Tools for Modeling Optimization Problems
A Short Course

Modeling Concepts

Dr. Ted Ralphs

What will this workshop be about?

- Modeling basic financial optimization problems as mathematical optimization problems.
- Expressing those optimization problems in an algebraic modeling language.
- Getting data into the model, especially through the use of spreadsheets.
- The use of Python as both a modeling tool and as a wrapper for creating applications that populate models with data, perform post-solution analysis and even chain multiple models together.
- Advanced techniques for development of more sophisticated models and methods of analysis. techniques.

What will this workshop NOT be about?

- I'll assume you know the basic principles behind the models we'll look at.
- I will not discuss (much of) the theory behind the models.
- The models I'll show are relatively unsophisticated—their purpose is only to illustrate basic principles of algebraic modeling.
- I won't be able to cover Python as a programming language in much detail, but I hope to leave you with enough knowledge to get started.

The (Open Source) Toolbox

- COIN-OR
- SolverStudio (Excel plug-in)
- Portable Python
- Eclipse
- PyDev (Eclipse plug-in)
- (AMPL)

What is Mathematical Optimization?

- A *mathematical optimization problem* consists of
  - a set of *variables* that describe the state of the system,
  - a set of *constraints* that determine the states that are allowable,
  - external input *parameters* and *data*, and
  - an *objective function* that provides an assessment of how well the system is functioning.
- The variables represent operating decisions that must be made.
- The constraints represent operating specifications.
- The goal is to determine the best operating state consistent with specifications.

Forming a Mathematical Programming Model

The general form of a mathematical programming model is:

$$
\begin{array}{lrcl}
\min \text{ or } \max & f(x_1, \ldots, x_n) & & \\
\text{s.t.} & g_i(x_1, \ldots, x_n) & \left\{ \begin{array}{c} \leq \\ = \\ \geq \end{array} \right\} & b_i, \; i \in M \\
& (x_1, \ldots, x_n) & \in & X
\end{array}
$$

X may be a discrete set, such as $\mathbb{Z}^n$.

Notes:

- There is an important assumption here that all input data are known and fixed.
- Such a mathematical program is called *deterministic*.
- Is this realistic?

Categorizing Mathematical Optimization Problems

- Mathematical optimization problems can be categorized along several fundamental lines.
  - Constrained vs. Unconstrained
  - Convex vs. Nonconvex
  - Linear vs. Nonlinear
  - Discrete vs. Continuous
- What is the importance of these categorizations?
  - Knowing what category an instance is in can tell us something about how difficult it will be to solve.
  - Different solvers are designed for different categories.

Unconstrained Optimization

- When $M = \emptyset$ and $X = \mathbb{R}^n$, we have an *unconstrained optimization problem*.
- Unconstrained optimization problems will not generally arise directly from applications.
- They do, however, arise as *subproblems* when solving constrained optimization problems.
- In unconstrained optimization, it is important to distinguish between the *convex* and *nonconvex* cases.
- In the convex case, optimizing globally is "easy."

Convex Functions

Definition 1. *Let S be a nonempty convex set on $\mathbb{R}^n$. Then the function $f : S \to \mathbb{R}$ is said to be convex on S if*

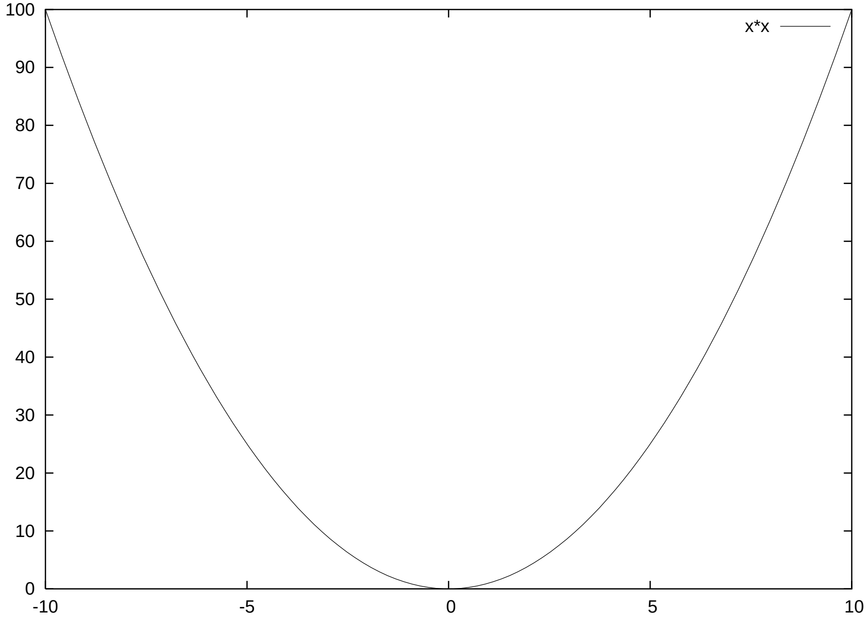
$$f(\lambda x_1 + (1-\lambda)x_2) \leq \lambda f(x_1) + (1-\lambda)f(x_2)$$

for each $x_1, x_2 \in S$ and $\lambda \in (0,1)$.

