Notes for Operations Research & More

Lan Peng, PhD Student

Department of Industrial and Systems Engineering
University at Buffalo, SUNY
lanpeng@buffalo.edu

August 27, 2019

August 27, 2019

Contents

Part I

Preliminary Topics

Chapter 1

Review of Linear Algebra

Chapter 2

Convex Sets

Chapter 3

Convex Functions and Generalizations

Part II

Linear Programming

Chapter 4

Formulation

4.1 Typical Problems

4.2 Formulation Skills

4.2.1 Absolute Value

Description: Consider the following model statement:

$$
\begin{aligned}
\min \quad & \sum_{j \in J} c_j |x_j|, \quad c_j > 0 \\
\text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \gtreqless b_i, \quad \forall i \in I \\
& x_j \quad \text{unrestricted}, \quad \forall j \in J
\end{aligned}
$$

Modeling:

$$
\begin{aligned}
\min \quad & \sum_{j \in J} c_j (x_j^+ + x_j^-), \quad c_j > 0 \\
\text{s.t.} \quad & \sum_{j \in J} a_{ij} (x_j^+ - x_j^-) \gtreqless b_i, \quad \forall i \in I \\
& x_j^+, x_j^- \geq 0, \quad \forall j \in J
\end{aligned}
$$

4.2.2 A Minimax Objective

Description: Consider the following model statement:

$$
\begin{aligned}
\min \quad & \max_{k \in K} \sum_{j \in J} c_{kj} x_j \\
\text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \gtreqless b_i, \quad \forall i \in I \\
& x_j \geq 0, \quad \forall j \in J
\end{aligned}
$$

Modeling:

$$
\begin{aligned}
\min \quad & z \\
\text{s.t.} \quad & \sum_{j \in J} a_{ij} x_j \gtreqless b_i, \quad \forall i \in I \\
& \sum_{j \in J} c_{kj} x_j \leq z, \quad \forall k \in K \\
& x_j \geq 0, \quad \forall j \in J
\end{aligned}
$$

4.2.3 A fractional Objective

Description: Consider the following model statement:

$$\begin{aligned}
\min \quad & \frac{\sum_{j\in J} c_j x_j + \alpha}{\sum_{j\in J} d_j x_j + \beta} \\
\text{s.t.} \quad & \sum_{j\in J} a_{ij} x_j \gtreqless b_i, \quad \forall i \in I \\
& x_j \geq 0, \quad \forall j \in J
\end{aligned}$$

Modeling:

$$\begin{aligned}
\min \quad & \sum_{j\in J} c_j x_j t + \alpha t \\
\text{s.t.} \quad & \sum_{j\in J} a_{ij} x_j \gtreqless b_i, \quad \forall i \in I \\
& \sum_{j\in J} d_j x_j t + \beta t = 1 \\
& t > 0 \\
& x_j \geq 0, \quad \forall j \in J \\
& (t = \frac{1}{\sum_{j\in J} d_j x_j + \beta})
\end{aligned}$$

4.2.4 A range Constraint

Description: Consider the following model statement:

$$\begin{aligned}
\min \quad & \sum_{j\in J} c_j x_j \\
\text{s.t.} \quad & d_i \leq \sum_{j\in J} a_{ij} x_j \leq e_i, \quad \forall i \in I \\
& x_j \geq 0, \quad \forall j \in J
\end{aligned}$$

Modeling:

$$\begin{aligned}
\min \quad & \sum_{j\in J} c_j x_j, \quad c_j > 0 \\
\text{s.t.} \quad & u_i + \sum_{j\in J} a_{ij} x_j = e_i, \quad \forall i \in I \\
& x_j \geq 0, \quad \forall j \in J \\
& 0 \leq u_i \leq e_i - d_i, \quad \forall i \in I
\end{aligned}$$

