Methods for Constrained Optimization

Dr. José Ernesto Rayas-Sánchez

Outline

- Constrained optimization problem
- Box constraints
- Methods for constrained optimization problems
- Elimination of variables
- Penalty methods
- Sequential quadratic programming (SQP)
- Minimax formulations
Constrained Optimization Problem

Standard form:

\[ x^* = \arg \min_x u(x) \]
subject to
\[ h(x) = 0 \]
\[ g(x) \leq 0 \]
\[ x^b \leq x \leq x^{ub} \]

- \( x, x^b, x^{ub} \in \mathbb{R}^n \)
- \( u: \mathbb{R}^n \rightarrow \mathbb{R}, h: \mathbb{R}^n \rightarrow \mathbb{R}^E, g: \mathbb{R}^n \rightarrow \mathbb{R}^I \)
- It is assumed \( n > E \)

Constrained Optimization Problem (cont)

\[ x^* = \arg \min_x u(x) \]
subject to
\[ h(x) = 0 \]
\[ g(x) \leq 0 \]
\[ x^b \leq x \leq x^{ub} \]

It is generally assumed that satisfying all the constraints is more important than minimizing \( u(x) \), i.e., feasibility is more important than optimality.
Constrained Optimization Problem (cont)

\[ x^* = \arg \min_{x \in \Omega} u(x) \]

subject to
\[ h(x) = 0 \]
\[ g(x) \leq 0 \]
\[ x^{lb} \leq x \leq x^{ub} \]

The feasible set:
\[ \Omega = \{ x \in \mathbb{R}^n \mid h(x) = 0 \land g(x) \leq 0 \land x^{lb} \leq x \leq x^{ub} \} \]
Dealing with Box Constraints

- Box constraints can be treated as inequality constraints

\[ x^\text{lb} \leq x \leq x^\text{ub} \rightarrow \]
\[ g_1(x) = x_1 - x_1^\text{ub} \leq 0 \]
\[ g_2(x) = x_1^\text{lb} - x_1 \leq 0 \]
\[ \vdots \]
\[ g_{2n-1}(x) = x_n - x_n^\text{ub} \leq 0 \]
\[ g_{2n}(x) = x_n^\text{lb} - x_n \leq 0 \]

- They can also be considered by restricting the optimization space (through variable transformations)

Box Constraints – Restricting Optimization Space

- Box constraints can be incorporated into an unconstrained optimization problem by transforming the optimization variables

\[ x^* = \arg \min_u u(x) \]
subject to
\[ x^\text{lb} \leq x \leq x^\text{ub} \]

we solve
\[ z^* = \arg \min_z u(z) \]

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_j \geq 0 )</td>
<td>( x_j = z_j^2 )</td>
</tr>
<tr>
<td>( x_j &gt; 0 )</td>
<td>( x_j = e^{z_j} )</td>
</tr>
<tr>
<td>( x_j \geq x_j^\text{lb} )</td>
<td>( x_j = x_j^\text{lb} + z_j^2 )</td>
</tr>
<tr>
<td>( x_j &gt; x_j^\text{lb} )</td>
<td>( x_j = x_j^\text{lb} + e^{z_j} )</td>
</tr>
<tr>
<td>(-1 \leq x_j \leq 1)</td>
<td>( x_j = \sin z_j )</td>
</tr>
<tr>
<td>( 0 \leq x_j \leq 1)</td>
<td>( x_j = (\sin z_j)^2 )</td>
</tr>
<tr>
<td>( 0 &lt; x_j &lt; 1)</td>
<td>( x_j = \frac{e^{z_j}}{1 + e^{z_j}} )</td>
</tr>
</tbody>
</table>

(Bandler, 1997)
Box Constraints – Restricting Opt. Space (cont)

<table>
<thead>
<tr>
<th>Constraint</th>
<th>Transformation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_i^{lb} \leq x_i \leq x_i^{ub}$</td>
<td>$x_i = x_i^{lb} + (x_i^{ub} - x_i^{lb})(\sin z_i)^2$</td>
</tr>
<tr>
<td></td>
<td>$x_i = \frac{1}{2}(x_i^{lb} + x_i^{ub}) + \frac{1}{2}(x_i^{ub} - x_i^{lb})\sin z_i$</td>
</tr>
<tr>
<td>$x_i^{lb} &lt; x_i &lt; x_i^{ub}$</td>
<td>$x_i = x_i^{lb} + (x_i^{ub} - x_i^{lb})\left(\frac{e^z_i}{1+e^z_i}\right)$</td>
</tr>
</tbody>
</table>

