivot step, if the smallest (i.e., the most negative) coefficient of the artificial objective function or the smallest positive ratio “bi/ai” appears more than one time, the initial basic feasible solution can be changed by depending on the selection of the pivot element in the pivot procedure.
- We have to find and check the solution until the nonlinear indeterminate equation (\(u_i^*s_i = 0\)) is satisfied.
7.5 Summary of the Sequential Quadratic Programming (SQP)
Formulation of the Quadratic Programming Problem to Determine the Search Direction

\[ \text{Minimize} \quad f(x + \Delta x) \equiv f(x) + \nabla f^T(x) \Delta x + 0.5 \Delta x^T H \Delta x \]

The second-order Taylor series expansion of the objective function

\[ \text{Subject to} \quad h_j(x + \Delta x) \equiv h_j(x) + \nabla h_j^T(x) \Delta x = 0; \quad j = 1 \text{ to } p \]

The first-order (linear) Taylor series expansion of the equality constraints

\[ g_j(x + \Delta x) \equiv g_j(x) + \nabla g_j^T(x) \Delta x \leq 0; \quad j = 1 \text{ to } m \]

The first-order (linear) Taylor series expansion of the inequality constraints

Define:

\[ \tilde{f} = f(x + \Delta x) - f(x), \quad e_j = -h_j(x), \quad b_j = -g_j(x), \]

\[ c_i = \partial f(x)/\partial x_i, \quad n_{ij} = \partial h_j(x)/\partial x_i, \quad a_{ij} = \partial g_j(x)/\partial x_i, \]

\[ d_i = \Delta x_i \]

Matrix form

\[ \text{Minimize} \quad \tilde{f} = c^T_{(1 \times n)} d_{(n \times 1)} + \frac{1}{2} d^T_{(1 \times n)} H_{(n \times n)} d_{(n \times 1)} \quad : \text{Quadratic objective function} \]

\[ \text{Subject to} \quad N^T_{(p \times n)} d_{(n \times 1)} = e_{(p \times 1)} \quad : \text{Linear equality constraints} \]

\[ A^T_{(m \times n)} d_{(n \times 1)} \leq b_{(m \times 1)} \quad : \text{Linear inequality constraints} \]
Determination of the Step Size by Using the Golden Section Search Method

Trial design point for which the descent condition is checked

\[ \mathbf{x}^{(k,j)} = \mathbf{x}^{(k)} + \alpha_{(k,j)} \mathbf{d}^{(k)} \]  

How can we determine the value of the \( \alpha_{(k,j)} \) to find the improved design point?

Find the improved design point which minimizes the descent function more than the current point by changing \( \alpha_{(k,j)} \). (One dimensional search method, such as the Golden section search method, can be used.)

Determination of the improved design point \( \mathbf{x}^{(k+1)} \) by using the one dimensional search method such as the Golden section search method (\( \mathbf{x}^{(k,j)} \) is changed to \( \mathbf{x}^{(k+1)} \)).

After finding the interval in which the minimum lies, find the minimum point, \( \mathbf{x} \), by reducing the interval (Golden section search method).
Formulation of the Quadratic Programming Problem

\[
\begin{align*}
\text{Minimize} \quad & \bar{f} = c^T (1 \times n) d_{(n \times 1)} + \frac{1}{2} d^T (1 \times n) H_{(n \times n)} d_{(n \times 1)} \\
\text{Subject to} \quad & N^T (p \times n) d_{(n \times 1)} = e_{(p \times 1)} \\
& A^T (m \times n) d_{(n \times 1)} \leq b_{(m \times 1)}
\end{align*}
\]

\[\text{Assumption: } H_{(n \times n)} = I_{(n \times n)}\]

\[
\begin{align*}
\text{Minimize} \quad & \bar{f} = c^T (1 \times n) d_{(n \times 1)} + \frac{1}{2} d^T (1 \times n) I_{(n \times n)} d_{(n \times 1)} = c^T (1 \times n) d_{(n \times 1)} + \frac{1}{2} d^T (1 \times n) d_{(n \times 1)} \\
\text{Subject to} \quad & N^T (p \times n) d_{(n \times 1)} = e_{(p \times 1)} \\
& A^T (m \times n) d_{(n \times 1)} \leq b_{(m \times 1)}
\end{align*}
\]

\[\Rightarrow \text{Since } H_{(n \times n)} = I_{(n \times n)}, \text{ the objective function is a quadratic form.}\]

\[\Rightarrow \text{All constraints are linear.}\]

\[\Rightarrow \text{This problem is called the convex programming problem and any local optimum solution is also a global optimum solution.}\]
Define the QP problem to determine the search direction $d^{(k)}$ and calculate the Lagrange multiplier at the given design point $x^{(k)}$.

Solve the QP problem (solution: search direction $d^{(k)}$, Lagrange multiplier) by using the Lagrange function and Kuhn-Tucker necessary condition.

Check for the stopping criteria $|d^{(k)}| \leq \varepsilon_2$ and the maximum constraint violation $V_k \leq \varepsilon_1$.

Yes

Set $x^* = x^{(k)}$ and stop.

No

Find the improved design point ($x^{(k+1)}$) to minimize the descent function along the search direction ($d^{(k)}$) by using the one dimensional search method (ex: Golden section search method)

$ k = k + 1 $
Summary of the SQP Algorithm (1/2)

☑ Step 1: Set $k = 0$. Estimate the initial value for the design variables as $x^{(0)}$. Select an appropriate initial value for the penalty parameter $R_0$, and two small number $\varepsilon_1$, $\varepsilon_2$ that define the permissible constraint violations and convergence parameter values, respectively.

☑ Step 2: At $x^{(k)}$, compute the objective and constraint functions and their gradient. Calculate the maximum constraint violation $V_k$.

☑ Step 3: Using the objective and constraints function values and their gradients, define the QP problem. Solve the QP problem to obtain the search direction $d^{(k)}(= x^{(k+1)} - x^{(k)})$ and Lagrange multiplier $v^{(k)}$ and $u^{(k)}$. 
Summary of the SQP Algorithm (2/2)

✓ Step 4: Check for the stopping criteria $|d^{(k)}| \leq \varepsilon_2$ and the maximum constraint violation $V_k \leq \varepsilon_1$. If these criteria are satisfied then stop. Otherwise continue.

✓ Step 5: Calculate the sum $r_k$ of the Lagrange multiplier. Set $R = \max\{R_k, r_k\}$.

✓ Step 6: Set $x^{(k,j)} = x^{(k)} + \alpha_{(k,j)} d^{(k)}$, where $\alpha = \alpha_{(k,j)}$ is a proper step size. As for the unconstrained problems, the step size can be obtained by minimizing the descent function along the search direction $d^{(k)}$. The one dimensional search method, such as the Golden section search method, can be used to determine the optimum step size. (If the one dimensional search method is completed, the current design point $x^{(k,j)}$ is changed to $x^{(k+1)}$.)

✓ Step 7: Save the current penalty parameter as $R_k = R$. Update the iteration counter as $k = k+1$ and go to Step 2.
Effect of the Starting Point in the SQP Algorithm

The starting point can affect performance of the algorithm.

For example, at some points, the Quadratic Programming problem defined to determine the search direction may not have any solution.
This does not mean that the original problem is infeasible.
The original problem may be highly nonlinear, so that the linearized constraints may be inconsistent giving infeasible Quadratic Programming problem.

This situation can be handled by either temporarily deleting the inconsistent constraints or starting from another point.
[Appendix] Another Method for Solving Nonlinear Indeterminate Equations

Applying the Simplex Method
Applying the Simplex Method
- Eliminate variables using relevant equations and introduce new ‘virtual’ linear variables $x'$.