Chapter 5

Simplex Method

5.1 Basic Feasible Solutions and Extreme Points

5.2 Simplex Method

5.2.1 Simplex Method Algorithm

5.2.2 Simplex Method Tableau

5.2.3 Simplex Method as a Search Algorithm

5.3 Revised Simplex Method

5.4 Simplex Method with Bounded Variables

5.5 Artificial Variable

5.5.1 Two-Phase Method

5.5.2 Big-M Method

5.5.3 Single Artificial Variable Technique

5.6 Degeneracy and Cycling

5.6.1 Degeneracy

5.6.2 Cycling

5.6.3 Cycling Prevention Rules

Lexicographic Rule

Bland's Rule

Successive Ratio Rule

Chapter 6

Duality Theory and Sensitivity Analysis

Chapter 7

Decomposition Principle

Chapter 8

Ellipsoid Algorithm

Chapter 9

Projective Algorithm

Chapter 10

Interior-Point Algorithm

Part III

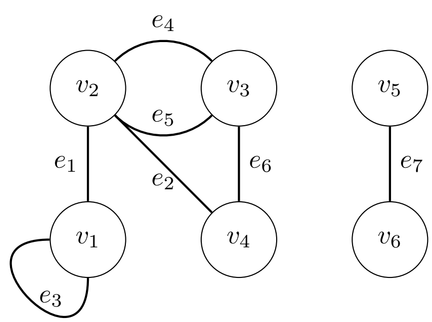
Graph and Network Theory

Chapter 11

Graph Theory

11.1 Basic concepts

A **Graph** G consists of a finite set $V(G)$ on vertices, a finite set $E(G)$ on edgesand an **incident relation** than associates with any edge $e \in E(G)$ an unordered pair of vertices not necessarily distinct called **ends**.



It can be represented as

$$V = V(G) = \{v_1, v_2, v_3, v_4, v_5, v_6\} \tag{11.1}$$

$$E = E(G) = \{e_1, e_2, e_3, e_4, e_5, e_6, e_7\} \tag{11.2}$$

$$e_1 = v_1v_2, e_2 = v_2v_4, ... \tag{11.3}$$

Serveral concepts:
- An edge with identical ends is called a **loop**
- Two edges having the same ends are said to be **parallel**
- A graph without loops or parallel edges is called **simple graph**
- two edges of a graph are **adjancent** if they have a common end
- two vertices are **adjancent** if they are jointed by an edge

11.2 Subgraph

Given two graphs **G** and **H**, **H** is a **subgraph** of **G** if $V(H) \subseteq V(G)$, $E(H) \subseteq E(G)$ and an edge has the smae ends in **H** as it does in **G**, if $E(H) \neq E(G)$ then **H** is a proper subgraph.

A subgraph **H** on **G** is **spanning** if $V(H) = V(G)$

For a subset $V^{'} \subseteq V(G)$ we define an **vertex-induced** subgraph $G[V^{'}]$ to be the subgraph with vertices $V^{'}$ and those edges of **G** having both ends in $V^{'}$

The **edge-induced** subgraph $G[E^{'}]$ has edges $E^{'}$ and those vertices of **G** that are ends to edges in $E^{'}$

If we combine node-indeced or edge-induced subgraphs $G(V^{'})$ and $G(V - V^{'})$, we cannot get the entire graph.

Let $v \in V(G)$, then the **degree** of $v \in V(G)$ denote by $d_G(v)$ is defines to be the number of edges incident of v. Loops counted twice.

Theorem 11.2.1. *For any graph* ***G=(V, E)***

$$\sum_{v \in V} d(v) = 2|E| \tag{11.4}$$

Proof. $\forall$ edge $e = \mu v$ with $\mu \neq v$, e is and counted once for μ and once for v, a total of two altogether. If $e = \mu\mu$, a loop, then it is counted twice for μ □

Corollary 11.2.1.1. *Every graph has an even number of odd degree vertices.*

Proof.

$$V = V_E \cup V_O \Rightarrow \sum_{v \in V} d(v) = \sum_{v \in V_E} d(v) + \sum_{v \in V_O} d(v) = 2|E| \tag{11.5}$$

□

Chapter 12

Paths, Trees, and Cycles

Chapter 13

Shortest-Path Problem

Chapter 14

Minimum Spanning Tree Problem

Chapter 15

Maximum Flow Problem

Chapter 16

Minimum Cost Flow Problem

Chapter 17

Assignment and Matching Problem

Chapter 18

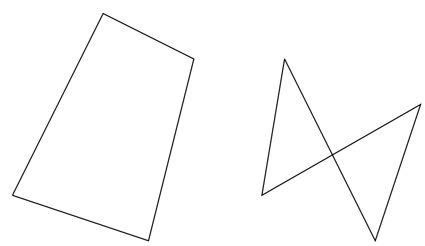
Graph Algorithms

Chapter 19

Polygon Triangulation

19.1 Types of Polygons

Def: A **simple polygon** is a closed polygonal curve without self-intersection.