(Bandler, 1997)

Methods for Constrained Optimization

- Indirect methods (or Sequential Unconstrained Minimization Techniques, SUMT):
  - Elimination of variables (equality constraints)
  - Exterior penalty function method (EPF)
  - Augmented Lagrange multiplier method (ALM)

- Direct methods:
  - Sequential linear programming (SLP)
  - Sequential quadratic programming (SQP)
  - Generalized reduced gradient method (GRG)
  - Sequential gradient restoration algorithm (SGRA)
### Equality Constraints – Elimination of Variables

- When solving
  \[ x^* = \arg \min_x u(x) \]
  subject to
  \[ h(x) = 0 \]
we can reduce the number of equality constraints by eliminating some of the optimization variables.

- If sufficient variables are eliminated, we can obtain an unconstrained optimization problem.

- This technique must be carefully used (the resultant problem can be ill-conditioned).

### Elimination of Variables – Example 1 😊

\[
\begin{align*}
\text{min } & \quad x_1^2 + x_2^2 \\
\text{subject to } & \quad x_1 + x_2 - 1 = 0
\end{align*}
\]

\[ x^* = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \]

\[
\begin{align*}
\text{min } & \quad x_1^2 + (1 - x_1)^2 \\
\end{align*}
\]
Elimination of Variables – Example 2

\[
\begin{align*}
\min & \quad x_1^2 + x_2^2 \\
\text{subject to} & \quad (x_1 - 1)^3 - x_2^2 = 0 \quad x^* = \begin{bmatrix} 1 \\ 0 \end{bmatrix}
\end{align*}
\]

Minimizer is unbounded

Equality Constraints – Penalty Functions

Instead of solving
\[
\begin{align*}
x^* &= \arg\min_{x} u(x) \\
\text{subject to} & \quad h(x) = 0
\end{align*}
\]

we solve
\[
\begin{align*}
x^* &= \arg\min_{x} U(x) \\
\text{where} & \quad U(x) = u(x) + r^h \| h(x) \|_2^2
\end{align*}
\]

\[\| h(x) \|_2^2 : \text{penalty function} \]
\[r^h \in \mathbb{R} : \text{penalty coefficient} \]
Inequality Constraints – Penalty Functions

Instead of solving
\[ x^* = \arg \min_x u(x) \]
subject to
\[ g(x) \leq 0 \]
we solve
\[ x^* = \arg \min_x U(x) \]
where
\[ U(x) = u(x) + r^g \left\| G(x) \right\|_2^2 \]
\[ G_j = \max \{ 0, g_j(x) \} \]
\[ \left\| G(x) \right\|_2^2 : \text{penalty function} \]
\[ r^g \in \mathbb{R} : \text{penalty coefficient} \]

Exterior Penalty Function (EPF) Method

Original problem:
\[ x^* = \arg \min_x u(x) \]
subject to
\[ h(x) = 0 \]
\[ g(x) \leq 0 \]
\[ x_l^b \leq x \leq x_u^b \]
Indirect solution through EPF method:
\[ z^* = \arg \min_z u(z) + p(z, r^h, r^g) \]
\[ p(z, r^h, r^g) = r^h \left\| h(z) \right\|_2^2 + r^g \left\| G(z) \right\|_2^2 \]
\[ G_j = \max \{ 0, g_j(z) \} \]
\[ x_j = x_j^l + (x_j^u - x_j^l)(\sin z_i)^2 \]
\[ j = 1, 2, \ldots, I \quad i = 1, 2, \ldots, n \]
Exterior Penalty Function (EPF) Method (cont)

- The optimal solution \( z^* = \arg \min_z u(z) + p(z, r^h, r^g) \) is a function of the penalty coefficients \( r^h \) and \( r^g \).
- Penalty coefficients should be gradually increased until all constraints are satisfied (exterior method).
- The EPF method is very sensitive to the initial values of the penalty coefficients, \( r^h \) and \( r^g \).

