Organization & Motivation

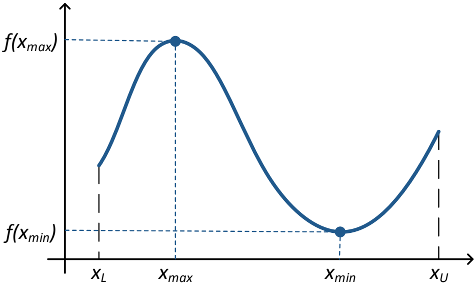
Computational Economics Practice
Winter Term 2015/16
Stefan Feuerriegel

Outline

Outline

Mathematical Optimization

- Optimization uses a rigorous mathematical model to determine the most efficient solution to a described problem
- One must first identify an objective
  - Objective is a quantitative measure of the performance
  - Examples: profit, time, cost, potential energy
  - In general, any quantity (or combination thereof) represented as a single number



Applications of Optimization

- Management
  - Determining product portfolios
  - Location planning
  - Investments decisions
- Game theory
  - Comparing players' strategies
- Logistics
  - Finding optimal routes and schedules
- Design decisions
  - Constructing processes, plants and other equipment
- Operation
  - Adjustment to changes in environmental conditions, production planning, control, etc.
- Mathematical modeling
  - Parameter estimation
  - Model discrimination

Optimization Problem

Optimization is the minimization or maximization of a function subject to (s. t.) constraints on its variables

Notation: General form

$$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \quad \text{s. t.} \quad h(\boldsymbol{x}) = 0$$
$$g(\boldsymbol{x}) \leq 0$$

with

- $\boldsymbol{x} \in \mathbb{R}^n$ as the variable, unknown or parameter
- Objective function $f : D \to \mathbb{R}$, $D \subseteq \mathbb{R}^n$
- Equality constraints $h : D_h \to \mathbb{R}^l$, $D_h \subseteq \mathbb{R}^n$
- Inequality constraints $g : D_g \to \mathbb{R}^k$, $D_g \subseteq \mathbb{R}^n$

Properties of Optimization Problems

Objective	Linear, quadratic, non-linear, etc.
Constraints	Equality and inequality
Variable types	$\boldsymbol{x}$ can be continuous, integer, mixed
Direction	$\min_{\boldsymbol{x}} f(\boldsymbol{x}) \Leftrightarrow \max_{\boldsymbol{x}} -f(\boldsymbol{x})$
Bounds	Lower $\boldsymbol{x}_L \leq \boldsymbol{x}$ or upper $\boldsymbol{x} \leq \boldsymbol{x}_U$
Dimension	One dimensional if $n = 1$, or multi-dimensional if $n > 1$
Optima	Isolated, local or global nature

Classification of Optimization Problems

- Linear Programming (LP)
  - Objective function and constraints are linear
  - $\min_{\boldsymbol{x}} \boldsymbol{c}^T \boldsymbol{x}$ s. t. $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{x} \geq 0$
- Quadratic Programming (QP)
  - Objective function is quadratic and constraints are linear
  - $\min_{\boldsymbol{x}} \boldsymbol{x}^T Q\boldsymbol{x}$ s. t. $A\boldsymbol{x} \leq \boldsymbol{b}$, $\boldsymbol{x} \geq 0$

- Non-Linear Programming (NLP): objective function or at least one constraint is non-linear

- Integer Programming (IP): all variables are discrete
- Mixed Integer Programming (MIP)
  - Continuous and discrete variables
  - Problem can be linear (MILP) or non-linear (MINLP)

Classification of Optimization Problems

- Dynamic Optimization: solution is a function of time
- Stochastic Optimization
  - Model cannot be fully specified, but has uncertainties with confidence estimates
  - Optimize expected performance given uncertainty

Question

- What type is the following optimization problem?

$$\max_{x,y} 3x + y^2 \quad \text{s. t.} \quad x + y < 10 \text{ and } y \in \{1, 2, 4, 8\}$$

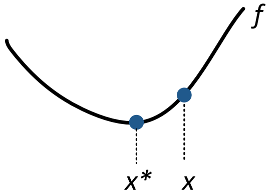
  - MIP
  - MILP
  - MINLP
- Visit `http://pingo.upb.de` with code 1523

