Notes for Operations Research & More

Lan Peng, PhD Student

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

July 24, 2019

July 24, 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

$\boxed{\textbf{Description:}}$ Consider the following model statement:

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

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

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

4.2.4 A range Constraint

$\boxed{\textbf{Description:}}$ Consider the following model statement:

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

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

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

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

Paths, Trees, and Cycles

Chapter 12

Shortest-Path Problem

Chapter 13

Minimum Spanning Tree Problem

Chapter 14

Maximum Flow Problem

Chapter 15

Minimum Cost Flow Problem

Chapter 16

Assignment and Matching Problem

Chapter 17

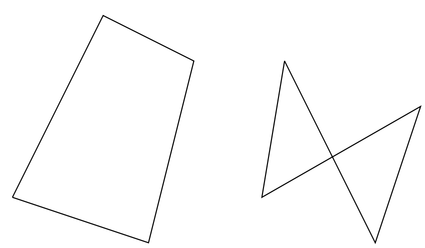
Graph Algorithms

Chapter 18

Polygon Triangulation

18.1 Types of Polygons

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

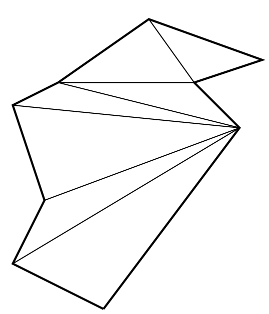


Simple Polygon Non-simple Polygon

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

18.2 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.

18.3 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.

18.4 Triangulation Algorithms

18.5 Shortest Path

Part IV

Integer and Combinatorial Programming

Chapter 19

Formulation

19.1 Typical Problems

19.2 Integer Programming Formulation Skills

19.2.1 A Variable Taking Discontinuous Values

$\boxed{\textbf{Description:}}$ In algebraic notation:

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

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

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

where

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

19.2.2 Fixed Costs

$\boxed{\textbf{Description:}}$ In algebraic notation:

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

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

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

where

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

19.2.3 Either-or Constraints

$\boxed{\textbf{Description:}}$ In algebraic notation:

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

Modeling:

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

where

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

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

19.2.4 Conditional Constraints

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

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

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

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

Furthermore, it is equivalent to

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

Where ϵ is a very small positive number.
Modeling:

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

19.2.5 Special Ordered Sets

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

$$
\begin{gathered}
x_1 = 1, x_2 = x_3 = \cdots = x_n = 0 \\
\text{or} \\
x_2 = 1, x_1 = x_3 = \cdots = x_n = 0 \\
\text{or ...}
\end{gathered}
$$

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*

19.2.6 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)

