Optimum Design: Numerical Solution Process

- Introduction to numerical search methods
- Optimum design: aspects of problem formulation
- Numerical solution process for optimum design
- EXCEL: Solver
- MATLAB: Optimization Toolbox
- Mathematica: Optimization Toolbox

Numerical Search Methods (1)

- Graphical method
  - Two-variable problems only
- Approach to solve optimality conditions
  - difficult to use when the number of variables and/or the number of constraints is greater than three
  - leads to a set of nonlinear equations that needs to be solved using a numerical method anyway
- Numerical methods
  - can handle many variables and constraints, as well as directly search for optimum points
  - start with an initial design estimate
  - search the feasible set for optimum designs
  - Derivative-Based Methods / Direct Search Methods / Derivative-Free Methods / Nature-Inspired Search Methods

Numerical Search Methods (2)

- Derivative-Based Methods: gradient-based search methods (Ch.10~13)
  - all functions: continuous and at least twice continuously differentiable
  - Accurate first-order derivatives of all the functions are available
  - design variables: assumed to be continuous within their allowable range
  - extensively developed since the 1950s, and many good ones are available to solve smooth nonlinear optimization problems
  - always converge to a local minimum point only, global solutions?

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} + \Delta\mathbf{x}^{(k)};\ k = 0,1,2,\ldots \\ \text{where} \begin{cases} \mathbf{x}^{(0)}:\ \text{starting design point} \\ \Delta\mathbf{x}^{(k)} = \alpha_k \mathbf{d}^{(k)} \xrightarrow{\text{two separate subproblems}} \begin{cases} \alpha_k > 0:\ \text{step size} \\ \mathbf{d}^{(k)}:\ \text{search direction} \end{cases} \end{cases} \\ x_i^{(k+1)} = x_i^{(k)} + \Delta x_i^{(k)};\ k = 0,1,2,\ldots;\ i = 1,\ldots,n \end{cases}$$

Numerical Search Methods (3)

- Direct Search Methods (Ch.11.9)
  - do not calculate/use/approximate derivatives of the problem functions
  - Functions are assumed to be continuous and differentiable; however, their derivatives are either unavailable or not trustworthy
  - Only functions' values are calculated and used in the search process
  - Hooke–Jeeves and Nelder–Mead in 1960s and 1970s
  - simplicity and ease of use
- Derivative-Free Methods
  - do not require explicit calculation of analytical derivatives of the functions
  - approximation of derivatives is used to construct a local model: finite difference approach
  - *response surface methods* that generate approximation for complex optimization functions

Numerical Search Methods (4)

- Nature-Inspired Search Methods
  - use only the values of the problem functions
  - classified as direct search methods
  - use statistical concepts and random numbers to advance the search toward a solution point
    - simulated annealing (Ch.15.5)
    - genetic algorithms (Ch.17.1)
  - quite general: solve all kinds of optimization problems
  - quite time-consuming: require a large number of function evaluations to reach an acceptable solution
  - do not have a good stopping criterion (no optimality conditions)

Selection of a Method

- Are the design variables continuous (can have any value in their range), discrete (must be selected from a specified list) or integer?
- Are the problem functions continuous and differentiable?
- Are derivatives of all the problem functions available (can be calculated efficiently)?

General Guidelines

- formulation of a design task as an optimization problem
  - Define a realistic model for the engineering system
  - Use designer's engineering knowledge, intuition, and experience
- generate a mathematical optimization model
  - In an initial formulation of the problem, all of the possible parameters should be viewed as potential design variables
  - The *existence of an optimum solution* to a design optimization model depends on its formulation
  - The problem of optimizing more than one objective functions simultaneously (*multi-objective problems*) can be transformed into the standard problem
  - In general, *it is desirable to normalize all of the constraints with respect to their limit values*

Scaling of Constraints

- In numerical calculations, it is impossible to require
  - an equality constraint to be precisely equal to zero
  - an active inequality constraint to be precisely equal to zero

$$h(\mathbf{x}) = 0 \rightarrow \left|h(\mathbf{x})\right| \leq \varepsilon$$

$$g(\mathbf{x}) \leq 0 \rightarrow \left|g(\mathbf{x})\right| \leq \varepsilon$$

ε : feasibility tolerance

- different constraints can involve different orders of magnitude → *constraint normalization*

$$\sigma \leq \sigma_a \rightarrow g_1 = \sigma - \sigma_a \leq 0 \rightarrow g_1 = \frac{\sigma}{\sigma_a} - 1 \leq 0 \rightarrow g_1 = R - 1 \leq 0$$

$$\delta \leq \delta_a \rightarrow g_2 = \delta - \delta_a \leq 0 \rightarrow g_2 = \frac{\delta}{\delta_a} - 1 \leq 0 \rightarrow g_2 = R - 1 \leq 0$$