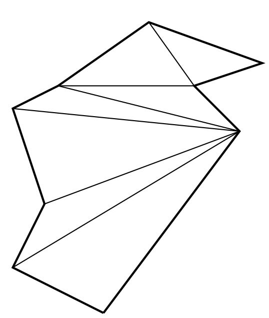


Simple Polygon   Non-simple Polygon

Polygons are basic building blocks in most geometric applications. It can model arbitrarily complex shapes, and apply simple algorithms and algebraic representation/manipulation.

19.2 Triangulation

Def: **Triangulation** is to partition polygon P into non-overlapping triangles using diagonals only. It reduces complex shapes to collection of simpler shapes. Every simple n-gon admits a triangulation which has $n-2$ triangles.



Triangulation

Theorem: Every polygon has a triangulation

Lemma: Every polygon with more than three vertices has a diagonal.

Proof: (by Meisters, 1975) Let P be a polygon with more than three vertices. Every vertex of a P is either *convex* or *concave*. W.L.O.G.(any polygon must has convex corner) Assume p is a convex vertex. Denote the neighbors of

p as q and r. If $\bar{qr}$ is a diagonal, done, and we call $\triangle pqr$ is an *ear*. If $\triangle pqr$ is not an ear, it means at least one vertex is inside $\triangle pqr$, assume among those vertexes inside $\triangle pqr$, s is a vertex closest to p, then $\bar{ps}$ is a diagonal.

19.3 Art Gallery Theorem

Problem: The floor plan of an art gallery modeled as a simple polygon with n vertices, there are guards which is stationed at fixed positions with 360 degree vision but cannot see through the walls. How many guards does the art gallery need for the security? (Fun fact: This problem was posted to Vasek Chvatal by Victor Klee in 1973)

Theorem: Every n-gon can be guarded with $\lfloor \frac{n}{3} \rfloor$ vertex guards

Lemma: Triangulation graph can be 3-colored.

Proof:

- P plus triangulation is a planar graph
- 3-coloring means there exist a 3-partition for vertices that no edge or diagonal has both endpoints within the same set of vertices.
- Proof by Induction:
    - Remove an ear (there will always exist ear)
    - Inductively 3-color the rest
    - Put ear back, coloring new vertex with the label not used by the boundary diagonal.

19.4 Triangulation Algorithms

19.5 Shortest Path

Part IV

Integer and Combinatorial Programming

Chapter 20

Formulation

20.1 Typical Problems

20.2 Integer Programming Formulation Skills

20.2.1 A Variable Taking Discontinuous Values

Description: In algebraic notation:

$$x = 0, \quad \text{or} \quad l \le x \le u$$

Modeling:

$$\begin{aligned} &x \le uy \\ &x \ge ly \\ &y \in \{0, 1\} \end{aligned}$$

where

$$y = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } l \le x \le u \end{cases}$$

20.2.2 Fixed Costs

Description: In algebraic notation:

$$C(x) = \begin{cases} 0 & \text{for } x = 0 \\ k + cx & \text{for } x > 0 \end{cases}$$

Modeling:

$$\begin{aligned} &C^*(x, y) = ky + cx \\ &x \le My \\ &x \ge 0 \\ &y \in \{0, 1\} \end{aligned}$$

where

$$y = \begin{cases} 0, & \text{if } x = 0 \\ 1, & \text{if } x \ge 0 \end{cases}$$

20.2.3 Either-or Constraints

Description: In algebraic notation:

$$\sum_{j \in J} a_{1j} x_j \le b_1 \text{ or } \sum_{j \in J} a_{2j} x_j \le b_2$$

Modeling:

$$\begin{aligned}
&\sum_{j\in J} a_{1j}x_j \le b_1 + M_1 y\\
&\sum_{j\in J} a_{2j}x_j \le b_2 + M_1(1-y)\\
&y \in \{0,1\}
\end{aligned}$$

where

$$y = \begin{cases} 0, & \text{if } \sum_{j\in J} a_{1j}x_j \le b_1\\ 1, & \text{if } \sum_{j\in J} a_{2j}x_j \le b_2 \end{cases}$$

Notice that the sign before M is determined by the inequality $\ge$ or $\le$, if it is "$\ge$", use "$-$", if it "$\le$", use "$+$".