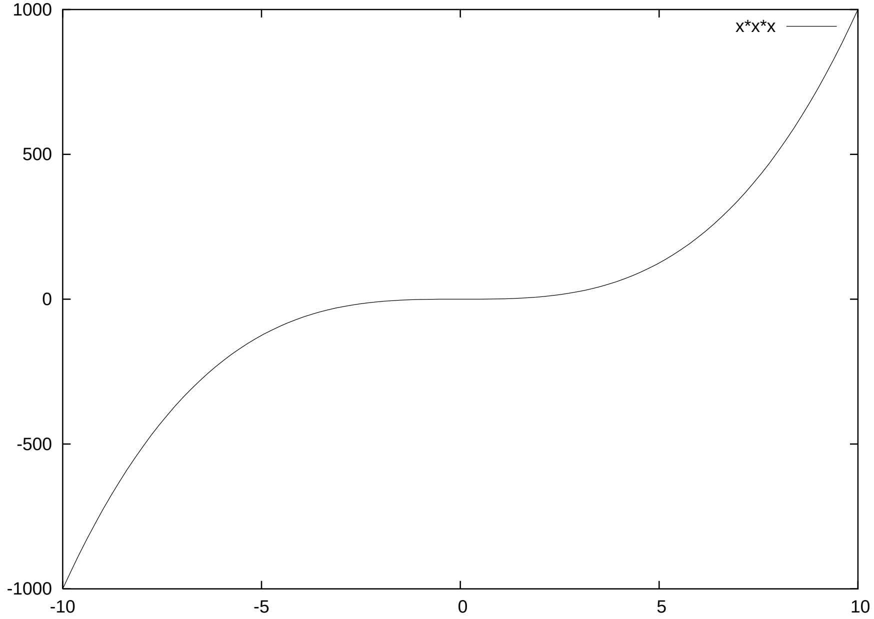
- *Strictly convex* means the inequality is strict.
- *(Strictly) concave* is defined analogously.
- A function that is neither convex nor concave is called *nonconvex*.
- A *linear* function is one that is both concave and convex.

Example: Convex and Nonconvex Functions

CONVEX



NONCONVEX



(Constrained) Linear Optimization

- A *linear program* is one that can be written in a form in which the functions f and g_i, $i \in M$ are all linear and $X = \mathbb{R}^n$.
- In general, a linear program is one that can be written as

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{s.t.} & a_i^\top x \geq b_i \; \forall i \in M_1 \\ & a_i^\top x \leq b_i \; \forall i \in M_2 \\ & a_i^\top x = b_i \; \forall i \in M_3 \\ & x_j \geq 0 \;\; \forall j \in N_1 \\ & x_j \leq 0 \;\; \forall j \in N_2 \end{array}$$

- Equivalently, a linear program can be written as

$$\begin{array}{ll} \text{minimize} & c^\top x \\ \text{s.t.} & Ax \geq b \end{array}$$

- Generally speaking, linear optimization problems are also “easy” to solve.

(Constrained) Nonlinear Optimization

- A *nonlinear program* is any mathematical program that cannot be expressed as a linear program.
- Usually, this terminology also assumes $X = \mathbb{R}^n$.
- Note that by this definition, it is not always obvious whether a given instance is really nonlinear.
- In general, nonlinear optimization problems are difficult to solve to global optimality.

Special Case: Convex Optimization

- A *convex optimization problem* is a nonlinear optimization problem in which the objective function f is convex and the feasible region

$$\mathcal{F} = \{x \in \mathbb{R}^n \mid g_i(x) \geq b_i\}$$

  is a convex set.

- In practice, convex programs are usually "easy" to solve.

