

# Notes for Operations Research & More

Lan Peng, PhD Student

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

June 10, 2019



# Contents

<b>I</b>	<b>Preliminary Topics</b>	<b>7</b>
1	Review of Linear Algebra	9
2	Convex Sets	11
3	Convex Functions and Generalizations	13
<b>II</b>	<b>Linear Programming</b>	<b>15</b>
4	Formulation	17
4.1	Typical Problems . . . . .	17
4.2	Formulation Skills . . . . .	17
4.2.1	Absolute Value . . . . .	17
4.2.2	A Minimax Objective . . . . .	17
4.2.3	A fractional Objective . . . . .	18
4.2.4	A range Constraint . . . . .	18
5	Simplex Method	19
5.1	Basic Feasible Solutions and Extreme Points . . . . .	19
5.2	Simplex Method . . . . .	19
5.2.1	Simplex Method Algorithm . . . . .	19
5.2.2	Simplex Method Tableau . . . . .	19
5.2.3	Simplex Method as a Search Algorithm . . . . .	19
5.3	Revised Simplex Method . . . . .	19
5.4	Simplex Method with Bounded Variables . . . . .	19
5.5	Artificial Variable . . . . .	19
5.5.1	Two-Phase Method . . . . .	19
5.5.2	Big-M Method . . . . .	19
5.5.3	Single Artificial Variable Technique . . . . .	19
5.6	Degeneracy and Cycling . . . . .	19
5.6.1	Degeneracy . . . . .	19
5.6.2	Cycling . . . . .	19
5.6.3	Cycling Prevention Rules . . . . .	19
6	Duality Theory and Sensitivity Analysis	21
7	Decomposition Principle	23
8	Ellipsoid Algorithm	25
9	Projective Algorithm	27
10	Interior-Point Algorithm	29
<b>III</b>	<b>Graph and Network Theory</b>	<b>31</b>
11	Paths, Trees, and Cycles	33

<b>12 Shortest-Path Problem</b>	<b>35</b>
<b>13 Minimum Spanning Tree Problem</b>	<b>37</b>
<b>14 Maximum Flow Problem</b>	<b>39</b>
<b>15 Minimum Cost Flow Problem</b>	<b>41</b>
<b>16 Assignment and Matching Problem</b>	<b>43</b>
<b>17 Graph Algorithms</b>	<b>45</b>
 <b>IV Integer and Combinatorial Programming</b>	 <b>47</b>
<b>18 Formulation</b>	<b>49</b>
18.1 Typical Problems . . . . .	49
18.2 Integer Programming Formulation Skills . . . . .	49
18.2.1 A Variable Taking Discontinuous Values . . . . .	49
18.2.2 Fixed Costs . . . . .	49
18.2.3 Either-or Constraints . . . . .	49
18.2.4 Conditional Constraints . . . . .	50
18.2.5 Special Ordered Sets . . . . .	50
18.2.6 Piecewise Linear Formulations . . . . .	51
18.2.7 Conditional Binary Variables . . . . .	51
18.2.8 Elimination of Products of Variables . . . . .	52
<b>19 Branch and Bound</b>	<b>53</b>
<b>20 Branch and Cut</b>	<b>55</b>
<b>21 Packing and Matching</b>	<b>57</b>
<b>22 Traveling Salesman Problem</b>	<b>59</b>
<b>23 Knapsack Problem</b>	<b>61</b>
 <b>V Nonlinear Programming</b>	 <b>63</b>
<b>24 KKT Optimality Conditions</b>	<b>65</b>
<b>25 Lagrangian Duality</b>	<b>67</b>
<b>26 Unconstrained Optimization</b>	<b>69</b>
<b>27 Penalty and Barrier Functions</b>	<b>71</b>
 <b>VI Algorithms and Computational Complexity</b>	 <b>73</b>
<b>28 Computational Complexity</b>	<b>75</b>
<b>29 Sorting</b>	<b>77</b>
29.1 Elementary Sorting Algorithms . . . . .	77
29.2 Heap-sort . . . . .	77
29.3 Quick-sort . . . . .	77
29.4 Sorting in Linear Time . . . . .	77
29.5 Medians and Order Statistics . . . . .	77