- some constraints that cannot be normalized: 0 ≤ x

Constraint Normalization

$$\left.\begin{aligned} &h(x_1,x_2)=x_1^{\,2}+\frac{1}{2}x_2-18=0\\ &\left|h(4.0,4.2)\right|=\left|+0.1\right|>\varepsilon(0.01)\\ &\left|h(-4.5,-4.8)\right|=\left|-0.15\right|>\varepsilon(0.01)\end{aligned}\right\}\rightarrow\begin{cases}\bar{h}(x_1,x_2)=\dfrac{1}{18}x_1^{\,2}+\dfrac{1}{36}x_2-1=0\\ \left|\bar{h}(4.0,4.2)\right|=\left|+0.0056\right|<\varepsilon(0.01)\\ \left|\bar{h}(-4.5,-4.8)\right|=\left|-0.0083\right|<\varepsilon(0.01)\end{cases}$$

$$\frac{\partial h}{\partial x_1}=2x_1\rightarrow\frac{\partial \bar{h}}{\partial x_1}=\frac{1}{18}(2x_1)$$

$$g(x_1,x_2)=500x_1-30,000x_2\le 0\rightarrow\begin{cases}\xrightarrow[30,000|x_2|]{\text{divided by}}\bar{g}(x_1,x_2)=\dfrac{x_1}{60|x_2|}-\dfrac{x_2}{|x_2|}\le 0\\ \bar{g}(x_1,x_2)=\dfrac{1}{60}x_1-x_2\le 0\end{cases}$$

$$\left.\begin{aligned} &g(80,1)=10,000>\varepsilon(0.01)\\ &g(60,0.995)=150>\varepsilon(0.01)\end{aligned}\right\}\rightarrow\begin{cases}\bar{g}(80,1)=0.33>\varepsilon(0.01)\\ \bar{g}(60,0.995)=0.005>\varepsilon(0.01)\end{cases}$$

Scaling of Design Variables

$$(1)\ a \le x \le b \leftrightarrow -1 \le y \le 1$$

$$(b-a):2=(x-a):(y+1)$$

$$(2)\ a \le x \le b \leftrightarrow \frac{a}{b} \le \frac{x}{b} \le 1 \xrightarrow{y=x/b} \frac{a}{b} \le y \le 1$$

$$(3)\ a \le x \le b \xleftrightarrow[y=1]{x=\frac{a+b}{2}} \frac{a}{\frac{a+b}{2}} \le \frac{x}{\frac{a+b}{2}} \le \frac{b}{\frac{a+b}{2}} \xleftrightarrow{y=\frac{2x}{a+b}} \frac{2a}{a+b} \le y \le \frac{2b}{a+b}$$

$$(4)\ \begin{cases} x_1 \sim O\left(10^5\right) \rightarrow \dfrac{x_1}{10^5} = y_1 \\ x_2 \sim O\left(10^{-5}\right) \rightarrow \dfrac{x_2}{10^{-5}} = y_2 \end{cases}$$

$$\frac{\partial f}{\partial y} = \frac{\partial f}{\partial x}\frac{\partial x}{\partial y}$$

Iterative Process for Development of Problem Formulation

- Many practical applications are complex requiring repeated updating of the initial formulation of the problem
  - some of the practical constraints may have been missed
  - the limits for some of the constraints may not be realistic
  - there may be conflicting constraints in the formulation
  - the constraint limits may be too severe such that there is no feasible solution for the problem

Numerical Solution Process

- optimization algorithm for smooth problems
  - Calculation of cost and constraint functions and their gradients at the current point
  - Definition of a subproblem
    - determine the search direction
    - Step size determination in the search direction
  - Update the current design point

- General purpose software: integration of
  - problem functions
  - gradient evaluation software
  - optimization software

A Feasible Point Cannot Be Obtained

- Check the formulation to ensure that the constraints are formulated properly and that there are no inconsistencies in them
- Scale the constraints if they have different orders of magnitude
- Check the feasibility of individual constraints or a subset of constraints while ignoring the remaining ones
- Ensure that the formulation and data are properly transferred to the optimization software
- The constraint limits may be too severe
- Check the constraint feasibility tolerance
- Check derivation and implementation of the gradients of the constraint functions
- Increase precision of all calculations, if possible

Algorithm Does Not Converge (1)