**Kuhn-Tucker necessary condition:** \( \nabla L(x, u, s, \zeta, \delta) = 0 \)

\[
\begin{align*}
\frac{\partial L}{\partial x_1} & = -2 + 2x_1 - 2u_1 - u_2 - \zeta_1 = 0, & \frac{\partial L}{\partial x_2} & = -2 + 2x_2 - u_1 - 2u_2 - \zeta_2 = 0 \\
\frac{\partial L}{\partial u_1} & = -2x_1 - x_2 + 4 + s_1^2 = 0, & \frac{\partial L}{\partial u_2} & = -x_1 - 2x_2 + 4 + s_2^2 = 0 \\
\frac{\partial L}{\partial s_1} & = 2u_1s_1 = 0, & \frac{\partial L}{\partial s_2} & = 2u_2s_2 = 0
\end{align*}
\]

\[\Rightarrow \text{Redefine } s_i^2 \text{ as } s'_i\]

**Reformulated Kuhn-Tucker necessary condition:**

\[
\begin{align*}
\frac{\partial L}{\partial x_1} & = -2 + 2x_1 - 2u_1 - u_2 - \zeta_1 = 0, & \frac{\partial L}{\partial x_2} & = -2 + 2x_2 - u_1 - 2u_2 - \zeta_2 = 0 \\
\frac{\partial L}{\partial u_1} & = -2x_1 - x_2 + 4 + s'_1 = 0, & \frac{\partial L}{\partial u_2} & = -x_1 - 2x_2 + 4 + s'_2 = 0 \\
\frac{\partial L}{\partial s'_1} & = 2u_1s'_1 = 0, & \frac{\partial L}{\partial s'_2} & = 2u_2s'_2 = 0
\end{align*}
\]

\[\text{where } u_i, \zeta_i, s'_i, x_i \geq 0\]

We eliminate two variables using relevant two equations and also introduce new variable $s'$ instead of $s^2$. 

...
**Kuhn-Tucker necessary condition:** \( \nabla L(x,u,s,\zeta,\delta) = 0 \)

\[
\begin{align*}
\frac{\partial L}{\partial x_1} &= -2 + 2x_1 - 2u_1 - u_2 - \zeta_1 = 0, \\
\frac{\partial L}{\partial x_2} &= -2 + 2x_2 - u_1 - 2u_2 - \zeta_2 = 0 \\
\frac{\partial L}{\partial u_1} &= -2x_1 - x_2 + 4 + s'_1 = 0, \\
\frac{\partial L}{\partial u_2} &= -x_1 - 2x_2 + 4 + s'_2 = 0 \\
\frac{\partial L}{\partial s_1} &= 2u_1 s'_1 = 0, \\
\frac{\partial L}{\partial s_2} &= 2u_2 s'_2 = 0 \\
\frac{\partial L}{\partial \xi_1} &= 2\xi_1 x_1 = 0, \\
\frac{\partial L}{\partial \xi_2} &= 2\xi_2 x_2 = 0
\end{align*}
\]

where \( u_1, \xi_1, s'_1, x_1 \geq 0 \)

---

**Solve the linear indeterminate equations by using the Simplex method of Phase 1.**

Define the standard LP problem:

\[
\begin{bmatrix}
2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\
0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\
-2 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\
-1 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
u_1 \\
u_2 \\
\xi_1 \\
\xi_2 \\
s'_1 \\
s'_2 \\
\end{bmatrix}
= \begin{bmatrix} 2 \\
2 \\
-4 \\
-4 \\
\end{bmatrix}
\]

Multiply both side of the constraints by -1:

\[
\begin{bmatrix}
2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\
0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\
-2 & -1 & 0 & 0 & 0 & 0 & 1 & 0 \\
-1 & -2 & 0 & 0 & 0 & 0 & 0 & 1 \\
\end{bmatrix}
\begin{bmatrix}
x_1 \\
x_2 \\
u_1 \\
u_2 \\
\xi_1 \\
\xi_2 \\
s'_1 \\
s'_2 \\
\end{bmatrix}
= \begin{bmatrix} 2 \\
0 \\
0 \\
0 \\
\end{bmatrix}
\]

---

**Computer Aided Ship Design, I-7 Constrained Nonlinear Optimization Method, Fall 2013, Myung-II Roh**
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (2/9)

Introduce the artificial variables to treat the linear equality constraints.

\[
\begin{bmatrix}
2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\
0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
x_1 (= X_1) \\
x_2 (= X_2) \\
u_1 (= X_3) \\
u_2 (= X_4) \\
\zeta_1 (= X_5) \\
\zeta_2 (= X_6) \\
s'_1 (= X_7) \\
s'_2 (= X_8)
\end{bmatrix}
+ \begin{bmatrix}
Y_1 \\
Y_2 \\
Y_3 \\
Y_4
\end{bmatrix}
= \begin{bmatrix}
2 \\
2 \\
4 \\
4
\end{bmatrix}
\]

Define the artificial objective function as sum of all the artificial variables \((Y_1+Y_2+Y_3+Y_4)\).

\[
5x_1 + 5x_2 - 3u_1 - 3u_2 - \zeta_1 - \zeta_2 - s'_1 - s'_2 + Y_1 + Y_2 + Y_3 + Y_4 = 12
\]

\[
-w
\]

\[
-5x_1 - 5x_2 + 3u_1 + 3u_2 + \zeta_1 + \zeta_2 + s'_1 + s'_2 = w - 12 : \text{Artificial objective function}
\]
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (3/9)

\[
\begin{bmatrix}
2 & 0 & -2 & -1 & -1 & 0 & 0 & 0 \\
0 & 2 & -1 & -2 & 0 & -1 & 0 & 0 \\
2 & 1 & 0 & 0 & 0 & 0 & -1 & 0 \\
1 & 2 & 0 & 0 & 0 & 0 & 0 & -1
\end{bmatrix}
\begin{bmatrix}
x_1 (= X_1) \\
x_2 (= X_2) \\
u_1 (= X_3) \\
u_2 (= X_4) \\
\zeta_1 (= X_5) \\
\zeta_2 (= X_6) \\
s'_1 (= X_7) \\
s'_2 (= X_8)
\end{bmatrix}
+ \begin{bmatrix} Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{bmatrix} = \begin{bmatrix} 2 \\ 2 \\ 4 \\ 4 \end{bmatrix}
\]

\[-5x_1 - 5x_2 + 3u_1 + 3u_2 + \zeta_1 + \zeta_2 + s'_1 + s'_2 = w - 12 : \text{Artificial objective function}\n\]

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Y2</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>-</td>
</tr>
<tr>
<td>Y3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>Y4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>-5</td>
<td>-5</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>w-12</td>
<td>-</td>
</tr>
</tbody>
</table>
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (4/9)

<table>
<thead>
<tr>
<th>2</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>-1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>-</td>
</tr>
<tr>
<td>Y2</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Y3</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Y4</td>
<td>0</td>
<td>2</td>
<td>1</td>
<td>1/2</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>0</td>
<td>1</td>
<td>3</td>
<td>3/2</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0</td>
<td>-5</td>
<td>-2</td>
<td>1/2</td>
<td>-3/2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>5/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>w-7</td>
<td>-</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>3</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>-1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>-1/2</td>
<td>-1</td>
<td>0</td>
<td>-1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Y3</td>
<td>0</td>
<td>0</td>
<td>5/2</td>
<td>2</td>
<td>1</td>
<td>1/2</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5/2</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>-9/2</td>
<td>-9/2</td>
<td>-3/2</td>
<td>-3/2</td>
<td>1</td>
<td>1</td>
<td>5/2</td>
<td>5/2</td>
<td>0</td>
<td>0</td>
<td>w-2</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>4</th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3/10</td>
<td>-1/10</td>
<td>1/5</td>
<td>-2/5</td>
<td>0</td>
<td>1/10</td>
<td>-1/5</td>
<td>2/5</td>
<td>0</td>
<td>7/5</td>
</tr>
<tr>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-3/5</td>
<td>1/5</td>
<td>-2/5</td>
<td>-1/5</td>
<td>0</td>
<td>-1/5</td>
<td>2/5</td>
<td>1/5</td>
<td>0</td>
<td>6/5</td>
<td></td>
</tr>
<tr>
<td>X3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4/5</td>
<td>2/5</td>
<td>1/5</td>
<td>-2/5</td>
<td>0</td>
<td>-2/5</td>
<td>-1/5</td>
<td>2/5</td>
<td>0</td>
<td>2/5</td>
<td></td>
</tr>
<tr>
<td>Y4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9/10</td>
<td>-3/10</td>
<td>3/5</td>
<td>4/5</td>
<td>-1</td>
<td>3/10</td>
<td>-3/5</td>
<td>-4/5</td>
<td>1</td>
<td>1/5</td>
<td></td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>-9/10</td>
<td>3/10</td>
<td>-3/5</td>
<td>-4/5</td>
<td>1</td>
<td>7/10</td>
<td>8/5</td>
<td>9/5</td>
<td>0</td>
<td>w-1/5</td>
<td></td>
</tr>
</tbody>
</table>
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (5/9)

Since the value of the objective function becomes zero, the initial basic feasible solution is obtained.

\[ X^T_{(1\times8)} = \begin{bmatrix} x_1 & x_2 & u_1 & u_2 & \zeta_1 & \zeta_2 & s'_1 & s'_2 \end{bmatrix} \]

One of the initial basic feasible solutions is \( X_1=X_2=4/3, X_3=X_4=2/9, X_5=X_6=X_7=X_8=0 \).

\[ x_1 = x_2 = \frac{4}{3}, u_1 = u_2 = \frac{2}{9}, \zeta_1 = \zeta_2 = s'_1 = s'_2 = 0 \]

And this solution satisfies the following nonlinear equations (constraints).

\[ u_2s'_2 = 0, \quad u_is'_t = 0, \quad \zeta_1x_1 = 0, \quad \zeta_2x_2 = 0 \]

Therefore, the optimal solution of this problem is \( x_1 = x_2 = \frac{4}{3}, u_1 = u_2 = \frac{2}{9}, \zeta_1 = \zeta_2 = s'_1 = s'_2 = 0 \).