20.2.4 Conditional Constraints

Description: If constraint A is satisfied, then constraint B must also be satisfied

$$\text{If } \sum_{j\in J} a_{1j}x_j \le b_1 \text{ then } \sum_{j\in J} a_{2j}x_j \le b_2$$

The key part is to find the opposite of the first condition. We are using $A \Rightarrow B \Leftrightarrow \neg B \Rightarrow \neg A$
Therefore it is equivalent to

$$\sum_{j\in J} a_{1j}x_j > b_1 \text{ or } \sum_{j\in J} a_{2j}x_j \le b_2$$

Furthermore, it is equivalent to

$$\sum_{j\in J} a_{1j}x_j \ge b_1 + \epsilon \text{ or } \sum_{j\in J} a_{2j}x_j \le b_2$$

Where ϵ is a very small positive number.

Modeling:

$$\begin{aligned}
&\sum_{j\in J} a_{1j}x_j \ge b_1 + \epsilon - M_2 y\\
&\sum_{j\in J} a_{2j}x_j \le b_2 + M_2(1-y)\\
&y \in \{0,1\}
\end{aligned}$$

20.2.5 Special Ordered Sets

SOS1 Description Out of a set of yes-no decisions, at most one decision variable can be yes.

$$x_1 = 1, x_2 = x_3 = \cdots = x_n = 0$$
or
$$x_2 = 1, x_1 = x_3 = \cdots = x_n = 0$$
or ...

Modeling:

$$\sum_i x_i = 1, \quad i \in N$$

SOS2 Description 1 Out of a set of binary variables, at most two variables can be nonzero. In addition, the two variables must be adjacent to each other in a fixed order list.

Modeling: If $x_1, x_2, ..., x_n$ is a SOS2, then

$$\sum_{i=1}^{n} x_i \leq 2$$
$$x_i + x_j \leq 1, \forall i \in \{1, 2, ..., n\}, j \in \{i + 2, i + 3, ..., n\}$$
$$x_i \in \{0, 1\}$$

SOS2 Description 2 There is another type of definition, that is out of a set of nonnegative variables **not binary here**, at most two variables can be nonzero. In addition, the two variables must be adjacent to each other in a fixed order list. All variables summing to 1.
This definition of SOS2 is used in the following section *Piecewise Linear Formulations*

20.2.6 Piecewise Linear Formulations

Description: The objective function is a sequence of line segments, e.g. $y = f(x)$, consists $k - 1$ linear segments going through k given points $(x_1, y_1), (x_2, y_2), ..., (x_k, y_k)$.
Denote

$$d_i = \begin{cases} 1, & x \in (x_i, x_{i+1}) \\ 0, & \text{otherwise} \end{cases}$$

Then the objective function is

$$\sum_{i \in \{1,2,...,k-1\}} y = d_i f_i(x)$$

Modeling: Given that objective function as a piecewise linear formulation, we can have these constraints

$$\sum_{i \in \{1,2,...,k-1\}} d_i = 1$$
$$d_i \in \{0, 1\}, i \in \{1, 2, ..., k - 1\}$$
$$x = \sum_{i \in \{1,2,...,k\}} w_i x_i$$
$$y = \sum_{i \in \{1,2,...,k\}} w_i y_i$$
$$w_1 \leq d_1$$
$$w_i \leq d_{i-1} + di, i \in \{2, 3, ..., k - 1\}$$
$$w_k \leq d_{k-1}$$

In this case, $w_i \in SOS2$ (second definition)

20.2.7 Conditional Binary Variables

Description: Choose at most n binary variable to be 1 out of $x_1, x_2, ...x_m, m \geq n$. If $n = 1$ then it is SOS1.
Modeling:

$$\sum_{k \in \{1,2,...,m\}} x_k \leq n$$

Description: Choose exactly n binary variable to be 1 out of $x_1, x_2, ...x_m, m \geq n$
Modeling:

$$\sum_{k \in \{1,2,...,m\}} x_k = n$$

$\boxed{\textbf{Description:}}$ Choose x_j only if $x_k = 1$

$\boxed{\textbf{Modeling:}}$

$$x_j = x_k$$

$\boxed{\textbf{Description:}}$ "and" condition, iff $x_1, x_2, ..., x_m = 1$ then $y = 1$