Convex Sets

A set S is *convex*

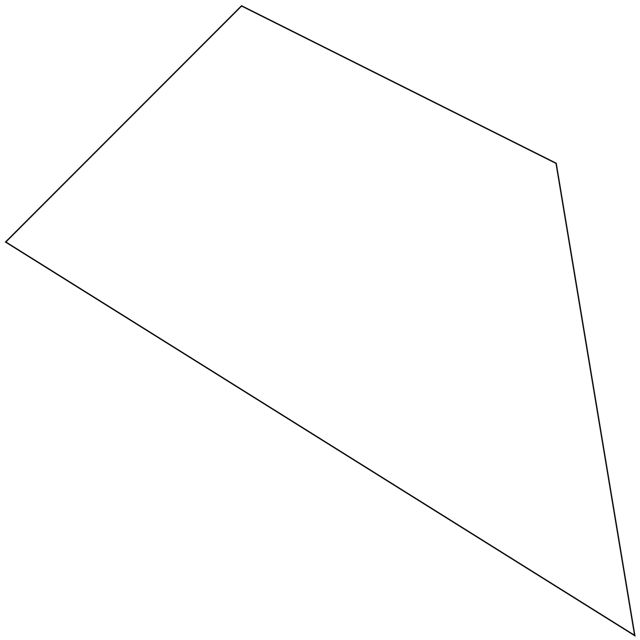
$$\Leftrightarrow$$

$$x_1, x_2 \in S, \lambda \in [0,1] \Rightarrow \lambda x_1 + (1-\lambda)x_2 \in S$$

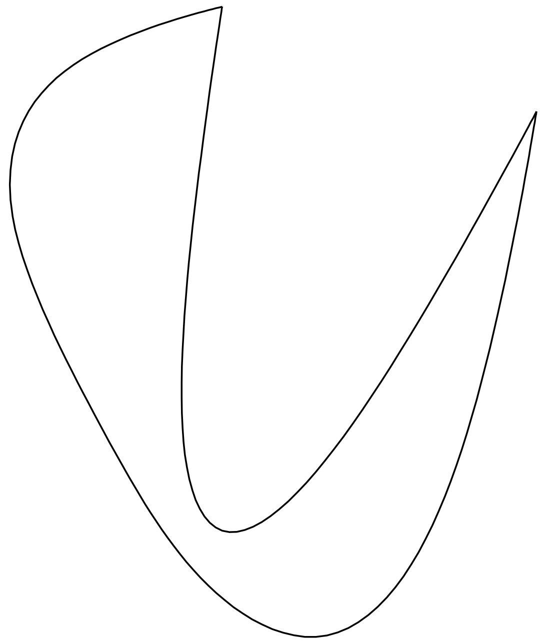
- If $y = \sum \lambda_i x_i$, where $\lambda_i \geq 0$ and $\sum \lambda_i = 1$, then y is a *convex combination* of the x_i's.
- If the positivity restriction on λ is removed, then y is an *affine combination* of the x_i's.
- If we further remove the restriction that $\sum \lambda_i = 1$, then we have a *linear combination*.

Example: Convex and Nonconvex Sets

CONVEX



NONCONVEX



Special Case: Quadratic Optimization

- If all of the functions f and g_i for $i \in M$ are quadratic functions, then we have a *quadratic program*.
- Often, the term quadratic program refers specifically to a program of the form

$$\begin{array}{ll} \text{minimize} & \frac{1}{2}x^\top Qx + c^\top x \\ \text{s.t.} & Ax \geq b \end{array}$$

- Because $x^\top Qx = \frac{1}{2}x^\top (Q + Q^\top)x$, we can assume without loss of generality that Q is *symmetric*.
- The objective function of the above program is then convex if and only if Q is *positive semidefinite*, i.e., $y^\top Qy \geq 0$ for all $y \in \mathbb{R}^n$.
- There are specialized methods for solving convex quadratic optimization problems efficiently.

Special Case: Discrete Optimization

- When $X = \mathbb{Z}^n$, we have a *discrete* or *integer optimization problem*.
- When $X = \mathbb{Z}^r \times \mathbb{R}^{n-r}$, we have a *mixed integer optimization problem*.
- If some of the functions are nonlinear, then we have a *mixed integer nonlinear optimization problem*.
- Almost all mathematical optimization with integer variables are difficult to solve.

Local versus Global Optimization

- For a function $f : \mathbb{R}^n \to \mathbb{R}$, a *local minimizer* is an $\hat{x} \in \mathbb{R}^n$ such that $f(\hat{x}) \leq f(x)$ for all x in a *neighborhood* of $\hat{x}$.
- In general, it is "easy" to find local minimizers.
- A *global minimizer* is $x^* \in \mathbb{R}^n$ such that $f(x^*) \leq f(x)$ for all $x \in \mathbb{R}$.
- The importance of convexity is the following:

Theorem 1. *For a convex function $f : \mathbb{R}^n \to \mathbb{R}$, if $x^* \in \mathbb{R}^n$ is a local optimal solution to $min_{x \in \mathbb{R}^n} f(x)$, then $x^* \in \mathbb{R}^n$ is also a global optimal solution.*