This result is same as that obtained by the first method.
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (6/9)

Instead of choosing the first column as the pivot column, we can choose the second column, because the coefficient of the objective function is same.

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y1</td>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y2</td>
<td>0</td>
<td>2</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y3</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y4</td>
<td>1</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>A. Obj.</td>
<td>-5</td>
<td>-5</td>
<td>3</td>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>w-12</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y1</td>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>-1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>-1/2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y3</td>
<td>2</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y4</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>A. Obj.</td>
<td>-5</td>
<td>0</td>
<td>1/2</td>
<td>-2</td>
<td>1</td>
<td>-3/2</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>5/2</td>
<td>0</td>
<td>0</td>
<td>w-7</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>-1/2</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>-1/2</td>
<td>-1</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y3</td>
<td>0</td>
<td>0</td>
<td>5/2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Y4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5/2</td>
<td>1/2</td>
</tr>
<tr>
<td></td>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>5/2</td>
<td>2</td>
<td>1</td>
<td>-3/2</td>
<td>3</td>
<td>5/2</td>
<td>5/2</td>
<td>0</td>
<td>0</td>
<td>w-2</td>
<td>-</td>
</tr>
</tbody>
</table>
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (7/9)

### Table 4

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3/10</td>
<td>-1/10</td>
<td>1/5</td>
<td>-2/5</td>
<td>0</td>
<td>1/10</td>
<td>-1/5</td>
<td>2/5</td>
<td>0</td>
<td>7/5</td>
<td>-</td>
</tr>
<tr>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-6/10</td>
<td>1/5</td>
<td>-2/5</td>
<td>-1/5</td>
<td>0</td>
<td>-1/5</td>
<td>2/5</td>
<td>1/5</td>
<td>0</td>
<td>6/5</td>
<td>-</td>
</tr>
<tr>
<td>X3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>4/5</td>
<td>2/5</td>
<td>1/5</td>
<td>-2/5</td>
<td>0</td>
<td>-2/5</td>
<td>-1/5</td>
<td>2/5</td>
<td>0</td>
<td>2/5</td>
<td>-</td>
</tr>
<tr>
<td>Y4</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9/10</td>
<td>-3/10</td>
<td>3/5</td>
<td>4/5</td>
<td>-1</td>
<td>3/10</td>
<td>-3/5</td>
<td>-4/5</td>
<td>1</td>
<td>1/5</td>
<td>1/4</td>
</tr>
</tbody>
</table>

A. Obj. | 0 | 0 | 0 | -9/10 | 3/10 | -3/5 | -4/5 | 1 | 7/10 | 8/5 | 9/5 | 0 | w-1/5 |

### Table 5

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>3/4</td>
<td>-1/4</td>
<td>1/2</td>
<td>0</td>
<td>-1/2</td>
<td>-1/4</td>
<td>-1/2</td>
<td>0</td>
<td>1/2</td>
<td>3/2</td>
<td>-</td>
</tr>
<tr>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>-3/8</td>
<td>1/8</td>
<td>-1/4</td>
<td>0</td>
<td>-1/4</td>
<td>-1/8</td>
<td>1/4</td>
<td>0</td>
<td>1/4</td>
<td>5/4</td>
<td>-</td>
</tr>
<tr>
<td>X3</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>5/4</td>
<td>1/4</td>
<td>1/2</td>
<td>0</td>
<td>-1/2</td>
<td>-1/4</td>
<td>-1/2</td>
<td>0</td>
<td>1/2</td>
<td>1/2</td>
<td>-</td>
</tr>
<tr>
<td>X7</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>9/8</td>
<td>-3/8</td>
<td>3/4</td>
<td>-5/4</td>
<td>3/8</td>
<td>-3/4</td>
<td>-1</td>
<td>5/4</td>
<td>1/4</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

A. Obj. | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | w-0 |

\[ X^T \}_{(1x8)} = \begin{bmatrix} x_1 & x_2 & u_1 & u_2 & \zeta_1 & \zeta_2 & s_1' & s_2' \end{bmatrix} \]

Since the value of the objective function becomes zero, the initial basic feasible solution is obtained.

The another initial basic feasible solution is \( X_1=3/2, X_2=5/4, X_3=1/2, X_4=X_5=X_6=0, X_7=1/4, X_8=0 \).

\( x_1 = \frac{2}{3}, x_2 = \frac{5}{4}, u_1 = \frac{1}{2}, u_2 = \zeta_1 = \zeta_2 = 0, s_1' = \frac{1}{4}, s_2' = 0 \)

But this solution does not satisfy the constraint of \( u_i s_i' = 0 \).

Therefore, this solution cannot be the optimal solution.

- When the smallest(i.e., the most negative) coefficient of the artificial objective function or the smallest positive ratio “bi/ai,” appears more than one entry, the initial basic feasible solution can be changed depending on the selection of the pivot element in the pivot operation.
- We have to check whether the solution obtained by the Simplex method satisfies the nonlinear equation. (constraint, \( u_i s_i' = 0 \)).
Applying the Simplex Method of Phase 1 for Solving Linear Indeterminate Equations (8/9)

In the tableau 3, if we choose the column 4 as the pivot column which has the same coefficient of the artificial objective function as of the column 3, what will happen?