<b>30 Data Structures</b>	<b>79</b>
30.1 Elementary Data Structures . . . . .	79
30.2 Hash Tables . . . . .	79
30.3 Binary Search Trees . . . . .	79
30.4 Red-Black Trees . . . . .	79
30.5 B-Trees . . . . .	79
30.6 Fibonacci Heaps . . . . .	79
30.7 van Emde Boas Trees . . . . .	79
<b>31 Design and Analysis Techniques</b>	<b>81</b>
31.1 Dynamic Programming . . . . .	81
31.2 Greedy Algorithms . . . . .	81
31.3 Amortized Analysis . . . . .	81
31.4 Multi-threaded Algorithms . . . . .	81
31.5 Matrix Operations . . . . .	81
31.6 Polynomials and the FFT . . . . .	81
31.7 Number-Theoretic Algorithms . . . . .	81
31.8 String Matching . . . . .	81
31.9 Computational Geometry . . . . .	81
31.10 NP-Completeness . . . . .	81
31.11 Approximation Algorithms . . . . .	81
<b>VII Heuristics and Meta-heuristics</b>	<b>83</b>
<b>VIII Game Theory</b>	<b>85</b>
<b>32 Games with Ordinal Payoffs</b>	<b>87</b>
32.1 Ordinal Games in Strategic Form . . . . .	87
32.2 Perfect-information Games . . . . .	87
32.3 General Dynamic Games . . . . .	87
<b>33 Games with Cardinal Payoffs</b>	<b>89</b>
33.1 Expected Utility Theory . . . . .	89
33.2 Strategic-form Games . . . . .	89
33.3 Extensive-form Games . . . . .	89
<b>34 Knowledge, Common Knowledge, Beliefs</b>	<b>91</b>
34.1 Common Knowledge . . . . .	91
34.2 Adding Beliefs to Knowledge . . . . .	91
34.3 Rationality . . . . .	91
<b>35 Refinements of Subgame-perfect Equilibrium</b>	<b>93</b>
35.1 Weak Sequential Equilibrium . . . . .	93
35.2 Sequential Equilibrium . . . . .	93
35.3 Perfect Bayesian Equilibrium . . . . .	93
<b>36 Incomplete Information</b>	<b>95</b>
36.1 Static Games . . . . .	95
36.2 Dynamic Games . . . . .	95
36.3 The Type-Space Approach . . . . .	95
<b>IX Decision Analysis</b>	<b>97</b>
<b>X Probability, Stochastic Processes and Markov Chains</b>	<b>99</b>
<b>37 Probability</b>	<b>101</b>

<b>38 Random Variables</b>	<b>103</b>
38.1 Relationship between Some Random Variables . . . . .	103
38.2 Discrete Random Variables . . . . .	104
38.3 Continuous Random Variables . . . . .	104
<b>39 Limit Theorems</b>	<b>105</b>
<b>40 The Bernoulli and Poisson Process</b>	<b>107</b>
<b>41 Discrete-Time Markov Chains</b>	<b>109</b>
<b>42 Continuous-Time Markov Chains</b>	<b>111</b>
 <b>XI Queuing Theory</b>	 <b>113</b>
43 Queuing Model	115
44 Birth-and-Death Queuing Models	117
45 Multidimensional Birth-and-Death Queuing Models	119
46 Phase-Type Queue	121
47 Bulk Queue	123
48 Imbedded-Markov-Chain Queuing Models	125
49 Queuing Network	127
 <b>XII Inventory Theory</b>	 <b>129</b>
 <b>XIII Reliability Theory</b>	 <b>131</b>
50 Maintenance Optimization	133
 <b>XIV Bayesian Statistic</b>	 <b>135</b>
 <b>XV Classical Statistic</b>	 <b>137</b>
 <b>XVI Simulation</b>	 <b>139</b>

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:

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

**Modeling:**

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

#### 4.2.2 A Minimax Objective

**Description:** Consider the following model statement:

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

**Modeling:**

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

### 4.2.3 A fractional Objective

**Description:** Consider the following model statement:

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

**Modeling:**

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

### 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

# 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



## Part IV

# Integer and Combinatorial Programming





# Chapter 18

## Formulation

### 18.1 Typical Problems

### 18.2 Integer Programming Formulation Skills

#### 18.2.1 A Variable Taking Discontinuous Values

**Description:** In algebraic notation:

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

**Modeling:**

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

where

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

#### 18.2.2 Fixed Costs

**Description:** In algebraic notation:

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

**Modeling:**

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

where

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

#### 18.2.3 Either-or Constraints

**Description:** In algebraic notation:

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

**Modeling:**

$$\begin{aligned}
\sum_{j \in J} a_{1j}x_j &\leq b_1 + M_1y \\
\sum_{j \in J} a_{2j}x_j &\leq b_2 + M_1(1 - y) \\
y &\in \{0, 1\}
\end{aligned}$$

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 " $+$ ".

### 18.2.4 Conditional Constraints

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

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

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  $\epsilon$  is a very small positive number.

**Modeling:**

$$\begin{aligned}
\sum_{j \in J} a_{1j}x_j &\geq b_1 + \epsilon - M_2y \\
\sum_{j \in J} a_{2j}x_j &\leq b_2 + M_2(1 - y) \\
y &\in \{0, 1\}
\end{aligned}$$

### 18.2.5 Special Ordered Sets

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

$$\begin{aligned}
x_1 = 1, x_2 = x_3 = \dots = x_n = 0 \\
\text{or} \\
x_2 = 1, x_1 = x_3 = \dots = x_n = 0 \\
\text{or } \dots
\end{aligned}$$

**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  $x_1, x_2, \dots, x_n$  is a SOS2, then

$$\begin{aligned}
\sum_{i=1}^n x_i &\leq 2 \\
x_i + x_j &\leq 1, \forall i \in \{1, 2, \dots, n\}, j \in \{i+2, i+3, \dots, n\} \\
x_i &\in \{0, 1\}
\end{aligned}$$

**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*

### 18.2.6 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  $(x_1, y_1), (x_2, y_2), \dots, (x_k, y_k)$ .

Denote

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

Then the objective function is

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

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

$$\begin{aligned} \sum_{i \in \{1, 2, \dots, k-1\}} d_i &= 1 \\ d_i &\in \{0, 1\}, i \in \{1, 2, \dots, k-1\} \\ x &= \sum_{i \in \{1, 2, \dots, k\}} w_i x_i \\ y &= \sum_{i \in \{1, 2, \dots, k\}} w_i y_i \\ w_1 &\leq d_1 \\ w_i &\leq d_{i-1} + d_i, i \in \{2, 3, \dots, k-1\} \\ w_k &\leq d_{k-1} \end{aligned}$$

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

### 18.2.7 Conditional Binary Variables

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

**Modeling:**

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

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

**Modeling:**

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

**Description:** Choose  $x_j$  only if  $x_k = 1$

**Modeling:**

$$x_j = x_k$$

**Description:** “and” condition, iff  $x_1, x_2, \dots, x_m = 1$  then  $y = 1$

**Modeling:**

$$\begin{aligned} y &\leq x_i, i \in \{1, 2, \dots, m\} \\ y &\geq \sum_{i \in \{1, 2, \dots, m\}} x_i - (m - 1) \end{aligned}$$

