Optimization in 15 minutes

John Lee
lee@ee.ucla.edu

What is Optimization and Mathematical Programming?

- Optimization is any process of improving
  - Optimize your writing process
  - Optimize your cooking
  - Optimize your PhD

- Mathematical Programming is a sub-field which deals with problems with a specific form:

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 0 \\ & g_j(x) = 0 \end{aligned}$$

Mathematical Programming

- Q: Why is everything in the form:

$$\begin{array}{ll} \text{minimize} & f_0(x) \\ \text{subject to} & f_i(x) \leqslant 0 \\ & g_j(x) = 0 \end{array}$$

  - Convenient form to see the structure of the problem
  - 50 years in convention

- This form covers most design problems, except:
  - o Problems that require a trade-off analysis
  - o Multi-objective problems
  - o (These problems can usually be formulated to a sequence of "minimize" "subject to" problems)

Different types of Optimization Problems

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 0 \\ & g_j(x) = 0 \end{aligned}$$

	domain	$f0$	fi	gi	Optimality condition
Linear Programming	Continuous, convex	linear	linear	linear	x
Geometric Programming	Continuous, convex	posynomial	posynomial	linear	x
Quadratic Programming	Continuous, Convex	Quadratic, convex	linear	linear	x
Convex Programming	Continuous, Convex	convex	convex	linear	x
Nonlinear Programming	General, connected	general	general	general	
Integer Programming	Integer	general	general	general	

Linear Programming

$$\begin{aligned} \text{minimize} \quad & c^T x \\ \text{subject to} \quad & Ax \leqslant b \\ & Gx = h \end{aligned}$$

- Everything is linear above
- Easiest problem to solve (aside from unconstrained):
  - Can solve very large problems (sometimes 1 million +)
  - Simplex methods or interior point methods

- Software:
  - MATLAB linprog, CVX, lp_solve (for large problems)

Quadratic Programming

$$\begin{array}{ll} \text{minimize} & x^T A^T A x + b^T x \\ \text{subject to} & Gx \leqslant h \end{array} \qquad \begin{array}{ll} \text{minimize} & \|Ax - b\|^2 \\ \text{subject to} & Gx \leqslant h \end{array}$$

- Can solve large designs
- Most cases are from fitting problems
  - Fitting data (||Ax-b||^2 is a measure of error)
  - Optimal sensor locations
- Software:
  - CPLEX, LOQO, Mosek, Matlab fmincon

Geometric Programming

$$\begin{aligned} &\text{minimize} && f_0(x) \\ &\text{subject to} && f_i(x) \leqslant 1 \\ & && x \geqslant 0 \end{aligned}$$

- f_0 and f_i are posynomials:

$$f(x) = \sum_k c_k \prod_j x_j^{\alpha_{jk}}, \quad c_k > 0, \ \alpha_{jk} \in \mathbb{R}$$

- A change of variables $x = e^y$ makes the problem a convex optimization problem
- Software:
  - CVX, GGPLab, Mosek,

Examples of Posynomials

- Gate delay: $D_i(x) = x_i^{-1} \sum_{j \in \mathrm{fo(i)}} x_j$
- Area: $A(w, l, h) = wlh$
- Opamp currents: $I_7 = \dfrac{W_5 L_8}{L_5 W_8}$ ⇨ $\begin{cases} W_5 L_8 L_5^{-1} W_8^{-1} I_7^{-1} \leqslant 1 \\ W_5^{-1} L_8^{-1} L_5 W_8 I_7 \leqslant 1 \end{cases}$
- Opamp, biases, communications theory, wire-sizing
  – see Boyd et. al, A Tutorial on Geometric Programming, Hershenson & Boyd, etc.

Integer / Discrete Programming

$$\begin{aligned} \text{minimize} \quad & f_0(x) \\ \text{subject to} \quad & f_i(x) \leqslant 1 \\ & x \in \{0, 1, 2, 3 \ldots\} \end{aligned}$$

- Generally very difficult to solve exactly
  - NP Complete class
- Solvers use heuristics (usually not exact!):
  - Branch and Bound
  - Sequential rounding
  - SAT
- Usually get good results, but not **optimal** results

A History of Mathematical Programming

1940. Computers emerge Dantzig invents Linear Programming (LP) and the Simplex Algorithm to solve LP

1960's- 1970's. Most theoretical results and algorithms are developed

1979. Khachiyan shows that LP's are polynomial time

1984-1990s. Interior Point Algorithms developed (reliable algorithms for mathematical programming); Emergence of fast and affordable computers

1990's. Convex Optimization is a hot topic

2000's. Robust Optimization is a hot topic

What problems can I solve in a couple hours? (a **very** rough guide)

Exact Optimum:

Unconstrained Smooth Convex Optimization	~10-100 million
Linear Programming	~1 million
Convex Smooth Optimization	~10,000 variables
Convex non-smooth	~1,000~1 million variables
Non-convex continuous	10~1,000 variables
Integer Programming	10~100 variables

Approximate Solutions:

- Pretty much anything!
    - with varying degrees of success

What was not covered in this talk

- Algorithms to perform mathematical programming
- Modeling real world problems as a convex programming problems
- Re-formulating problems for faster solving
- Statistical optimization
- Integer Programming
- Convex analysis

References

- Convex Optimization and Interior Point Methods
  - Convex Optimization, Boyd, Vandenberghe
- Nonlinear Programming Algorithms
  - Numerical Optimization, Nocedal and Wright
- First-order methods and Lagrangian Methods
  - Nonlinear Programming, Bertsekas
- Pre-1990 Optimization
  - Introductory Lectures on Convex Optimization, Nesterov
- Geometric Programming
  - "A Tutorial on Geometric Programming", Boyd et al
- Stochastic Programming
  - "A Tutorial on Stochastic Programming", Shapiro and Philpott

References

- Geometric Programming for analog design
  - Hershenson, Boyd (assorted papers)