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>-1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>-1/2</td>
<td>-1</td>
<td>0</td>
<td>-1/2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Y3</td>
<td>0</td>
<td>0</td>
<td>5/2</td>
<td>2</td>
<td>1</td>
<td>1/2</td>
<td>-1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td>Y4</td>
<td>0</td>
<td>0</td>
<td>2</td>
<td>5/2</td>
<td>1/2</td>
<td>1</td>
<td>0</td>
<td>-1</td>
<td>-1/2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>-9/2</td>
<td>-9/2</td>
<td>-3/2</td>
<td>-3/2</td>
<td>1</td>
<td>1</td>
<td>5/2</td>
<td>5/2</td>
<td>0</td>
<td>0</td>
<td>w-2</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-6/10</td>
<td>0</td>
<td>-2/5</td>
<td>1/5</td>
<td>0</td>
<td>-1/5</td>
<td>2/5</td>
<td>-1/5</td>
<td>0</td>
<td>1/5</td>
<td>6/5</td>
</tr>
<tr>
<td></td>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>3/10</td>
<td>0</td>
<td>1/5</td>
<td>-1/10</td>
<td>0</td>
<td>-2/5</td>
<td>-1/5</td>
<td>1/10</td>
<td>0</td>
<td>2/5</td>
<td>7/5</td>
</tr>
<tr>
<td></td>
<td>Y3</td>
<td>0</td>
<td>0</td>
<td>9/10</td>
<td>0</td>
<td>3/5</td>
<td>-3/10</td>
<td>-1</td>
<td>4/5</td>
<td>-3/5</td>
<td>3/10</td>
<td>1</td>
<td>-4/5</td>
<td>1/5</td>
</tr>
<tr>
<td></td>
<td>Y4</td>
<td>0</td>
<td>0</td>
<td>4/5</td>
<td>1</td>
<td>1/5</td>
<td>2/5</td>
<td>0</td>
<td>-2/5</td>
<td>-1/5</td>
<td>-2/5</td>
<td>0</td>
<td>2/5</td>
<td>2/5</td>
</tr>
<tr>
<td></td>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>-9/10</td>
<td>0</td>
<td>-3/5</td>
<td>3/10</td>
<td>1</td>
<td>-4/5</td>
<td>8/5</td>
<td>7/10</td>
<td>0</td>
<td>9/5</td>
<td>w-1/5</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>X5</th>
<th>X6</th>
<th>X7</th>
<th>X8</th>
<th>Y1</th>
<th>Y2</th>
<th>Y3</th>
<th>Y4</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>X1</td>
<td>1</td>
<td>0</td>
<td>-3/8</td>
<td>0</td>
<td>-1/4</td>
<td>1/8</td>
<td>-1/4</td>
<td>0</td>
<td>1/4</td>
<td>-1/8</td>
<td>1/4</td>
<td>0</td>
<td>5/4</td>
</tr>
<tr>
<td></td>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>3/4</td>
<td>0</td>
<td>1/2</td>
<td>-1/4</td>
<td>-1/2</td>
<td>0</td>
<td>-1/2</td>
<td>-1/4</td>
<td>1/2</td>
<td>0</td>
<td>3/2</td>
</tr>
<tr>
<td></td>
<td>X8</td>
<td>0</td>
<td>0</td>
<td>9/8</td>
<td>0</td>
<td>3/4</td>
<td>-3/8</td>
<td>-5/4</td>
<td>1</td>
<td>3/4</td>
<td>3/8</td>
<td>5/4</td>
<td>-1</td>
<td>1/4</td>
</tr>
<tr>
<td></td>
<td>X4</td>
<td>0</td>
<td>0</td>
<td>5/4</td>
<td>1</td>
<td>1/2</td>
<td>1/4</td>
<td>-1/2</td>
<td>0</td>
<td>-1/2</td>
<td>-1/4</td>
<td>1/2</td>
<td>0</td>
<td>1/2</td>
</tr>
<tr>
<td></td>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>w-0</td>
</tr>
</tbody>
</table>
Since the value of the objective function becomes zero, the initial basic feasible solution is obtained.

\[
X^T_{(1x8)} = \begin{bmatrix} x_1 & x_2 & u_1 & u_2 & \xi_1 & \xi_2 & s_1 & s_2 \end{bmatrix}
\]

The another initial basic feasible solution is \(X_1=5/4, X_2=3/2, X_3=0, X_4=1/2, X_5=X_6=0=X_7=0, X_8=1/4\).

\[
x_1 = \frac{4}{3}, x_2 = \frac{5}{4}, u_1 = 0, u_2 = \frac{1}{2}, \xi_1 = \xi_2 = s'_1 = 0, s'_2 = \frac{1}{4}
\]

But this solution does not satisfy the constraint of \(u_2s'_2 = 0\).

Therefore, this solution cannot be the optimal solution.
[Reference] Reason to Decompose the Unrestricted Variable into the Difference of Two Nonnegative Design Variables for Using the Simplex Method (1/2)

To use the Simplex method, the variables must be nonnegative in the LP problem.

We can use the Simplex method only for the case that all the variables are nonnegative at the optimal point.

The variables unrestricted in sign at the optimal point should be decomposed into the difference of two nonnegative variables in the LP problem.

Minimize \( z = -y_1 - 2y_2 \)

Subject to
\[
\begin{align*}
3y_1 + 2y_2 &\leq 12 \\
2y_1 + 3y_2 &\geq 6 \\
y_1 &\geq 0 \\
y_2 &\text{ is unrestricted in sign.}
\end{align*}
\]

\( y_2 = y_2^+ - y_2^- \)
\( y_2^+, y_2^- \geq 0 \)

Minimize \( f = -y_1 - 2y_2^+ + 2y_2^- \)

Subject to
\[
\begin{align*}
3y_1 + 2y_2^+ - 2y_2^- + x_1 &\leq 12 \\
2y_1 + 3y_2^+ - 3y_2^- - x_2 + x_3 &\geq 6 \\
y_1, y_2^+, y_2^- &\geq 0, \quad x_i \geq 0; i = 1 \text{ to } 3
\end{align*}
\]

Since \( y_2 \) is free in sign, it should be decomposed into the difference of two nonnegative variables.

For “\( \leq \)” type inequality constraint, we introduce the slack variable.

For “\( \geq \)” type inequality constraint, we introduce the surplus variable and the artificial variable.

Solve the problem using the Simplex method.
Minimize \[ f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \] : Quadratic objective function

where, \( x_1, x_2 \) are free in sign.

Minimize \[ f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \]

Definition of the Lagrange function

\[ L(x_1, x_2) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \]

\[ \frac{\partial L}{\partial x_1} = 2x_1 + 2 = 0 \quad \ldots \ldots \ (1) \]

\[ \frac{\partial L}{\partial x_2} = 2x_2 - 2 = 0 \quad \ldots \ldots \ (2) \]

We try to solve this problem by using the Simplex method

Since these constraints are the equality constraints, we must introduce artificial variables for the equality constraints and define an artificial objective function, and solve it.

\[-2x_1 = 2 \]
\[2x_2 = 2 \]

\[ 2x_1 - y_1 = 2 \quad \ldots \ldots \ (3) \]
\[ 2x_2 + y_2 = 2 \quad \ldots \ldots \ (4) \]

Since the artificial objective function is the sum of all the artificial variables, its minimum value must be zero.

Eq. \( 3 + 4 \longrightarrow -2x_1 + 2x_2 + y_1 + y_2 = 4 \]
\[ 2x_1 - 2x_2 = y_1 + y_2 - 4 \]

Redefine the variables as \( x_1 = X_1, x_2 = X_2, y_1 = Y_1, y_2 = Y_2 \) and express in Matrix form.

- All coefficients of the artificial objective function become nonnegative.
- However, the sum of all the artificial variables \( w \) does not become zero. \( x_1 = 0, x_2 = 1, y_1 = 2, y_2 = 0 \)
- The simplex method does not give the optimum solution of \( x_1 = -1, x_2 = 1 \), rather \( x_1 = 0, x_2 = 1 \).
- The reason is that the simplex method assume that all the variables are nonnegative, whereas the variables \( x_1, x_2 \) of this example are free in sign. From this, we can see that to use the simplex method, the unrestricted variables must be decomposed into the two nonnegative variables.
[Review]

Minimize \( f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \) : Quadratic objective function

Subject to \( x_1 \geq 0 \) : Linearized inequality constraint

Minimize \( f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \)

Subject to \( x_1 \geq 0 \rightarrow -x_1 \leq 0 \rightarrow -x_1 + \delta^2 = 0 \)

Definition of the Lagrange function

\[ L(x_1, x_2, \zeta, \delta) = x_1^2 + x_2^2 + 2x_1 - 2x_2 + \zeta (-x_1 + \delta^2) \]

Kuhn-Tucker necessary condition: \( \nabla L(x_1, x_2, \zeta, \delta) = 0 \)

\[
\begin{align*}
\frac{\partial L}{\partial x_1} &= 2x_1 + 2 - \zeta = 0 \quad \cdots (1) \\
\frac{\partial L}{\partial x_2} &= 2x_2 - 2 = 0 \quad \cdots (2) \\
\frac{\partial L}{\partial \zeta} &= -x_1 + \delta^2 = 0 \quad \cdots (3) \\
\frac{\partial L}{\partial \delta} &= 2\zeta \delta = 0 \quad \cdots (4)
\end{align*}
\]

If we assume \( \zeta = 0, \ x_1 = -1 \) \rightarrow The equation \( 3 \) is not satisfied.

