Lecture 1: Introduction to Engineering Optimization

Kevin Carlberg

Stanford University

July 27, 2009

Goals

- An introduction to mathematical optimization, which is quite useful for many applications spanning a large number of fields
  - Design (automotive, aerospace, biomechanical)
  - Control
  - Signal processing
  - Communications
  - Circuit design
- Cool and useful applications of the tools learned so far: can we use finite element modeling to design an aircraft or to detect internal damage in a structure?

References

- J. Nocedal and S. J. Wright. *Numerical Optimization*, Springer, 1999.
- S. Boyd and L. Vadenberghe. *Convex Optimization*, Cambridge University Press, 2004.
- P.E. Gill, W. Murray, and M.H. Wright, *Practical Optimization*, London, Academic Press, 1981.
- I. Kroo, J. Alonso, D. Rajnarayan, *Lecture Notes from AA 222: Introduction to Multidisciplinary Design Optimization*, http://adg.stanford.edu/aa222/.

Course information

- **Instructor**: Kevin Carlberg (carlberg@stanford.edu)
- **Lectures**: There will be five lectures covering
  1. Introduction to Engineering Optimization
  2. Unconstrained Optimization
  3. Constrained Optimization
  4. Optimization with PDE constraints
- **Assignments**: There will be a few minor homework and in-class assignments

Why optimization?

- Mathematical optimization: make something the *best* it can possibly be.

$$\begin{array}{rl} \text{maximize} & \text{objective} \\ \text{by choosing} & \text{variables} \\ \text{subject to} & \text{constraints} \end{array}$$

- Are you optimizing right now?

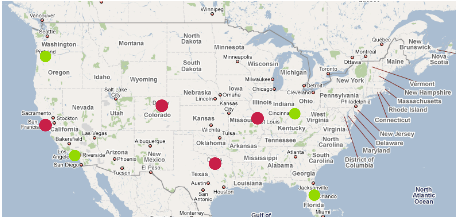
  objective: learning; variables: actions; constraints: physical limitations

- Perhaps more realistically,

  objective: comfort

Applications

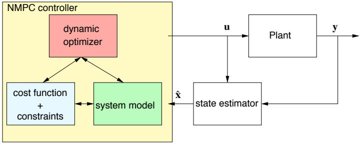
- Physics. Nature chooses the state that minimizes an energy functional (variational principle).
- Transportation problems. Minimize cost by choosing routes to transport goods between warehouses and outlets.



- Portfolio optimization. Minimize risk by choosing allocation of capital among some assets.
- Data fitting. Choose a model that best fits observed data.

Applications *with PDE constraints*

- Design optimization
- Model predictive control Figure from R. Findeisen and F. Allgower, "An Introduction to Nonlinear Model Predictive Control," 21st Benelux Meeting on Systems and Control, 2002.



- Structural damage detection

Brachistochrone Problem History

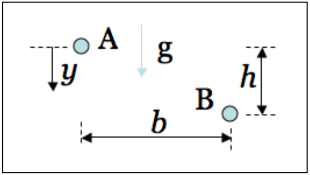
- One of the first problems posed in the calculus of variations.
- Galileo considered the problem in 1638, but his answer was incorrect.
- Johann Bernoulli posed the problem in 1696 to a group of elite mathematicians:

  I, Johann Bernoulli... hope to gain the gratitude of the whole scientific community by placing before the finest mathematicians of our time a problem which will test their methods and the strength of their intellect. If someone communicates to me the solution of the proposed problem, I shall publicly declare him worthy of praise.

- Newton solved the problem the very next day, but proclaimed "I do not love to be dunned [pestered] and teased by foreigners about mathematical things."

Brachistochrone Problem (homework)

- Problem: *Find the frictionless path that minimizes the time for a particle to slide from rest under the influence of gravity between two points A and B separated by vertical height h and horizontal length b.*



- Conservation of energy: $\frac{1}{2}mv^2 + mgh = C$
- Beltrami Identity: for $I(y) = \int_{x_A}^{x_B} f(y(x))dx$, the stationary point solution y^* characterized by $\delta I(y^*) = 0$ satisfies $f - y' \frac{\partial f}{\partial y'} = C$.

Numerical Solution

- Although the analytic solution is available, an approximate solution can be computed using numerical optimization techniques.

