4M020 Design tools
Algorithms for numerical optimization

L.F.P. Etman

Department of Mechanical Engineering
Eindhoven University of Technology

Wednesday September 3, 2008
Outline

1. Problem formulation: classes and properties
2. Optimization algorithms
The mathematical problem formulation is to the heart of the success of design optimization
Two-bar truss example

\[
\begin{align*}
\min_{x} & \quad f(x) = C_0 \ x_2^2 \sqrt{S^2 + x_1^2} \\
\text{s.t.} & \quad g_1(x) = C_1 x_1 - 1 \leq 0 \\
& \quad g_2(x) = C_2 \frac{\sqrt{S^2 + x_1^2}}{x_2 x_1} - 1 \leq 0 \\
& \quad g_3(x) = C_3 \left( \frac{S^2 + x_1^2}{x_2^4 x_1} \right)^{3/2} - 1 \leq 0 \\
\chi & : x_1, x_2 > 0,
\end{align*}
\]

How can we solve this optimization problem?
Mathematical problem formulation

Minimize \( f(x) \)
subject to \( h_j(x) = 0 \)
\( g_k(x) \leq 0 \)
\( x \in \mathcal{X} \subseteq \mathbb{R}^n \)

\( x = \) (column) vector of design variables
\( j = 1, \ldots, m_h \)
\( k = 1, \ldots, m_g \)

[Papalambros & Wilde 2000: Principles of optimal design]
Formulation examples

Design variables:

- sizing (dimensions)
- shape (geometry of boundary)
- topological (material distribution)
Formulation examples

Design variables $x$:
- sizing (dimensions)
- shape (geometry of boundary)
- topological (material distribution)

Objective function $f(x)$:
- profit (cost, efficiency, weight, ...)

Constraint functions $h_j(x)$ and $g_k(x)$:
- geometrical (width, length, height, ...)
- structural (stresses, displacements, ...)
- dynamical (accelerations, eigenfrequency, ...)
- physical (temperatures, pressures, ...)

Formulation classes  Optimization algorithms
Formulation classes

Design variables:
• continuous or discrete
• single-variable or multi-variable

Objective function:
• minimization or maximization
• single-objective or multi-objective

Constraint functions:
• unconstrained or constrained
• equality or inequality
Formulation classes: linear programming

Linear programming (LP) problem:
- Linear objective function
- Linear constraint functions
- Continuous design variables

Example:

\[
\begin{align*}
\min_{x \in \mathbb{R}^2} & \quad f(x) = -2x_1 - x_2 \\
\text{s.t.} & \quad g_1(x) = x_1 + 2x_2 - 8 \leq 0 \\
& \quad g_2(x) = 2x_1 - 2x_2 - 3 \leq 0 \\
& \quad g_3(x) = -2x_1 + 1 \leq 0 \\
& \quad g_4(x) = -2x_2 + 1 \leq 0
\end{align*}
\]

(Some) design variables discrete:
- (Mixed-)integer linear programming problem: (M)ILP
Formulation classes: quadratic programming

Quadratic programming (QP) problem:
- Quadratic objective function
- Linear constraint functions
- Continuous design variables

Example:

$$\min_{x \in \mathbb{R}^2} f(x) = 3x_1^2 - 2x_1 + 5x_2^2 + 30x_2$$

s.t.  $$g_1(x) = -2x_1 - 3x_2 + 8 \leq 0$$
       $$g_2(x) = 3x_1 + 2x_2 - 15 \leq 0$$
       $$g_3(x) = x_2 - 5 \leq 0$$

(Some) design variables discrete:
- (Mixed-)integer quadratic programming problem: (M)IQP
Formulation classes: nonlinear programming

Nonlinear programming (NLP) problem:
- Non-linear objective function
- Non-linear constraint functions
- Continuous design variables

Example:

\[
\min_{x \in \mathbb{R}^2} f(x) = 3x_1 + \sqrt{3}x_2 \\
\text{s.t. } g_1(x) = \frac{18}{x_1} + \frac{6\sqrt{3}}{x_2} - 3 \leq 0 \\
\quad g_2(x) = 5.73 - x_1 \leq 0 \\
\quad g_3(x) = 7.17 - x_2 \leq 0
\]

(Some) design variables discrete:
- (Mixed-)integer nonlinear programming problem: (M)INLP
Design variables:
- Domain: \(\{0, 1\}, \mathcal{N}, \mathcal{N}^+, \mathcal{Z}, \mathcal{R}, \mathcal{R}^+\)

Objective and constraint functions:
- Linearity: linear or nonlinear
- Continuity: none/once/twice-differentiable

Optimization problem:
- Modality: uni-modal or multi-modal
Two-bar truss example

$$\min_x f(x) = C_0 \ x_2^2 \sqrt{S^2 + x_1^2}$$

s.t. $$g_1(x) = C_1 x_1 - 1 \leq 0$$

$$g_2(x) = C_2 \frac{\sqrt{S^2 + x_1^2}}{x_2^2 x_1} - 1 \leq 0$$

$$g_3(x) = C_3 \left( \frac{S^2 + x_1^2}{x_2^4 x_1} \right)^{3/2} - 1 \leq 0$$

$$\chi : x_1, x_2 > 0,$$

Exercise 1: introduction to the Matlab algorithm \texttt{fmincon}
Select an optimization algorithm in accordance with the optimization problem class and properties
Searching for a zero

Solving a set of $m$ nonlinear equations with $m$ unknowns:

$$\Phi(q) = 0$$

Example:

$$1 + e^{-q_2} - q_2^2 e^{-q_1} = 0$$
$$1 - q_1 e^{-q_2} + 2q_2 e^{-q_1} = 0$$
A function $f(x)$ with $n$ variables is defined as:

$$
\text{Minimize } f(x) = x_1 + x_2 + x_1 e^{-x_2} + x_2^2 e^{-x_1}
$$
Finding the solution to a set of nonlinear equations:

\[ \Phi(q) = 0 \]

can be reformulated as a minimization problem:

Minimize \( e^T e \quad q \)

with the error (deviations to zero) defined as:

\[ e = \Phi(q) \]
Some optimization problems can be *analytically* solved:

$$\text{Min } f(x) = 2x_1 + x_1^{-2} + 2x_2 + x_2^{-2}$$

Many optimization problems can only be *numerically* solved:

$$\text{Min } f(x) = x_1 + x_2 + x_1 e^{-x_2} + x_2^2 e^{-x_1}$$

An optimization algorithm is an *iterative* procedure to solve an unconstrained or constrained minimization (maximization) problem.