If we assume \( \delta = 0, \ x_1 = 0, x_2 = 1, \zeta = 2 \)
Solving the Problem by Using the Simplex Method (1/2)

**Minimize** \( f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \) : Quadratic objective function  
**Subject to** \( x_1 \geq 0 \) : Linearized inequality constraint

**Minimize** \( f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2 \)  
**Subject to** \( x_1 \geq 0 \rightarrow -x_1 \leq 0 \rightarrow -x_1 + \delta^2 = 0 \)

Definition of the Lagrange function  
\[ L(x_1, x_2, \zeta, \delta) = x_1^2 + x_2^2 + 2x_1 - 2x_2 + \zeta(-x_1 + \delta^2) \]

Kuhn-Tucker necessary condition: \( \nabla L(x_1, x_2, \zeta, \delta) = 0 \)

We try to solve this problem by using the Simplex method.

\[ \frac{\partial L}{\partial x_1} = 2x_1 + 2 - \zeta = 0 \]  
\[ \frac{\partial L}{\partial x_2} = 2x_2 - 2 = 0 \]  
\[ \frac{\partial L}{\partial \zeta} = -x_1 + \delta^2 = 0 \]  
\[ \frac{\partial L}{\partial \delta} = 2\zeta\delta = 0 \]

Since \( x_2 \) is free in sign, we may decompose it as  
\[ x_2 = x_2^+ - x_2^- \]

\[ \frac{\partial L}{\partial x_1} = 2x_1 + 2 - \zeta = 0 \quad \text{...... ①} \]  
\[ \frac{\partial L}{\partial x_2} = 2x_2^+ - 2x_2^- + 2 = 0 \quad \text{...... ②} \]  
\[ \frac{\partial L}{\partial \zeta} = -x_1 + \delta^2 = 0 \quad \text{...... ③} \]  
\[ \frac{\partial L}{\partial \delta} = 2\zeta\delta = 0 \quad \text{...... ④} \]

Multiply the both side of equation ④ by \( \delta \) and substitute the equation ③ into that. We eliminate one variable \( \delta \) and one equation.

1st stage: Find the solution satisfying the equation ①, ②, and ⑤.

2nd stage: Check whether the solution obtained in the 1st stage satisfies the nonlinear equation ⑤ or not.

① \( \rightarrow \) \( 2x_1 - \zeta = -2 \)  
② \( \rightarrow \) \( 2x_2^+ - 2x_2^- = 2 \)  
③ \( \rightarrow \) \( x_1 = \delta^2 \)  
④ \( \rightarrow \) \( 2\zeta\delta^2 = 0 \)  
\[ \zeta x_1 = 0 \quad \text{...... ⑤} \]
Solving the Problem by Using the Simplex Method (2/2)

**Minimize** $f(x) = x_1^2 + x_2^2 + 2x_1 - 2x_2$ : Quadratic objective function

**Subject to** $x_1 \geq 0$ : Linearized inequality constraint

\[
\begin{align*}
2x_1 - \zeta &= -2 \\
2x_2^+ - 2x_2^- &= 2 \\
-2x_1 + \zeta &= 2 \\
2x_2^+ - 2x_2^- &= 2
\end{align*}
\]

The right hand sides of the equations have to be nonnegative.

\[
\begin{align*}
2x_1 - \zeta &= -2 \\
2x_2^+ - 2x_2^- &= 2
\end{align*}
\]

The artificial variables have to be zero.

Since the constraints are the equality constraints, introduce the artificial variables.

\[
\begin{align*}
2x_1 - \zeta + y_1 &= 2 \\
2x_2^+ - 2x_2^- + y_2 &= 2
\end{align*}
\]

Since the equation (5) is satisfied, this is the solution of this problem.

---

**Basic variable**

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>Y1</th>
<th>Y2</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>Y2</td>
<td>0</td>
<td>2</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>2</td>
<td>-2</td>
<td>2</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>w-4</td>
</tr>
</tbody>
</table>

**Basic variable**

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>Y1</th>
<th>Y2</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y1</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>2</td>
<td>0</td>
<td>0</td>
<td>-1</td>
<td>0</td>
<td>1</td>
<td>w-2</td>
</tr>
</tbody>
</table>

**Basic variable**

<table>
<thead>
<tr>
<th>X1</th>
<th>X2</th>
<th>X3</th>
<th>X4</th>
<th>Y1</th>
<th>Y2</th>
<th>bi</th>
<th>bi/ai</th>
</tr>
</thead>
<tbody>
<tr>
<td>X4</td>
<td>-2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>X2</td>
<td>0</td>
<td>1</td>
<td>-1</td>
<td>0</td>
<td>0</td>
<td>1/2</td>
<td>1</td>
</tr>
<tr>
<td>A. Obj.</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>w-0</td>
</tr>
</tbody>
</table>

$X_1 = 0, X_2 = 1, X_3 = 0, X_4 = 2, Y_1 = 0, Y_2 = 0$

$x_1 = 0, x_2^+ = 1, x_2^- = 0, \zeta = 2, y_1 = 0, y_2 = 0$

$x_2 = x_2^+ - x_2^- = 1 - 0 = 1$

$\zeta x_1 = 2 \cdot 0 = 0 \quad \boxed{5}$ Since the equation (5) is satisfied, this is the solution of this problem.
Having the Design Variables whose Sign is Unrestricted (1/2)

Matrix Form

$$\begin{bmatrix}
H_{(n\times n)} & -H_{(n\times n)} & A_{(n\times m)} & 0_{(n\times m)} & N_{(n\times p)} & -N_{(n\times p)} \\
A^T_{(m \times n)} & -A^T_{(m \times n)} & 0_{(m \times m)} & I_{(m \times m)} & 0_{(m \times p)} & 0_{(m \times p)} \\
N^T_{(p \times n)} & -N^T_{(p \times n)} & 0_{(p \times m)} & 0_{(p \times m)} & 0_{(p \times p)} & 0_{(p \times p)} \\
\end{bmatrix}
= B_{((n+m+p) \times (2n+2m+2p))}$$

$$u_i s'_i = 0; i = 1 \text{ to } m$$

$$B_{((n+m+p) \times (2n+2m+2p))}X_{((2n+2m+2p) \times 1)} = D_{((n+m+p) \times 1)}$$

The number of the design variables is the same with that of the equations as $n+2m+p$ in the original problem. Since the equations $v_{(p \times 1)} = y_{(p \times 1)} - z_{(p \times 1)}$ and $d_{(n \times 1)} = d^+_{(n \times 1)} - d^-_{(n \times 1)}$ are introduced, the number of the design variables is also increased by $n+p$.

The interesting variables $v_i$ and $d_i$ are determined by the equation $v_{(p \times 1)} = y_{(p \times 1)} - z_{(p \times 1)}$.

Example

<table>
<thead>
<tr>
<th>Equation</th>
<th>Replace $z$ as $z_1 - z_2$</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x + y + z = 2$</td>
<td>$x + y + z_1 - z_2 = 2$</td>
<td>$x = 1, y = 3, z = -2$</td>
</tr>
<tr>
<td>$2x + 2y + z = 6$</td>
<td>$2x + 2y + z_1 - z_2 = 6$</td>
<td>$x = 1, y = 3, z_1 = 0, z_2 = 2$</td>
</tr>
<tr>
<td>$2x + y = 5$</td>
<td>$2x + y = 5$</td>
<td>$x = 1, y = 3, z = -2$</td>
</tr>
</tbody>
</table>

After replacing the variable, this problem becomes the indeterminate equation. The value of $z_1 - z_2$ is always $-2$.

Reference: Solution of the Problem of Constrained Nonlinear Optimization Method, Fall 2013, Myung Il Roh
[Reference] Solution of the Problem
Having the Design Variables whose Sign is Unrestricted (2/2)

Example

<table>
<thead>
<tr>
<th>Equation</th>
<th>( x + y + z = 5 )</th>
<th>( 2x + 3y + z = 11 )</th>
<th>( xz = 0 )</th>
<th>( x + y + z_1 - z_2 = 5 )</th>
<th>( 2x + 3y + z_1 - z_2 = 11 )</th>
<th>( xz = 0 )</th>
</tr>
</thead>
</table>

Case #1

Introduce the artificial variables for using the Simplex method
\[
\begin{align*}
x + y + z + Y_1 &= 5 \\
2x + 3y + z + Y_2 &= 11
\end{align*}
\]

Solve this problem by assuming the three design variables as zero.
Stop the Simplex method, if the sum of all the artificial variables \((Y_1+Y_2)\) zero.

<table>
<thead>
<tr>
<th>(x, y, z, Y1, Y2)</th>
<th>( x + y + z_1 - z_2 = 5 )</th>
<th>( 2x + 3y + z_1 - z_2 = 11 )</th>
<th>( xz = 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>① (4, 1, 0, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>② (6, 0, -1, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
<tr>
<td>③ (0, 3, 2, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
</tr>
</tbody>
</table>

Between the solution ① and ③ obtained by using the Simplex method, the final solution has to satisfy the equation \( xz = 0 \).