### 18.2.8 Elimination of Products of Variables

**Description:** For variables  $x_1$  and  $x_2$ ,

$$y = x_1 x_2$$

**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

$$y \leq ux_1$$

$$y \leq x_2$$

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

$$y \geq 0$$

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

## Chapter 19

# Branch and Bound



## Chapter 20

# Branch and Cut





## Chapter 21

# Packing and Matching



## Chapter 22

# Traveling Salesman Problem



## Chapter 23

# Knapsack Problem



Part V

**Nonlinear Programming**





## Chapter 24

# KKT Optimality Conditions



## Chapter 25

# Lagrangian Duality



## Chapter 26

# Unconstrained Optimization



## Chapter 27

# Penalty and Barrier Functions





## Part VI

# Algorithms and Computational Complexity



## Chapter 28

# Computational Complexity



## Chapter 29

# Sorting

29.1 Elementary Sorting Algorithms

29.2 Heap-sort

29.3 Quick-sort

29.4 Sorting in Linear Time

29.5 Medians and Order Statistics



## Chapter 30

# Data Structures

30.1 Elementary Data Structures

30.2 Hash Tables

30.3 Binary Search Trees

30.4 Red-Black Trees

30.5 B-Trees

30.6 Fibonacci Heaps

30.7 van Emde Boas Trees





## Chapter 31

# Design and Analysis Techniques

31.1 Dynamic Programming

31.2 Greedy Algorithms

31.3 Amortized Analysis

31.4 Multi-threaded Algorithms

31.5 Matrix Operations

31.6 Polynomials and the FFT

31.7 Number-Theoretic Algorithms

31.8 String Matching

31.9 Computational Geometry

31.10 NP-Completeness

31.11 Approximation Algorithms



## Part VII

# Heuristics and Meta-heuristics



Part VIII

Game Theory



## Chapter 32

# Games with Ordinal Payoffs

**32.1 Ordinal Games in Strategic Form**

**32.2 Perfect-information Games**

**32.3 General Dynamic Games**





## Chapter 33

# Games with Cardinal Payoffs

33.1 Expected Utility Theory

33.2 Strategic-form Games

33.3 Extensive-form Games



## Chapter 34

# Knowledge, Common Knowledge, Beliefs

34.1 Common Knowledge

34.2 Adding Beliefs to Knowledge

34.3 Rationality



## Chapter 35

# Refinements of Subgame-perfect Equilibrium

35.1 Weak Sequential Equilibrium