An EPF Algorithm

\[
x^* = \text{EPF}(u, x_0, h, g)
\]

\( u: \mathbb{R}^n \rightarrow \mathbb{R} ; h: \mathbb{R}^n \rightarrow \mathbb{R}^E ; g: \mathbb{R}^n \rightarrow \mathbb{R}^I ; \\
x_0, x^* \in \mathbb{R}^n
\]

\[
\begin{align*}
\text{begin} & \quad \text{set} \ c^h, c^g, \text{and } \varepsilon \\
& \quad r^h_0 = \frac{|u(x_0)|}{\|h(x_0)\|_2 + \varepsilon} ; \quad r^g_0 = \frac{|u(x_0)|}{\|g(x_0)\|_2 + \varepsilon} ; \quad i = 0 \\
& \quad \text{repeat until StoppingCriteria} \\
& \quad x_{i+1} = \arg \min_x u(x) + r^h_i \sum_{k=1}^E h_k^2(x) + r^g_i \sum_{k=1}^J (\max \{0, g_k(x)\})^2 \\
& \quad r^h_{i+1} = c^h r^h_i \\
& \quad r^g_{i+1} = c^g r^g_i \\
& \quad i = i + 1 \\
\text{end} & \quad x^* = x_i \\
\text{end}
\]
Sequential Quadratic Programming (SQP)

- SQP are considered the state-of-the-art in nonlinear programming
- At each iteration, the objective function is approximated by a quadratic function, and the nonlinear constraints are approximated by linear constraints
- The quadratic sub-problem is solved to find a search direction at the current iterate
- The next iterate is obtained from a line search

SQP Sub-problem

At the current iterate \( x_i \),

- The objective function is expanded quadratically
  \[
  u^{(i)}(d) = u(x_i) + d^T \nabla u(x_i) + \frac{1}{2} d^T H(u(x_i)) d \approx u(x_i + d)
  \]

- The constraints are expanded linearly
  \[
  h^{(i)}(x) = h(x_i) + J(h(x_i))d \approx h(x_i + d)
  \]
  \[
  g^{(i)}(x) = g(x_i) + J(g(x_i))d \approx g(x_i + d)
  \]
SQP Sub-problem (cont)

At the current iterate \( x_i \),

- The search direction \( d_i \) is found by solving
  \[
  d_i = \arg \min_d \quad d^T \nabla u(x_i) + \frac{1}{2} d^T H(u(x_i)) d
  \]
  subject to
  \[
  h(x_i) + J(h(x_i)) d = 0 \\
  g(x_i) + J(g(x_i)) d \leq 0 \\
  x^{lb} \leq x \leq x^{ub}
  \]

- The next iterate is calculated using \( x_{i+1} = x_i + \alpha^* d_i \)
  where \( \alpha^* \) is obtained from a line search on \( u(x_i + \alpha d_i) \)

Minimax Formulations

- Minimax formulations are used to minimize the maximum error of a function (model response) with respect to a number of specifications

- A minimax formulation can be implemented as a constrained or as an unconstrained optimization problem
Minimax Formulations – Unconstrained

\[ x^* = \arg \min_{\mathbf{x}} \max \{ \ldots e_k(\mathbf{x}) \ldots \} \]

where

\[ e_k(\mathbf{x}) = \begin{cases} R_k(\mathbf{x}) - S_{k}^{\text{ub}} & \text{for all } k \in I^{\text{ub}} \smallbreak S_{k}^{\text{lb}} - R_k(\mathbf{x}) & \text{for all } k \in I^{\text{lb}} \end{cases} \]

- \( R_k(\mathbf{x}) \) is the \( k \)-th model response at point \( \mathbf{x} \)
- \( S_{k}^{\text{ub}} \) and \( S_{k}^{\text{lb}} \) are the upper and lower bound specifications
- \( I^{\text{ub}} \) and \( I^{\text{lb}} \) are index sets (not necessarily disjoint)
- Vector \( \mathbf{e}(\mathbf{x}) \) contains all the error functions with respect to the design specifications