Solutions

- What is the result of analyzing a mathematical program?
- A *solution* is an assignment of values to variables.
- A solution can be thought of as a vector.
- A *feasible solution* is an assignment of values to variables such that all the constraints are satisfied.
- The *objective function value* of a solution is obtained by evaluating the objective function at the given solution.
- An *optimal solution* (assuming minimization) is one whose objective function value is less than or equal to that of all other feasible solutions.
- We may also be interested in some additional qualities of the solution.
  - Sensitivity
  - Robustness
  - Risk

Stochastic Optimization

- In the real world, little is known ahead of time with certainty.
- A *risky investment* is one whose return is not known ahead of time.
- a *risk-free* investment is one whose return is fixed.
- To make decision involving risky investments, we need to incorporate some degree of *stochasticity* into our models.

Types of Uncertainty

- Where does uncertainty come from?
  - Weather
  - Financial Uncertainty
  - Market Uncertainty
  - Competition
  - Technology
- In decision analysis, we must proceed through this list and identify items that might affect the outcome of a decision.

The Scenario Approach to Uncertainty

- The scenario approach assumes that there are a finite number of possible future outcomes of uncertainty.
- Each of these possible outcomes is called a *scenario*.
  - Demand for a product is "low, medium, or high."
  - Weather is "dry or wet."
  - The market will go "up or down."
- Even if this is not reality, often a discrete approximation is useful.

Multi-period Optimization Models

- When we introduce time as an element of a stochastic optimization model, we also have to address the following questions.
  - When do we have to make a given decision?
  - What will we know at the time we are making the decision?
  - How far into the future are we looking?
- Multi-stage models generally assume that decision are made in stages and that in each stage, some amount of uncertainty is resolved.
- Example
  - Fred decides to rebalance his investment portfolio once a quarter.
  - At the outset, he only knows current prices and perhaps some predictions about future prices.
  - At the beginning of each quarter, prices have been revealed and Fred gets a chance to make a "recourse decision."

Example 1: Short Term Financing

A company needs to make provisions for the following cash flows over the coming five months: $-150K$, $-100K$, $200K$, $-200K$, $300K$.

- The following options for obtaining/using funds are available,
  - The company can borrow up to $\$100K$ at 1% interest per month,
  - The company can issue a 2-month zero-coupon bond yielding 2% interest over the two months,
  - Excess funds can be invested at 0.3% monthly interest.
- How should the company finance these cash flows if no payment obligations are to remain at the end of the period?

Example 1 (cont.)

- All investments are risk-free, so there is no stochasticity.
- What are the decision variables?
  - x_i, the amount drawn from the line of credit in month i,
  - y_i, the number of bonds issued in month i,
  - z_i, the amount invested in month i,
- What is the goal?
  - To maximize the the cash on hand at the end of the horizon.

Example 1 (cont.)

The problem can then be modelled as the following linear program:

$$
\begin{aligned}
\max_{(x,y,z,v)\in\mathbb{R}^{12}} \; & f(x,y,z,v) = v \\
\text{s.t. } & x_1 + y_1 - z_1 = 150 \\
& x_2 - 1.01x_1 + y_2 - z_2 + 1.003z_1 = 100 \\
& x_3 - 1.01x_2 + y_3 - 1.02y_1 - z_3 + 1.003z_2 = -200 \\
& x_4 - 1.01x_3 - 1.02y_2 - z_4 + 1.003z_3 = 200 \\
& -1.01x_4 - 1.02y_3 - v + 1.003z_4 = -300 \\
& 100 - x_i \geq 0 \quad (i = 1, \dots, 4) \\
& x_i \geq 0 \quad (i = 1, \dots, 4) \\
& y_i \geq 0 \quad (i = 1, \dots, 3) \\
& z_i \geq 0 \quad (i = 1, \dots, 4) \\
& v \geq 0.
\end{aligned}
$$

Example 2: Portfolio Optimization

- We have the choice to invest in either Intel, Wal-Mart, or G.E..
- All stocks currently cost $\$100$/share.
- In one year's time, the market will be either "up" or "down."
- the Following are the possible outcomes

	Intel	Pepsico	Wal-Mart
Up	130	112	115
Down	90	108	103

- What would you do?

Example 2 (cont.)

- If the probability that the market is up is p, then the value at year 1 from investing $\$1000$ are as follows:
  - Intel: $900 + 400p$
  - Pepsico: $1080 + 40p$
  - Wal-Mart: $1030 + 120p$
- What is your objective?