$\boxed{\textbf{Modeling:}}$

$$y \leq x_i, i \in \{1, 2, ..., m\}$$
$$y \geq \sum_{i \in \{1,2,...,m\}} x_i - (m - 1)$$

20.2.8 Elimination of Products of Variables

$\boxed{\textbf{Description:}}$ For variables x_1 and x_2,

$$y = x_1 x_2$$

$\boxed{\textbf{Modeling:}}$ If x_1, x_2 are binary, it is the same as "and" condition of binary variables.
If x_1 is binary, while x_2 is continuous and $0 \leq x_2 \leq u$, then

$$y \leq u x_1$$
$$y \leq x_2$$
$$y \geq x_2 - u(1 - x_1)$$
$$y \geq 0$$

If both x_1 and x_2 are continuous, it is non-linear, we can use Piecewise linear formulation to simulate.

Chapter 21

Branch and Bound

Chapter 22

Branch and Cut

Chapter 23

Packing and Matching

Chapter 24

Traveling Salesman Problem

Chapter 25

Knapsack Problem

Part V

Nonlinear Programming

Chapter 26

KKT Optimality Conditions

Chapter 27

Lagrangian Duality

Chapter 28

Unconstrained Optimization

Chapter 29

Penalty and Barrier Functions

Part VI

Algorithms and Computational Complexity

Chapter 30

Computational Complexity

Chapter 31

Sorting

31.1 Elementary Sorting Algorithms

31.2 Heap-sort

31.3 Quick-sort

31.4 Sorting in Linear Time

31.5 Medians and Order Statistics

Chapter 32

Data Structures

32.1 Elementary Data Structures

32.2 Hash Tables

32.3 Binary Search Trees

32.4 Red-Black Trees

32.5 B-Trees

32.6 Fibonacci Heaps

32.7 van Emde Boas Trees

Chapter 33

Design and Analysis Techniques

33.1 Dynamic Programming

33.2 Greedy Algorithms

33.3 Amortized Analysis

33.4 Multi-threaded Algorithms

33.5 Matrix Operations

33.6 Polynomials and the FFT

33.7 Number-Theoretic Algorithms

33.8 String Matching

33.9 Computational Geometry

33.10 NP-Completeness

33.11 Approximation Algorithms

Part VII

Heuristics ans Meta-heuristics

Part VIII

Game Theory

Chapter 34

Games with Ordinal Payoffs

34.1 Ordinal Games in Strategic Form

34.2 Perfect-information Games