35.2 Sequential Equilibrium

35.3 Perfect Bayesian Equilibrium



## Chapter 36

# Incomplete Information

36.1 Static Games

36.2 Dynamic Games

36.3 The Type-Space Approach





Part IX

Decision Analysis



## Part X

# Probability, Stochastic Processes and Markov Chains



## Chapter 37

# Probability



# Chapter 38

## Random Variables

### 38.1 Relationship between Some Random Variables

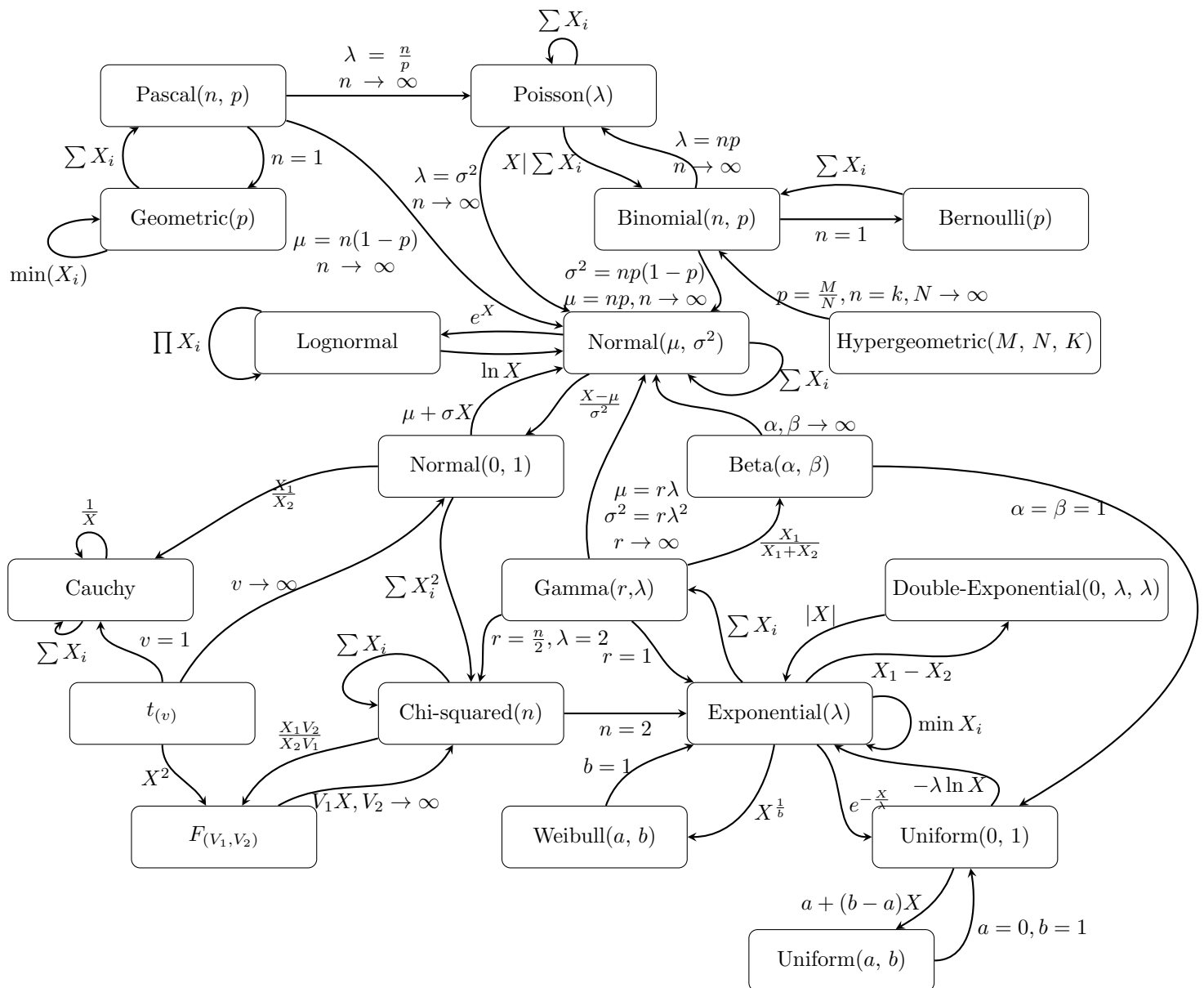


Figure 38.1: Relationship between Some Random Variables

## 38.2 Discrete Random Variables

Table 38.1: Discrete Random Variables

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

## 38.3 Continuous Random Variables

Table 38.2: 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( $\mu, \sigma$ )	$f(x) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $x \in \mathbb{R}$	$F(x) = \int_{-\infty}^x \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$ $x \in \mathbb{R}$	$E[X] = \mu$	$D[X] = \sigma^2$	$e^{\frac{t(\mu\sigma^2+2\mu)}{2}}$ $t \in \mathbb{R}$
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( $n, \lambda$ )	$f(x) = \frac{\lambda^n x^{n-1} e^{-\lambda x}}{(n-1)!}$ $x > 0$	$F(x) = 1 - \sum_{i=0}^{n-1} \frac{\lambda^i x^i e^{-\lambda x}}{i!}$ $x > 0$	$E[X] = \frac{n}{\lambda}$	$D[X] = \frac{n}{\lambda^2}$	$\frac{1}{(1-\frac{t}{\lambda})^n}$ $t < \lambda$



## Chapter 39

# Limit Theorems



## Chapter 40

# The Bernoulli and Poisson Process



## Chapter 41

# Discrete-Time Markov Chains



## Chapter 42

# Continuous-Time Markov Chains





Part XI

Queuing Theory



## Chapter 43

# Queuing Model



## Chapter 44

# Birth-and-Death Queuing Models



## Chapter 45

# Multidimensional Birth-and-Death Queuing Models





## Chapter 46

# Phase-Type Queue



## Chapter 47

# Bulk Queue



## Chapter 48

# Imbedded-Markov-Chain Queuing Models



## Chapter 49

# Queuing Network





Part XII

Inventory Theory



Part XIII

Reliability Theory



## Chapter 50

# Maintenance Optimization



Part XIV

Bayesian Statistic





Part XV

Classical Statistic



**Part XVI**

**Simulation**