Minimax Formulations – Unconstrained (cont)

\[ x^* = \arg \min_{\mathbf{x}} U(\mathbf{x}) \]

\[ U(\mathbf{x}) = \max \{ \ldots e_k(\mathbf{x}) \ldots \} \]

where

\[ e_k(\mathbf{x}) = \begin{cases} R_k(\mathbf{x}) - S_{k}^{\text{ub}} & \text{for all } k \in I^{\text{ub}} \smallbreak S_{k}^{\text{lb}} - R_k(\mathbf{x}) & \text{for all } k \in I^{\text{lb}} \end{cases} \]
Minimax Formulations – Unconstrained (cont)

\[ x^* = \arg\min_{\mathbf{x}} U(\mathbf{x}) \]

\[ U(\mathbf{x}) = \max\{ \ldots e_k(\mathbf{x}) \ldots \} \]

Equality specifications \( S_{k_{eq}} \) can be implemented as a combination of upper and lower specifications (\( \Delta S_{eq} > 0 \))

\[
e_k(\mathbf{x}) =\begin{cases} R_k(\mathbf{x}) - (S_{k_{eq}} + \Delta S_{eq}) & \text{for all } k \in I_{eq} \\ (S_{k_{eq}} - \Delta S_{eq}) - R_k(\mathbf{x}) & \text{for all } k \in I_{eq} \end{cases}
\]

or as a single error function

\[
e_k(\mathbf{x}) = |R_k(\mathbf{x}) - S_{k_{eq}}| - \Delta S_{eq} \quad \text{for all } k \in I_{eq}
\]

Minimax Formulations – Unconstrained (cont)

\[ x^* = \arg\min_{\mathbf{x}} \max\{ \ldots e_k(\mathbf{x}) \ldots \} \]

where

\[
e_k(\mathbf{x}) =\begin{cases} R_k(\mathbf{x}) - S_{k_{ub}} & \text{for all } k \in I_{ub} \\ S_{k_{lb}} - R_k(\mathbf{x}) & \text{for all } k \in I_{lb} \\ |R_k(\mathbf{x}) - S_{k_{eq}}| - \Delta S_{eq} & \text{for all } k \in I_{eq} \end{cases}
\]

- \( R_k(\mathbf{x}) \) is the \( k \)-th model response at point \( \mathbf{x} \)
- \( S_{k_{ub}} \) and \( S_{k_{lb}} \) are the upper and lower bound specifications
- \( S_{k_{eq}} \) is an equality specification (\( \pm \Delta S_{eq} \))
- \( I_{ub} \) and \( I_{lb} \) are index sets (not necessarily disjoint)
Minimax Formulations – Relative Errors

- Formulation

\[ e_k(x) = \begin{cases} 
R_k(x) - S_k^{ub} & \text{for all } k \in I^{ub} \\
S_k^{ub} - R_k(x) & \text{for all } k \in I^{lb} \\
\left|R_k(x) - S_k^{eq}\right| - \Delta S^{eq} & \text{for all } k \in I^{eq}
\end{cases} \]

may require the usage of weighting factors

- We can use instead a relative formulation for the error functions

Minimax Formulations – Relative Errors (cont)

- Using relative error functions (assuming \( S_k^{ub} > 0 \) and \( S_k^{lb} > 0 \))

\[ e_k(x) = \begin{cases} 
\frac{R_k(x)}{S_k^{ub} + \varepsilon} - 1 & \text{for all } k \in I^{ub} \\
1 - \frac{R_k(x)}{S_k^{lb} + \varepsilon} & \text{for all } k \in I^{lb} \\
\frac{\left|R_k(x) - S_k^{eq}\right|}{\varepsilon} - 1 & \text{for all } k \in I^{eq}
\end{cases} \]

where \( \varepsilon \) is an arbitrary small positive number
Minimax Formulations – Constrained

- We define an additional optimization variable (ceiling)
  \[ x^* = \arg \min_{\mathbf{x}} x_{n+1} \]
  subject to
  \[ e_k(x) - x_{n+1} \leq 0 \]
  where
  \[ e_k(x) = \begin{cases} R_k(x) - S^\text{ub}_k & \text{for } k \in I^\text{ub} \\ S^\text{lb}_k - R_k(x) & \text{for } k \in I^\text{lb} \end{cases} \]

- \( R_k(x) \) is the \( k \)-th model response at point \( x \)
- \( S^\text{ub}_k \) and \( S^\text{lb}_k \) are the upper and lower bound specifications
- \( I^\text{ub} \) and \( I^\text{lb} \) are index sets (not necessarily disjoint)