34.3 General Dynamic Games

Chapter 35

Games with Cardinal Payoffs

35.1 Expected Utility Theory

35.2 Strategic-form Games

35.3 Extensive-form Games

Chapter 36

Knowledge, Common Knowledge, Beliefs

36.1 Common Knowledge

36.2 Adding Beliefs to Knowledge

36.3 Rationality

Chapter 37

Refinements of Subgame-perfect Equilibrium

37.1 Weak Sequential Equilibrium

37.2 Sequential Equilibrium

37.3 Perfect Bayesian Equilibrium

Chapter 38

Incomplete Information

38.1 Static Games

38.2 Dynamic Games

38.3 The Type-Space Approach

Part IX

Probability, Stochastic Processes and Markov Chains

Chapter 39

Probability

Chapter 40

Random Variables

40.1 Relationship between Some Random Variables

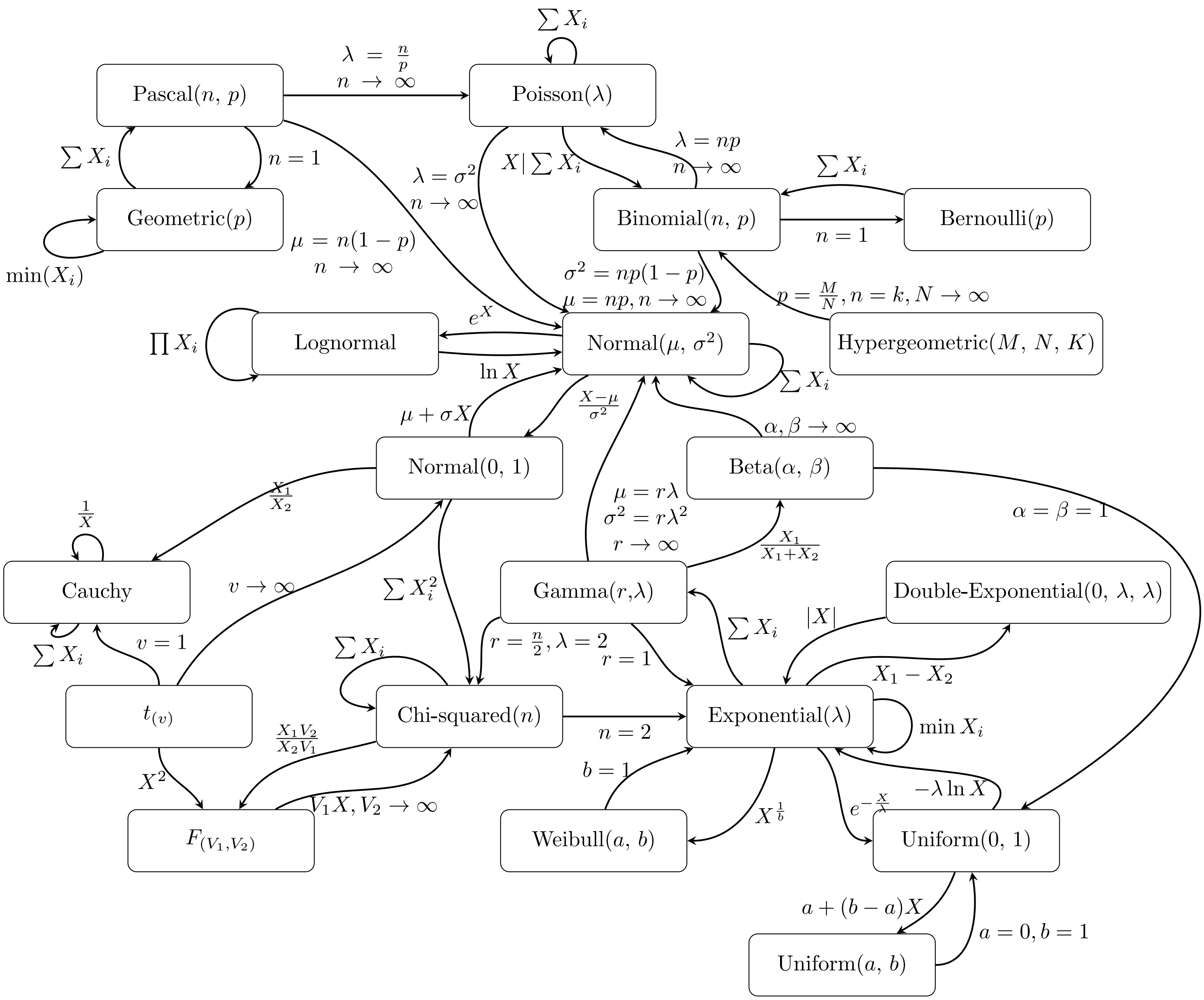


Figure 40.1: Relationship between Some Random Variables

40.2 Discrete Random Variables

Table 40.1: Discrete Random Variables

Distribution	PMF	CDF	Expectation	Variance	MGF
Discrete Uniform(a, b)	$f(x) = \frac{1}{b-a+1}$ $x = a, a+1, ..., b$	$F(x) = \frac{x-a+1}{b-a+1}$ $x = a, a+1, ..., b$	$E[X] = \frac{b-a}{2}$	$D[X] = \frac{(b-a+1)^2-1}{12}$	$M(t) = \frac{e^{at}-e^{(b+1)t}}{(b-a+1)(1-e^t)}$ $t \in \mathbb{R}$
Bernoulli(p)	$f(x) = p^x(1-p)^{1-x}$ $x \in \{0, 1\}$	$F(x) = \begin{cases} 0, & x < 0 \\ 1-p, & 0 \leq x \leq 1 \\ 1, & x > 1 \end{cases}$	$E[X] = p$	$D[X] = p(1-p)$	$M(t) = 1 - p + pe^t$ $t \in \mathbb{R}$
Binomial(n, p)	$f(x) = \binom{n}{x} p^x(1-p)^{n-x}$ $x = 0, 1, ..., n$	$F(x) = \sum_{k=0}^{x} \binom{n}{k} p^k(1-p)^{n-k}$ $x = 0, 1, ..., n$	$E[X] = np$	$D[X] = np(1-p)$	$M(t) = (1 - p + pe^t)^n$ $t \in \mathbb{R}$
Poisson(μ)	$f(x) = \frac{\mu^x e^{\mu}}{x!}$ $x = 0, 1, ..., n, ...$	$f(x) = \frac{\Gamma(x+1,\mu)}{\Gamma(x+1)}$ $x = 0, 1, ..., n, ...$	$E[X] = \mu$	$D[X] = \mu$	$M(t) = e^{\mu(e^t-1)}$ $t \in \mathbb{R}$
Geometric(p)	$f(x) = p(1-p)^x$ $x = 0, 1, ..., n, ...$	$F(x) = 1 - (1-p)^{x+1}$ $x = 0, 1, ..., n, ...$	$E[X] = \frac{1-p}{p}$	$D[X] = \frac{1-p}{p^2}$	$M(t) = \frac{p}{1-(1-p)e^t}$ $t < -\ln(1-p)$