If the solution whose value of \( z = z_1 - z_2 \) is negative is excluded, the solution of the Case #1 is the same with that of the Case #2.

Case #2

Introduce the artificial variables for using the Simplex method
\[
\begin{align*}
x + y + z_1 - z_2 + Y_1 &= 5 \\
2x + 3y + z_1 - z_2 + Y_2 &= 11
\end{align*}
\]

Solve this problem by assuming the four design variables as zero.
Stop the Simplex method, if the sum of all the artificial variables \((Y_1+Y_2)\) zero.

<table>
<thead>
<tr>
<th>(x, y, z1, z2, Y1, Y2)</th>
<th>( z = z_1 - z_2 )</th>
<th>( x + y + z_1 - z_2 = 5 )</th>
<th>( 2x + 3y + z_1 - z_2 = 11 )</th>
<th>( xz = 0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>① (4, 1, 0, 0, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
<tr>
<td>② (6, 0, 0, 1, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
<tr>
<td>③ (6, 0, -1, 0, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
<tr>
<td>④ (0, 0, 0, 0, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
<tr>
<td>⑤ (0, 3, 0, -2, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
<tr>
<td>⑥ (0, 3, 2, 0, 0, 0)</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td>( 0 )</td>
<td></td>
</tr>
</tbody>
</table>

Among the solution ①, ②, and ⑥ obtained by using the Simplex method, the final solution has to satisfy the equation \( xz = 0 \).
[Appendix] Another Method for Sequential Quadratic Programming (SQP)

Use of the Descent Condition Method for SQP
Instead of the Golden Section Search Method
Use of the Descent Condition Method for SQP
Instead of the Golden Section Search Method (1/4)

(vi) Step 6: By using the one dimensional search method (ex. Descent condition method), calculate the step size to minimize the descent function along the search direction ($d^{(0)}$) and determine the improved design point.

\[
\Phi(x^{(k,j)}) = f(x^{(k,j)}) + R_k \cdot V(x^{(k,j)})
\]

\[
= x_1^2 + x_2^2 - 3x_1x_2 + 10 \cdot V(x^{(k,j)})
\]

\[
V(x^{(k,j)}) = \max \{0, g_1(x^{(k,j)}), g_2(x^{(k,j)}), g_3(x^{(k,j)})\}, (k=0)
\]

\[
x^{(k,j)} = x^{(k)} + t^{(k,j)}d^{(k)}
\]

\[
x^{(k,j)} \rightarrow \text{k iteration of CSD algorithm} \quad j \rightarrow \text{j iteration of one dimensional search method}
\]

\[
d_0 = (1,1) \quad x^{(0,0)} = (1,1),
\]

\[
\Phi(x^{(1,j)}) = \Phi(x^{(0)} + t^{(0,j)}d^{(0)}) = \Phi(t^{(0,j)})
\]

\[
\Phi(x^{(0,0)}) = \Phi(x^{(0,0)}) - t^{(0,j)}\beta_k
\]

where, $\beta_k = \gamma \|d^{(k)}\|^2$, ($\gamma = 0.5$, Defined by user)

By reducing the value of $i$ from 1 to a half, find the point to satisfy the following equation.

\[
\Phi(t^{(0,j)}) = -1 - t^{(0,j)}\beta_k
\]

where, $\beta_k = \gamma \|d^{(k)}\|^2 = 0.5(1^2 + 1^2) = 1$

\[
\Phi(x^{(0,0)}) = f(x^{(0,0)}) + R_0 \cdot V(x^{(0,0)}) = -1 + 10 \times 0 = -1
\]

\[
V(x^{(0,0)}) = \max \{0, -\frac{1}{3}, -1, -1\} = 0
\]

\[
f(x) = x_1^2 + x_2^2 - 3x_1x_2 \geq 0
\]

\[
g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0
\]

\[
g_2(x) = x_1 \leq 0
\]

\[
g_3(x) = x_2 \leq 0
\]

Point to be found by the Descent condition method

Point to be found by the Golden section search method
Use of the Descent Condition Method for SQP
Instead of the Golden Section Search Method (2/4)

(vi) Step 6: By using the one dimensional search method (ex. Descent condition method), calculate the step size to minimize the descent function along the search direction \( d^{(0)} \) and determine the improved design point.

\[
\Phi(x^{(k,j)}) = f(x^{(k,j)}) + R_k \cdot V(x^{(k,j)})
\]

\[
= x_1^2 + x_2^2 - 3x_1x_2 + 10 \cdot V(x^{(k,j)})
\]

\[
V(x^{(k,j)}) = \max \{0, g_1(x^{(k,j)}), g_2(x^{(k,j)}), g_3(x^{(k,j)})\}, \quad (k=0)
\]

\[
x^{(k,j)} = x^{(k)} + t_{(k,j)}d^{(k)}
\]

By reducing the value of \( t \) from 1 to a half, find the point to satisfy the following equation.

\[
\Phi(t_{(0,j)}) \leq -1 - t_{(0,j)} \quad k = 0, \ j = 0
\]

When \( t_{(0,j)} = 1 \)

\[
x^{(0,j)} = x^{(0)} + t_{(0,j)} \cdot d^{(0)} = (1,1) + 1 \cdot (1,1) = (2,2)
\]

\[
\Phi(t_{(0,j)}) = f(2,2) + R_0 \cdot V(x^{(0,j)}) = -4 + 10 \times 0.333 = -0.667
\]

where, \( V(x^{(0,j)}) = \max \{0, \frac{1}{3}, -2, -2\} = 0.333 \)

\[-1 - t_{(0,j)} = -1 - 1 = -2\]

If \( \Phi(t_{(0,j)}) \leq -1 - t_{(0,j)} \) is not satisfied, \( t \) is reduced to 0.5.
Use of the Descent Condition Method for SQP
Instead of the Golden Section Search Method (3/4)

(vi) Step 6: By using the one dimensional search method (ex. Descent Condition method), calculate the step size to minimize the descent function along the search direction ($d^{(0)}$) and determine the improved design point.

$$\Phi(x^{(k,j)}) = f(x^{(k,j)}) + R_k \cdot V(x^{(k,j)})$$

$$= x_1^2 + x_2^2 - 3x_1x_2 + 10 \cdot V(x^{(k,j)})$$

$$V(x^{(k,j)}) = \max \{0, g_1(x^{(k,j)}), g_2(x^{(k,j)}), g_3(x^{(k,j)})\}$$

$$x^{(k,j)} = x^{(k)} + t_{(k,j)}d^{(k)}$$

By reducing the value of $t$ from 1 to a half, find the point to satisfy the following equation.

$$-2.25 = \Phi(t^{(0,j)}) \leq -1 - t^{(0,j)} \leq -1.5$$

When $t^{(0,j)} = 0.5$

$$x^{(0,j)} = x^{(0)} + t^{(0,j)} \cdot d^{(0)} = (1,1) + 0.5 \cdot (1,1) = (1.5,1.5)$$

$$\Phi(t^{(0,j)}) = f(1.5,1.5) + R_0 \cdot V(x^{(0,j)}) = -2.25 + 10 \times 0 = -2.25$$

where, $V(x^{(0,j)}) = \max \{0, -\frac{2}{8}, -1.5, -1.5\} = 0$

$$-1 - t^{(0,j)} = -1 - 0.5 = -1.5$$

Since $\Phi(t^{(0,j)}) \leq -1 - t^{(0,j)}$ is satisfied, (1.5, 1.5) is the next design point.

The step size obtained by Descent condition is different from the step size obtained by Golden section search method.

Since the improved design points obtained by two method are different, the number of iteration of defining the QP problem is changed.

If we use the **Golden section search method** in the right example,

- The number of iteration of the one dimensional search in the first iteration of CSD is 62.
- By defining the QP problem two times, we can find the optimal design point.
  + The step size obtained by one dimensional search direction is exact size.