19.2.7 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 $y = 1$

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

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

19.2.8 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 20

Branch and Bound

Chapter 21

Branch and Cut

Chapter 22

Packing and Matching

Chapter 23

Traveling Salesman Problem

Chapter 24

Knapsack Problem

Part V

Nonlinear Programming

Chapter 25

KKT Optimality Conditions

Chapter 26

Lagrangian Duality

Chapter 27

Unconstrained Optimization

Chapter 28

Penalty and Barrier Functions

Part VI

Algorithms and Computational Complexity

Chapter 29

Computational Complexity

Chapter 30

Sorting

30.1 Elementary Sorting Algorithms

30.2 Heap-sort

30.3 Quick-sort

30.4 Sorting in Linear Time

30.5 Medians and Order Statistics

Chapter 31

Data Structures

31.1 Elementary Data Structures

31.2 Hash Tables

31.3 Binary Search Trees

31.4 Red-Black Trees

31.5 B-Trees

31.6 Fibonacci Heaps

31.7 van Emde Boas Trees

Chapter 32

Design and Analysis Techniques

32.1 Dynamic Programming

32.2 Greedy Algorithms

32.3 Amortized Analysis

32.4 Multi-threaded Algorithms

32.5 Matrix Operations

32.6 Polynomials and the FFT

32.7 Number-Theoretic Algorithms

32.8 String Matching

32.9 Computational Geometry

32.10 NP-Completeness

32.11 Approximation Algorithms

Part VII

Heuristics ans Meta-heuristics

Part VIII

Game Theory

Chapter 33

Games with Ordinal Payoffs

33.1 Ordinal Games in Strategic Form

33.2 Perfect-information Games

33.3 General Dynamic Games

Chapter 34

Games with Cardinal Payoffs

34.1 Expected Utility Theory

34.2 Strategic-form Games

34.3 Extensive-form Games

Chapter 35

Knowledge, Common Knowledge, Beliefs

35.1 Common Knowledge

35.2 Adding Beliefs to Knowledge

35.3 Rationality

Chapter 36

Refinements of Subgame-perfect Equilibrium

36.1 Weak Sequential Equilibrium

36.2 Sequential Equilibrium

36.3 Perfect Bayesian Equilibrium

Chapter 37

Incomplete Information

37.1 Static Games

37.2 Dynamic Games

37.3 The Type-Space Approach

Part IX

Probability, Stochastic Processes and Markov Chains

Chapter 38

Probability

Chapter 39

Random Variables

39.1 Relationship between Some Random Variables

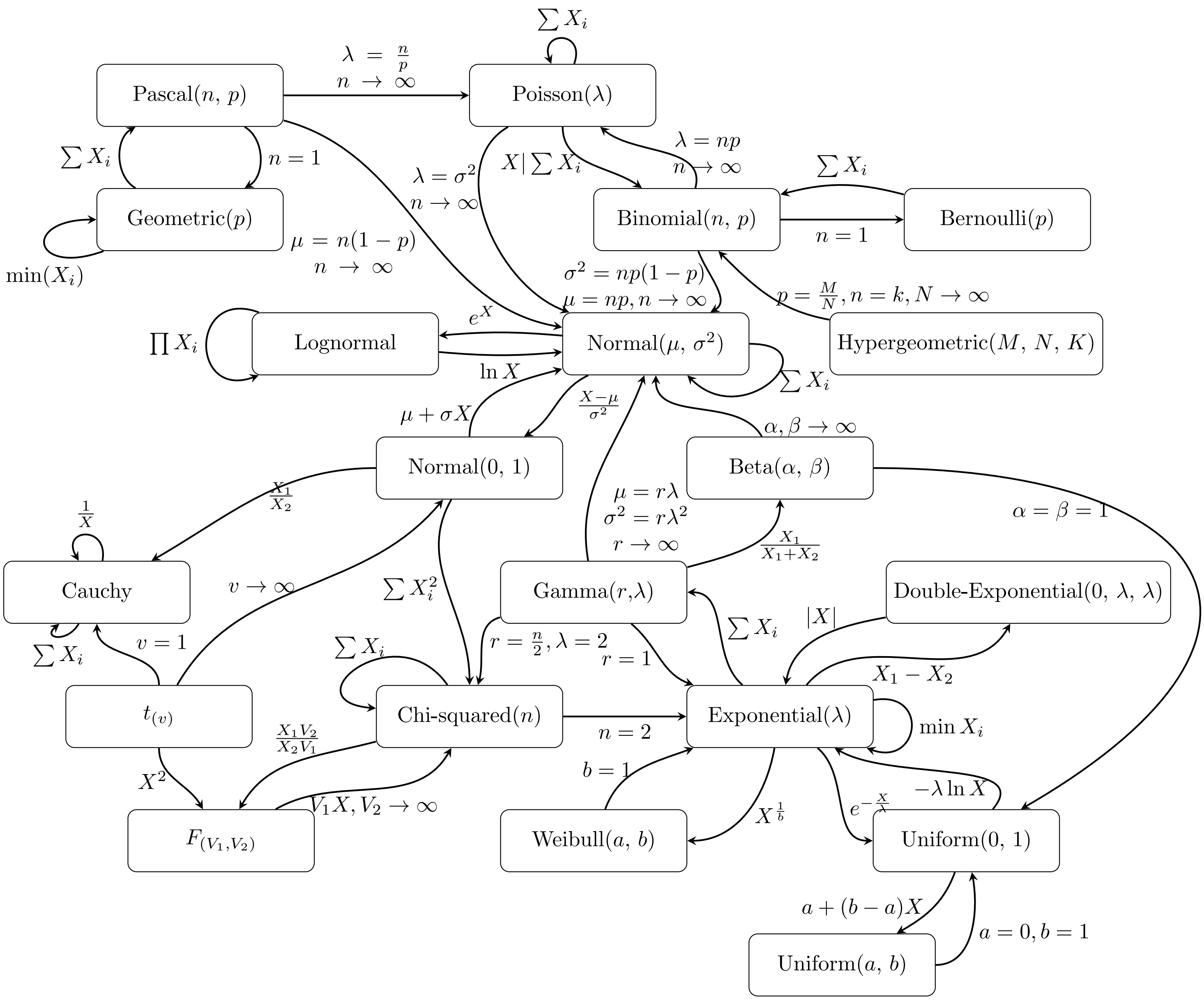


Figure 39.1: Relationship between Some Random Variables

39.2 Discrete Random Variables

Table 39.1: 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)	$f(x)=p^x(1-p)^{1-x}$ $x\in\{0,1\}$	$F(x)=\begin{cases}0, & x<0\\ 1-p, & 0\le x\le 1\\ 1, & x>1\end{cases}$	$E[X]=p$	$D[X]=p(1-p)$	$M(t)=1-p+pe^t$ $t\in\mathbb{R}$
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)	$f(x)=\binom{n-1+x}{x}p^n(1-p)^x$ $x=0,1,2,...,n,...$	$F(x)=1-I_p(k+1,n)$ $x=0,1,2,...,n,...$	$E[X]=\frac{n(1-p)}{p}$	$D[X]=\frac{n(1-p)}{p^2}$	$M(t)=(\frac{p}{1-(1-p)e^t})^n$ $t<-\ln(1-p)$

39.3 Continuous Random Variables

Table 39.2: 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)	$f(x)=\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $x\in\mathbb{R}$	$F(x)=\int_{-\infty}^{x}\frac{1}{\sqrt{2\pi}\sigma}e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $x\in\mathbb{R}$	$E[X]=\mu$	$D[X]=\sigma^2$	$e^{\frac{t(t\sigma^2+2\mu)}{2}}$ $t\in\mathbb{R}$
Exponentialλ)	$f(x)=\lambda e^{-\lambda x}$ $x>0$	$F(x)=1-e^{\lambda x}$ $x>0$	$E[X]=\frac{1}{\lambda}$	$D[X]=\frac{1}{\lambda^2}$	$\frac{1}{1-\frac{t}{\lambda}}$ $t<\lambda$
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 40

Limit Theorems

Chapter 41

The Bernoulli and Poisson Process

Chapter 42

Discrete-Time Markov Chains

Chapter 43

Continuous-Time Markov Chains

Part X

Queuing Theory

Chapter 44

Queuing Model

Chapter 45

Birth-and-Death Queuing Models

Chapter 46

Multidimensional Birth-and-Death Queuing Models

Chapter 47

Phase-Type Queue

Chapter 48

Bulk Queue

Chapter 49

Imbedded-Markov-Chain Queuing Models

Chapter 50

Queuing Network

Part XI

Inventory Theory

Part XII

Reliability Theory

Part XIII

Statistic

Part XIV

Simulation