Pascal(n, p)	$f(x) = \binom{n-1+x}{x} p^n(1-p)^x$ $x = 0, 1, 2, ..., n, ...$	$F(x) = 1 - I_p(k+1, n)$ $x = 0, 1, 2, ..., n, ...$	$E[X] = \frac{n(1-p)}{p}$	$D[X] = \frac{n(1-p)}{p^2}$	$M(t) = (\frac{p}{1-(1-p)e^t})^n$ $t < -\ln(1-p)$

40.3 Continuous Random Variables

Table 40.2: Continuous Random Variables

Distribution	PDF	CDF	Expectation	Variance	MGF
Uniform(a, b)	$f(x) = \frac{1}{b-a}$ $x = [a, b]$	$F(x) = \frac{x-a}{b-a}$ $x = [a, b]$	$E[X] = \frac{b-a}{2}$	$D[X] = \frac{(b-a)^2}{12}$	$M(t) = \begin{cases} 1, & t = 0 \\ \frac{e^{bt}-e^{at}}{t(b-a)}, & t \neq 0 \end{cases}$
Normal(μ, σ)	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $x \in \mathbb{R}$	$F(x) = \int_{-\infty}^{x} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $x \in \mathbb{R}$	$E[X] = \mu$	$D[X] = \sigma^2$	$e^{\frac{t(t\sigma^2+2\mu)}{2}}$ $t \in \mathbb{R}$
Exponential$\lambda)$	$f(x) = \lambda e^{-\lambda x}$ $x > 0$	$F(x) = 1 - e^{\lambda x}$ $x > 0$	$E[X] = \frac{1}{\lambda}$	$D[X] = \frac{1}{\lambda^2}$	$\frac{1}{1-\frac{t}{\lambda}}$ $t < \lambda$
Erlang(n, λ)	$f(x) = \frac{\lambda^n x^{n-1} e^{-\lambda x}}{(n-1)!}$ $x > 0$	$F(x) = 1 - \sum_{i=0}^{n-1} \frac{\lambda^n x^n e^{-\lambda x}}{n!}$ $x > 0$	$E[X] = \frac{n}{\lambda}$	$D[X] = \frac{n}{\lambda^2}$	$\frac{1}{(1-\frac{t}{\lambda})^n}$ $t < \lambda$

Chapter 41

Limit Theorems

Chapter 42

The Bernoulli and Poisson Process

Chapter 43

Discrete-Time Markov Chains

Chapter 44

Continuous-Time Markov Chains

Part X

Queuing Theory

Chapter 45

Queuing Model

Chapter 46

Birth-and-Death Queuing Models

Chapter 47

Multidimensional Birth-and-Death Queuing Models

Chapter 48

Phase-Type Queue

Chapter 49

Bulk Queue

Chapter 50

Imbedded-Markov-Chain Queuing Models

Chapter 51

Queuing Network

Part XI

Inventory Theory

Part XII

Reliability Theory

Part XIII

Statistic

Part XIV

Simulation