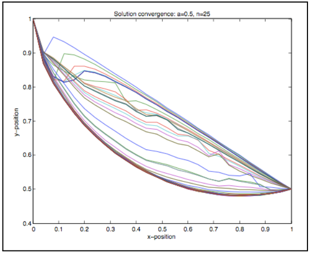
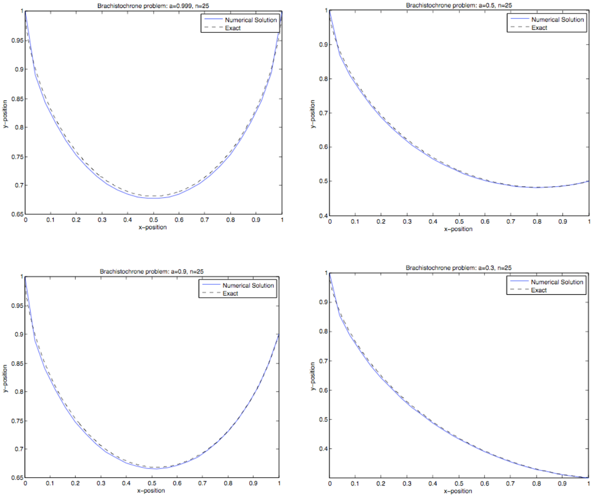


Figure: Evolution of the solution using a gradient-based algorithm

Numerical Solution (for different h)



Mathematical Optimization

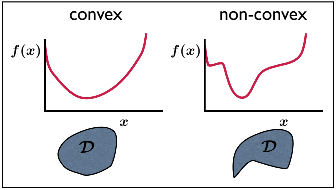
- **Mathematical optimization**: the minimization of a function subject to constraints on the variables. "Standard form":

$$\begin{aligned} \underset{x \in \mathbb{R}^n}{\text{minimize}} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad i = 1, \ldots, n_e \\ & d_j(x) \geq 0, \quad j = 1, \ldots, n_i \end{aligned}$$

  - Variables: $x \in \mathbb{R}^n$
  - Objective function: $f : \mathbb{R}^n \to \mathbb{R}$
  - Equality constraint functions: $c_i : \mathbb{R}^n \to \mathbb{R}$
  - Inequality constraint functions: $d_j : \mathbb{R}^n \to \mathbb{R}$
- **Feasible set**: $\mathcal{D} = \{x \in \mathbb{R}^n \mid c_i(x) = 0,\ d_j(x) \geq 0\}$
- Different optimization algorithms are appropriate for different problem types

Convex v. non-convex

- **Convex problems**: Convex objective and constraint functions: $g(\alpha x + \beta y) \leq \alpha g(x) + \beta g(y)$



  - **LP** (linear programming): linear objective and constraints. Common in management, finance, economics.
  - **QP** (quadratic programming): quadratic objective, linear constraints. Often arise as algorithm subproblems.
- **NLP** (nonlinear programming): the objective or some constraints are general nonlinear functions.
  Common in the physical sciences.

Convex v. non-convex significance

- Convex: a *unique* optimum (local solution=global solution)
- NLP: A global optimum is desired, but can be difficult to find

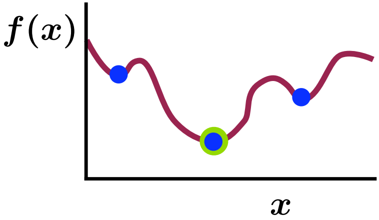


Figure: Local and global solutions for a nonlinear objective function.

- Local optimization algorithms can be used to find the global optimum (from different starting points) for NLPs

Continuous v. discrete optimization

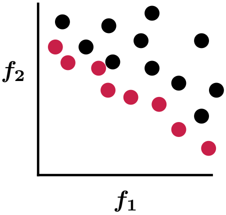
- **Discrete**: The feasible set is finite
  - Always non-convex
  - Many problems are NP-hard
  - Sub-types: combinatorial optimization, integer programming
  - Example: How many warehouses should we build?
- **Continuous**: The feasible set is uncountably infinite
  - Continuous problems are often much easier to solve because derivative information can be exploited
  - Example: How thick should airplane wing skin be?
- Discrete problems are often reformulated as a sequence of continuous problems (e.g. branch and bound methods)

Constrained v. unconstrained

- Unconstrained problems ($n_e = n_i = 0$) are usually easier to solve
- Constrained problems are thus often reformulated as a sequence of unconstrained problems (e.g. penalty methods)

Single-objective v. Multi-objective optimization

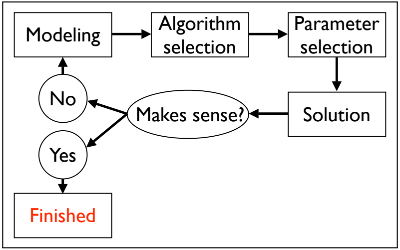
- We may want to optimize two competing objectives f_1 and f_2 (e.g. manufacturing cost and performance)
- **Pareto frontier**: set of candidate solutions among which no solution is better than any other solution in both objectives



- These problems are often solved using evolutionary algorithms

Modeling

- **Modeling**: the process of identifying the objective, variables, and constraints for a given problem



- The more abstract the problem, the more difficult modeling becomes

Example (Homework)

- You live in a house with two other housemates and two vacancies. You are trying to choose two of your twenty mutual friends (who all want to live there) to fill the vacancies.

- Model the problem as a mathematical optimization problem, and categorize the problem as constrained/unconstrained, continuous/discrete, convex/NLP, and single/multi-objective

Rest of the week

- Unconstrained optimization
- Constrained optimization
- PDE-constrained optimization