- Check the formulation to ensure that the constraints and the cost function are formulated properly, Ensure that all of the functions are continuous and differentiable for a smooth optimization algorithm
- Scale the constraints and the cost function if they have different orders of magnitude
- Check implementation of the cost function and the constraint functions evaluations
- Check the derivation and implementation of the gradients of all of the functions, If the gradients are evaluated using finite differences, then their accuracy needs to be verified
- Examine the final point reported by the program
- If an overflow of calculations is reported, the problem may be unbounded

Algorithm Does Not Converge (2)

- Try different starting points
- Ignore some of the constraints and solve the resulting problem
- Use a smaller limit on the number of iterations and restart the algorithm with the final point of the previous run of the program as the starting point
- If two design variables are of differing orders of magnitude, scale them so that the scaled variables have the same order of magnitude
- Ensure that the optimization algorithm has been proven to converge to a local minimum point starting from any initial point
- Increase the precision of the calculations, if possible

EXCEL Solver(해 찾기)

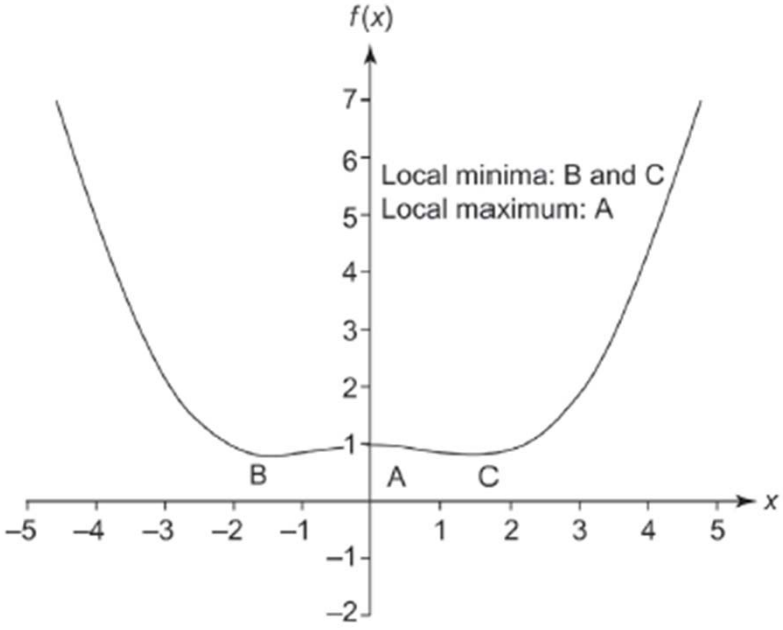
- Introduction
- Roots of a nonlinear equation
- Roots of a set of nonlinear equation
- Unconstrained optimization problems
- Linear programming problems
- Nonlinear programming
  - Optimum design of spring
  - Optimum design of plate girders

Example 4.22

- Numerical solution for the first-order necessary conditions

$$f(x)=\frac{1}{3}x^2+\cos x \rightarrow f'(x)=\frac{2}{3}x-\sin x=0$$

$$\begin{cases} x^*=0,\ f(0)=1 \\ x^*=1.496,\ f(1.496)=0.821 \\ x^*=-1.496,\ f(1.496)=0.821 \end{cases}$$



Example 4.31

- Solution of the KKT necessary conditions

$$\underset{\mathbf{x}}{\text{Minimize}} \quad f(\mathbf{x}) = x_1^2 + x_2^2 - 3x_1x_2$$

$$\text{subject to} \quad g = x_1^2 + x_2^2 - 6 \le 0$$

$$L = x_1^2 + x_2^2 - 3x_1x_2 + u\left(x_1^2 + x_2^2 - 6 + s^2\right)$$

$$\begin{cases} \dfrac{\partial L}{\partial x_1} = 2x_1 - 3x_2 + 2ux_1 = 0 \\ \dfrac{\partial L}{\partial x_2} = 2x_2 - 3x_1 + 2ux_2 = 0 \\ \dfrac{\partial L}{\partial u} = x_1^2 + x_2^2 - 6 + s^2 = 0 \\ \dfrac{\partial L}{\partial s} = 2us = 0; \qquad u \ge 0 \end{cases}$$

$$\rightarrow \begin{cases} x_1^* = x_2^* = 0, \;\; u^* = 0, \; s = 6, \;\; f = 0 \\ x_1^* = x_2^* = \sqrt{3}, \;\; u^* = 1/2, \;\; s = 0, \; f = -3 \\ x_1^* = x_2^* = -\sqrt{3}, \;\; u^* = 1/2, \;\; s = 0, \; f = -3 \end{cases}$$