Optimal Solution

- x^* is a global minimum if $x^* \in D$ and

$$f(x^*) \leq f(x) \qquad \text{for all } x \in D$$

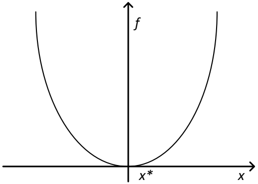
$\rightarrow$ Global minimizers are desired, though often one has only local knowledge of f



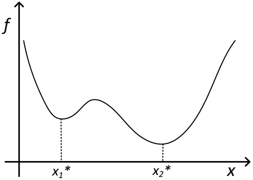
Optimal Solution

Examples of optimal solutions:

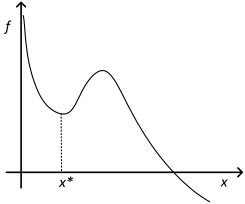
Isolated global solution



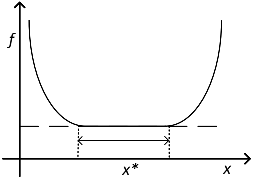
One global, two local solutions



A local but no global solution



Many non-isolated, global solutions



Optimization Procedure

Formulation and solution of optimization problems usually follows:

1. Analysis of environment to determine the variables of interest
2. Definition of optimality criteria as an objective function with (additional) constraints
3. Formulation as a mathematical model with degrees of freedom
4. Numerical optimization to find a solution
5. Verification of the solution through sensitivity analysis (with respect to the assumptions made in the problem formulation)

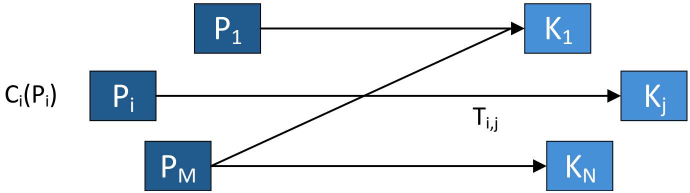
Outline

Production Scheduling

Problem

- Given plants $i = 1, \ldots, M$ where each manufactures P_i goods
- Each plant has a maximal output O_i
- Each plant manufactures at a capacity-specific cost $C_i(P_i)$, which gives the cost as a function of the production
- Each customer $j = 1, \ldots, N$ requests C_j goods

Objective function: find the optimal production schedule such that the manufacturing and shipment costs are minimized



Portfolio Optimization

Problem

- Investor wants to invest money such that it maximizes the investor's utility
- Utility U depends on daily return μ and risk σ^2
- Given risk taking κ, then $U(\mu, \sigma^2) = \mu - \frac{\kappa}{2}\sigma^2$

Objective function: Maximize $U(\mu, \sigma^2)$ among a range of stocks $s_1, \ldots s_N$

$\mathbf{c}^T$	s_1	s_2	s_3
μ	0.2	0.5	0.1

Q	s_1	s_2	s_3
s_1	0.1	0.02	0.02
s_2	0.02	0.1	0.02
s_3	0.02	0.02	0.1

$$\Rightarrow \max_{\mathbf{x}} \mathbf{c}^T \mathbf{x} - \frac{\kappa}{2} \mathbf{x}^T Q \mathbf{x} \quad \text{s.t.} \quad x_i \geq 0 \text{ and } \sum_i x_i = 1$$

Portfolio Optimization in R

```
library(quadprog) # load necessary library
```

```
kappa <- 4 # set risk taking
# objective function
c <- c(0.02, 0.05, 0.01)
Q <- matrix(c(0.1,0.02,0.02, 0.02,0.1,0.02,
              0.02,0.02,0.1), nrow=3)
# constraints
A <- matrix(c(1,1,0,0, 1,0,1,0, 1,0,0,1), nrow=4)
b <- c(1, 0, 0, 0)

sol <- solve.QP(kappa/2*Q, c, t(A), b, meq=1) # call solver
sol$solution # ratio of stocks in portfolio

## [1] 0.2916667 0.4791667 0.2291667

sol$value # minimum value of objective function

## [1] 0.01729167
```

Outlook

1. Introduction to R
2. Advanced R
   - Programming prerequisites
   - Visualize optimization routines
3. Numerical Analysis:
   - Mathematical prerequisites to derive and formalize optimization routines
4. Optimization in R:
   - Use of built-in optimization routines