If we use the **Descent condition method** in the right example,

- The number of iteration of the one dimensional search in the first iteration of CSD is 1.
- Since the step size obtained by one dimensional search direction is not exact size, the QP problem should be defined in 20 times to find the optimal solution.
Comparison between the Golden Section Search Method and Descent Condition Method - Example #1 (1/2)

**Minimize**  
\( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

**Subject to**  
\( g_1(x) = \frac{1}{6} x_1^2 + \frac{1}{6} x_2^2 - 1.0 \leq 0 \)  
\( g_2(x) = -x_1 \leq 0 \)  
\( g_3(x) = -x_1 \leq 0 \)

Optimal Solution:  
\( x^* = (\sqrt{3}, \sqrt{3}), f(x^*) = -3 \)
Comparison between the Golden Section Search Method and Descent Condition Method - Example #1 (2/2)

Minimize: \( f(x) = x_1^2 + x_2^2 - 3x_1x_2 \)

Subject to: 
\[
g_1(x) = \frac{1}{6}x_1^2 + \frac{1}{6}x_2^2 - 1.0 \leq 0 \\
g_2(x) = -x_1 \leq 0 \\
g_3(x) = -x_1 \leq 0
\]

Solution: \( x = (\sqrt{3}, \sqrt{3}) \), \( f(x) = -3 \)

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Method</th>
<th>Iteration of defining the QP problem</th>
<th>Iteration of one dimensional search method</th>
<th>Local Optimum Point</th>
<th>Optimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(1, 1)</td>
<td>Descent condition method</td>
<td>19</td>
<td>19</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.0 )</td>
<td>19</td>
<td>19</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.1 )</td>
<td>19</td>
<td>19</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.5 )</td>
<td>19</td>
<td>19</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.9 )</td>
<td>19</td>
<td>19</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>1</td>
<td>62</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td>(0.1, 0.1)</td>
<td>Descent condition method</td>
<td>35</td>
<td>85</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.0 )</td>
<td>36</td>
<td>52</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.1 )</td>
<td>29</td>
<td>44</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.5 )</td>
<td>44</td>
<td>124</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.9 )</td>
<td>1</td>
<td>38</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>1</td>
<td>38</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td>(1.5, 1.5)</td>
<td>Descent condition method</td>
<td>18</td>
<td>18</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.0 )</td>
<td>18</td>
<td>18</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.1 )</td>
<td>18</td>
<td>18</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.5 )</td>
<td>18</td>
<td>18</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>( r = 0.9 )</td>
<td>18</td>
<td>18</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>2</td>
<td>68</td>
<td>(1.732, 1.732)</td>
<td>-3.0</td>
</tr>
</tbody>
</table>
Comparison between the Golden Section Search Method and Descent Condition Method - Example #2

**Minimize:** \( f(x_1, x_2) = x_1 - x_2 + 2x_1^2 + 2x_1x_2 + x_2^2 \)

**Solution:** \( x = (-1.0, 1.5) \), \( f(x) = -1.25 \)

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Method</th>
<th>Iteration of defining the QP problem</th>
<th>Iteration of one dimensional search method</th>
<th>Local Optimum Point</th>
<th>Optimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>39</td>
<td>59</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>38</td>
<td>58</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>41</td>
<td>67</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>60</td>
<td>127</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td></td>
<td></td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td>(1, 1)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>40</td>
<td>63</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>40</td>
<td>63</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>40</td>
<td>66</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>72</td>
<td>194</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td></td>
<td></td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td>(-1, 2)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>35</td>
<td>55</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>35</td>
<td>55</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>37</td>
<td>61</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>66</td>
<td>177</td>
<td>(-1.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td></td>
<td></td>
<td>(-1.0, 1.5)</td>
</tr>
</tbody>
</table>
Comparison between the Golden Section Search Method and Descent Condition Method - Example #3 (1/2)

Minimize

\[ f(x_1, x_2) = -\left[ 25 - (x_1 - 5)^2 - (x_2 - 5)^2 \right] \]

Subject to

\[ g_1(x_1, x_2) = -32 + 4x_1 + x_2^2 \leq 0 \]
\[ g_2(x_1, x_2) = -x_1 \leq 0 \]
\[ g_3(x_1, x_2) = x_1 \leq 10 \]
\[ g_4(x_1, x_2) = -x_2 \leq 0 \]
\[ g_5(x_1, x_2) = x_2 \leq 10 \]

Solution

\[ x_1^* = 4.374, x_2^* = 3.808, f(x_1^*, x_2^*) = -4.815 \]
Comparison between the Golden Section Search Method and Descent Condition Method - Example #3 (2/2)

\[ \text{Minimize: } f(x_1, x_2) = -\left[ 25 - (x_1 - 5)^2 - (x_2 - 5)^2 \right] \quad \text{Subject to: } \begin{align*} g_1(x_1, x_2) &= -32 + 4x_1 + x_2^2 \leq 0 \\ g_2(x_1, x_2) &= -x_1 \leq 0 \\ g_3(x_1, x_2) &= x_1 \leq 10 \\ g_4(x_1, x_2) &= -x_2 \leq 0 \\ g_5(x_1, x_2) &= x_2 \leq 10 \end{align*} \]

\[ \text{Solution: } x = (4.374, 3.808), \quad f(x) = -4.815 \]

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Method</th>
<th>Iteration of defining the QP problem</th>
<th>Iteration of one dimensional search method</th>
<th>Local Optimum Point</th>
<th>Optimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>22</td>
<td>23</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>22</td>
<td>23</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>22</td>
<td>23</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>22</td>
<td>24</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>590</td>
<td>13,509</td>
<td>(4.374, 3.808)</td>
<td>-23.188</td>
</tr>
<tr>
<td>(7, 1)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>15</td>
<td>22</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>15</td>
<td>22</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>15</td>
<td>22</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>24</td>
<td>45</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>1143</td>
<td>26,804</td>
<td>(4.374, 3.808)</td>
<td>-23.188</td>
</tr>
<tr>
<td>(-3, -10)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>19</td>
<td>35</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>19</td>
<td>35</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>19</td>
<td>35</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>28</td>
<td>61</td>
<td>(4.374, 3.808)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>884</td>
<td>20,005</td>
<td>(4.374, 3.808)</td>
<td>-23.188</td>
</tr>
</tbody>
</table>
Comparison between the Golden Section Search Method and Descent Condition Method - Example #4 (1/2)

Find \( x_1 (= B / T), x_2 (= 1 / C_B) \)

Minimize \( f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2 + 10 \)

Subject to \( g_1(x_1, x_2) = x_1 - 3 \leq 0 \)
\( g_2(x_1, x_2) = x_2 - 5 / 3 \leq 0 \)

Optimization problem having two unknown variables and two inequality constraints

Contour line \((f = \text{const.})\) of objective function

\( \mathbf{A} \): True solution \( x_1^* = 3.0, x_2^* = 1.5, f(x_1^*, x_2^*) = 2.5 \)

Feasible region
Comparison between the Golden Section Search Method and Descent Condition Method - Example #4 (2/2)

Minimize: \( f(x_1, x_2) = x_1^2 + 2x_2^2 - 4x_1 - 2x_1x_2 + 10 \)

Subject to: \( g_1(x_1, x_2) = x_1 - 3 \leq 0 \)
\( g_2(x_1, x_2) = x_2 - 5/3 \leq 0 \)

Solution: \( x = (3.0, 1.5), \ f(x) = 2.5 \)

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Method</th>
<th>Iteration of defining the QP problem</th>
<th>Iteration of one dimensional search method</th>
<th>Local Optimum Point</th>
<th>Optimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>Descent condition method</td>
<td>( r = 0.0 )</td>
<td>22</td>
<td>24</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.1 )</td>
<td>22</td>
<td>24</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.5 )</td>
<td>22</td>
<td>26</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.9 )</td>
<td>24</td>
<td>33</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td>13</td>
<td>203</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td>(2, 1)</td>
<td>Descent condition method</td>
<td>( r = 0.0 )</td>
<td>19</td>
<td>20</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.1 )</td>
<td>19</td>
<td>20</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.5 )</td>
<td>19</td>
<td>20</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.9 )</td>
<td>19</td>
<td>20</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td>4</td>
<td>89</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td>(-3, -5)</td>
<td>Descent condition method</td>
<td>( r = 0.0 )</td>
<td>26</td>
<td>52</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.1 )</td>
<td>25</td>
<td>28</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.5 )</td>
<td>25</td>
<td>28</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>( r = 0.9 )</td>
<td>25</td>
<td>30</td>
<td>(3.0, 1.5)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td>9</td>
<td>255</td>
<td>(3.0, 1.5)</td>
</tr>
</tbody>
</table>
Comparison between the Golden Section Search Method and Descent Condition Method - Example #5 (1/2)