[Papalambros & Wilde 2000: Principles of optimal design]
Classification of optimization algorithms

Four classifiers of an optimization algorithm:

- unconstrained / constrained search
- local / global search
- deterministic / stochastic search
- 0th / 1st / 2nd-order search
Unconstrained / constrained search

Unconstrained search:

- aims to minimize a nonlinear, possibly multi-modal, objective function in n-dimensional space

Constrained search:

- aims to minimize a (non)linear objective function in n-dimensional space while accounting for (non)linear equality and/or inequality constraint functions
Local search:

- seeks improvement based on the Taylor series expansion;

in a sufficiently small region any nonlinear function can be represented by a quadratic approximation:

\[
\tilde{f}(x) = f(x^*) + \sum_{i=1}^{n} \frac{\partial f(x^*)}{\partial x_i} (x_i - x_i^*) + \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} \frac{\partial^2 f(x^*)}{\partial x_i \partial x_j} (x_i - x_i^*)^2
\]

Global search:

- seeks improvement by design space exploration
Deterministic / stochastic search

Deterministic search:
• gives exactly the same search path when running the algorithm twice for unchanged algorithmic settings

Stochastic search:
• generates random search paths for every run of the algorithm
A $0^{th}$-order algorithm
  - uses function evaluations only

A $1^{th}$-order algorithm
  - uses function and gradient evaluations

A $2^{nd}$-order algorithm
  - uses function, gradient, and Hessian evaluations
Newton type of algorithms

Characteristics

- unconstrained/constrained
- local
- deterministic
- 2\textsuperscript{nd}-order

Matlab Optimization toolbox

- fminunc
- fmincon

Line search

Trust-region
Bio-inspired algorithms

**Characteristics**
- unconstrained/constrained
- global
- stochastic
- $0^{th}$-order

**Algorithms**
- Particle swarms
- Genetic algorithms (GAs)
- Simulated Annealing

**Matlab GA toolbox**
- `ga`
Direct search algorithms

**Characteristics**
- unconstrained/constrained
- local
- deterministic
- 0\textsuperscript{th}-order

**Algorithms**
- Nelder-Mead simplex search
- Generalized Pattern Search

**Matlab Optimization toolbox**
- fminsearch

**Matlab Direct Search toolbox**
- patternsearch
Formulation classes
Optimization algorithms

Two-bar truss example

\[
\begin{align*}
\text{min}_{x} & \quad f(x) = C_0 \ x_2^2 \sqrt{S^2 + x_1^2} \\
\text{s.t.} & \quad g_1(x) = C_1 x_1 - 1 \leq 0 \\
& \quad g_2(x) = C_2 \frac{\sqrt{S^2 + x_1^2}}{x_2 x_1} - 1 \leq 0 \\
& \quad g_3(x) = C_3 \frac{(S^2 + x_1^2)^{3/2}}{x_2^4 x_1} - 1 \leq 0 \\
\chi & : x_1, x_2 > 0,
\end{align*}
\]

Exercise 2: visualization of the optimization search path
Gradients

Calculating derivatives:

- Analytically (by hand)
- Symbolically (e.g. Mathematica, Maple)
- Finite differencing
- By the CAE analysis code (e.g. MARC)
- Automatic differentiation (e.g. ADIFOR, ADIC)

Matlab:

- \texttt{fmincon} and \texttt{fminunc} calculate gradients by finite differencing if the user does not provide them
- Matlab Symbolic Toolbox
- Matlab Automatic Differentiation Toolbox
Conditions for optimality

Karush-Kuhn-Tucker (KKT) conditions

**Unconstrained:** gradients objective function zero

**Constrained:** linear combination of gradients objective and gradients active constraints zero (i.e. gradients Lagrange function $L$ zero)
Termination criteria

Condition on optimality

\[ \|\nabla L(x_{k+1})\| < \varepsilon \]  

(Matlab: TolFun)

\[ g_j(x_{k+1}) < \varepsilon \quad \text{and} \quad |h_j(x_{k+1})| < \varepsilon \]  

(Matlab: TolCon)

or a condition on change in x

\[ \|x_{k+1} - x_k\| < \varepsilon \]  

(Matlab: TolX)

or a condition on the number of iterations

\[ k \leq k_{\text{max}} \]  

(Matlab: MaxIter)

or a combination of these, with \( \varepsilon > 0 \)
Scale design variables, objective function and constraint functions to avoid numerical difficulties and premature termination
Two-bar truss example

\[
\begin{align*}
\min_{x} & \quad f(x) = m \\
\text{s.t.} & \quad g_1(x) = C_1 x_1 - 1 \leq 0 \\
& \quad g_2(x) = \frac{\sigma(x)}{\sigma_y} - 1 \leq 0 \\
& \quad g_3(x) = \frac{P(x)}{P_c(x)} - 1 \leq 0 \\
\chi : & \quad x_1, x_2 > 0,
\end{align*}
\]

Exercise 3: FEM-model in the optimization loop