Goldstein-Price Function

Minimize

\[ f(x_1, x_2) = (1 + (x_1 + x_2 + 1)^2 \cdot (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)) \cdot (30 + (2x_1 - 3x_2)^2 \cdot (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)) \]

Subject to

\[ g_1(x_1, x_2) = -2 - x_1 \leq 0, \quad g_2(x_1, x_2) = -2 - x_2 \leq 0, \]
\[ g_3(x_1, x_2) = x_1 - 2 \leq 0, \quad g_4(x_1, x_2) = x_2 - 2 \leq 0 \]

\[ f(x_1, x_2) \]

A : Global Minimum
\[ x_1^* = 0.0, \quad x_2^* = -1.0, \quad f(x_1^*, x_2^*) = 3.0 \]

B : Local Minimum
\[ x_1^* = -0.6, \quad x_2^* = -0.4, \quad f(x_1^*, x_2^*) = 30.0 \]

C : Local Minimum
\[ x_1^* = 1.2, \quad x_2^* = 0.8, \quad f(x_1^*, x_2^*) = 840.0 \]

D : Local Minimum
\[ x_1^* = 1.8, \quad x_2^* = 0.2, \quad f(x_1^*, x_2^*) = 84.0 \]
**Comparison between the Golden Section Search Method and Descent Condition Method - Example #5 (2/2)**

Minimize:

\[ f(x_1, x_2) = \{1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \times 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \} \]

Subject to:

\[
\begin{align*}
g_1(x_1, x_2) &= -2 - x_1 \leq 0, \\
g_2(x_1, x_2) &= -2 - x_2 \leq 0, \\
g_3(x_1, x_2) &= x_1 - 2 \leq 0, \\
g_4(x_1, x_2) &= x_2 - 2 \leq 0
\end{align*}
\]

In this example, since there are some local minimum design points, the optimal solution to be obtained is changed depending on the initial design point. So, the calculating the optimal solutions by assuming the initial design point in many times and comparing the results are needed.

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Method</th>
<th>Iteration of defining the QP problem</th>
<th>Iteration of one dimensional search method</th>
<th>Local Optimum Point</th>
<th>Optimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0, 0)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>30</td>
<td>302</td>
<td>(-0.6, -0.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>26</td>
<td>258</td>
<td>(-0.6, -0.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>21</td>
<td>208</td>
<td>(-0.6, -0.4)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>62</td>
<td>739</td>
<td>(-0.6, -0.4)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td>15</td>
<td>467</td>
<td>(-0.6, -0.4)</td>
</tr>
<tr>
<td>(2, 3)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>77</td>
<td>605</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>31</td>
<td>194</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>28</td>
<td>172</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>56</td>
<td>523</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td>13</td>
<td>417</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td>(-5, -5)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>70</td>
<td>545</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>24</td>
<td>135</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>24</td>
<td>136</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>51</td>
<td>459</td>
<td>(0.0, -1.0)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td></td>
<td>17</td>
<td>497</td>
<td>(0.0, -1.0)</td>
</tr>
</tbody>
</table>
Comparison between the Golden Section Search Method and Descent Condition Method - Example #6 (1/3)

Rastrigin’s Function

\[ \text{Minimize} \]
\[ f(x_1, x_2) = 20 + x_1^2 - 10 \cos(2\pi \cdot x_1) + x_2^2 - 10 \cos(2\pi \cdot x_2) \]

\[ \text{Subject to} \]
\[ g_1(x_1, x_2) = -5.12 - x_1 \leq 0 \]
\[ g_2(x_1, x_2) = -5.12 - x_2 \leq 0 \]
\[ g_3(x_1, x_2) = x_1 - 5.12 \leq 0 \]
\[ g_4(x_1, x_2) = x_2 - 5.12 \leq 0 \]

\[ \text{Solution} \]
\[ x_1^* = 0.0, x_2^* = 0.0, f(x_1^*, x_2^*) = 0.0 \]
Comparison between the Golden Section Search Method and Descent Condition Method - Example #6 (2/3)

Global and Local minimum point of the Rastrigins’s Function

A : Global Optimum

\[ x_1 = x_2 = 0.0, \ f = 0.0 \]
**Minimize:**
\[ f(x_1, x_2) = 20 + x_1^2 - 10 \cos(2\pi \cdot x_1) + x_2^2 - 10 \cos(2\pi \cdot x_2) \]

**Subject to:**
\[
\begin{align*}
&g_1(x_1, x_2) = -5.12 - x_1 \leq 0 \\
&g_2(x_1, x_2) = -5.12 - x_2 \leq 0 \\
&g_3(x_1, x_2) = x_1 - 5.12 \leq 0 \\
&g_4(x_1, x_2) = x_2 - 5.12 \leq 0
\end{align*}
\]

In this example, since there are some local minimum design points, the optimal solution to be obtained is changed depending on the initial design point. So, the calculating the optimal solutions by assuming the initial design point in many times and comparing the results are needed.

<table>
<thead>
<tr>
<th>Initial Value</th>
<th>Method</th>
<th>Iteration of defining the QP problem</th>
<th>Iteration of one dimensional search method</th>
<th>Local Optimum Point</th>
<th>Optimum Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>(0.1, 0.1)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>18</td>
<td>147</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>18</td>
<td>147</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>9</td>
<td>82</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>39</td>
<td>427</td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>1</td>
<td>47</td>
<td></td>
<td>(0.0, 0.0)</td>
</tr>
<tr>
<td>(2.1, 2.1)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>16</td>
<td>134</td>
<td>(1.990, 1.990)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>16</td>
<td>134</td>
<td>(1.990, 1.990)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>7</td>
<td>69</td>
<td>(1.990, 1.990)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>32</td>
<td>358</td>
<td>(1.990, 1.990)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>1</td>
<td>45</td>
<td></td>
<td>(1.990, 1.990)</td>
</tr>
<tr>
<td>(-2.1, -3)</td>
<td>Descent condition method</td>
<td>r = 0.0</td>
<td>18</td>
<td>144</td>
<td>(-1.990, -2.985)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.1</td>
<td>18</td>
<td>144</td>
<td>(-1.990, -2.985)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.5</td>
<td>9</td>
<td>82</td>
<td>(-1.990, -2.985)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>r = 0.9</td>
<td>36</td>
<td>395</td>
<td>(-1.990, -2.985)</td>
</tr>
<tr>
<td></td>
<td>Golden section search method</td>
<td>7</td>
<td>229</td>
<td></td>
<td>(-1.990, -2.985)</td>
</tr>
</tbody>
</table>
### Comparison between the Golden Section Search Method and Descent Condition Method

<table>
<thead>
<tr>
<th>Step of calculation</th>
<th>Descent Condition method</th>
<th>Golden Section Search method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Iteration number of one dimensional search method</td>
<td>Little</td>
<td>Many</td>
</tr>
</tbody>
</table>

- Comparison between the Golden Section Search Method and Descent Condition Method
  - When we use the one dimensional search method, we have to calculate the value of the objective function and constraints repetitively.
  - If it takes much time to calculate the value of the objective function and constraints, the Descent condition method is more useful.
Reference Slides
[Reference] Taylor Series Expansion for the Function of Two Variables

The second-order Taylor series expansion of \( f(x_1, x_2) \) at \( (x_1^*, x_2^*) \)
\[
f(x_1, x_2) = f(x_1^*, x_2^*) + \frac{\partial f(x_1^*, x_2^*)}{\partial x_1}(x_1 - x_1^*) + \frac{\partial f(x_1^*, x_2^*)}{\partial x_2}(x_2 - x_2^*)
\]
\[
+ \frac{1}{2} \left( \frac{\partial^2 f(x_1^*, x_2^*)}{\partial x_1^2}(x_1 - x_1^*)^2 + 2 \frac{\partial^2 f(x_1^*, x_2^*)}{\partial x_1 \partial x_2}(x_1 - x_1^*)(x_2 - x_2^*) + \frac{\partial^2 f(x_1^*, x_2^*)}{\partial x_2^2}(x_2 - x_2^*)^2 \right)
\]

Define: \( c = \nabla f(x^*) \), \( d = (x - x^*) \)

\( f(x) = f(x^*) + \nabla f(x^*)^T (x - x^*) + \frac{1}{2} (x - x^*)^{\top} H(x^*) (x - x^*) = f(x^*) + c^T d + \frac{1}{2} d^T H(x^*) d \)  

\[ \tag{2} \]