Operations Research Summary

Mathematical Notes

October 27, 2025

Contents

1 Linear Programming **3**
- 1.1 Standard Form . . . 3
- 1.2 Duality . . . 3
- 1.3 Duality Theorems . . . 3
- 1.4 Simplex Method . . . 3
- 1.5 Sensitivity Analysis . . . 3

2 Integer Programming **4**
- 2.1 Integer Linear Programming . . . 4
- 2.2 Branch and Bound . . . 4
- 2.3 Cutting Plane Method . . . 4
- 2.4 Gomory Cuts . . . 4

3 Network Optimization **4**
- 3.1 Minimum Cost Flow . . . 4
- 3.2 Shortest Path Problem . . . 4
- 3.3 Dijkstra's Algorithm . . . 4
- 3.4 Maximum Flow Problem . . . 5
- 3.5 Ford-Fulkerson Algorithm . . . 5
- 3.6 Assignment Problem . . . 5
- 3.7 Hungarian Algorithm . . . 5

4 Queuing Theory **5**
- 4.1 Kendall Notation . . . 5
- 4.2 Poisson Process . . . 6
- 4.3 M/M/1 Queue . . . 6
- 4.4 Little's Law . . . 6
- 4.5 M/M/c Queue . . . 6
- 4.6 Erlang's Loss Formula . . . 6

5 Dynamic Programming **6**
- 5.1 Principle of Optimality . . . 6
- 5.2 Bellman Equation . . . 7
- 5.3 Inventory Management . . . 7
- 5.4 Newsvendor Problem . . . 7

6 Game Theory **7**
6.1 Normal Form Games . . . 7
6.2 Nash Equilibrium . . . 7
6.3 Zero-Sum Games . . . 7
6.4 Minimax Theorem . . . 7

7 Stochastic Programming **8**
7.1 Two-Stage Stochastic Programming . . . 8
7.2 Sample Average Approximation . . . 8

8 Multi-Objective Optimization **8**
8.1 Pareto Optimality . . . 8
8.2 Weighted Sum Method . . . 8
8.3 Goal Programming . . . 8

9 Heuristic Methods **9**
9.1 Genetic Algorithms . . . 9
9.2 Simulated Annealing . . . 9
9.3 Tabu Search . . . 9

10 Applications **9**
10.1 Supply Chain Management . . . 9
10.2 Finance . . . 9
10.3 Healthcare . . . 10
10.4 Transportation . . . 10
10.5 Energy . . . 10

11 Important Algorithms **10**
11.1 Linear Programming . . . 10
11.2 Network Algorithms . . . 10
11.3 Integer Programming . . . 11

12 Key Theorems **11**
12.1 Farkas' Lemma . . . 11
12.2 Carathéodory's Theorem . . . 11
12.3 Separating Hyperplane Theorem . . . 11

13 Software and Tools **11**
13.1 Commercial Solvers . . . 11
13.2 Open Source Solvers . . . 11
13.3 Modeling Languages . . . 12

1 Linear Programming

1.1 Standard Form

Definition 1.1. A **linear programming problem** in standard form is:

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} && (1) \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b} && (2) \\ & \mathbf{x} \geq \mathbf{0} && (3) \end{aligned}$$

where $\mathbf{x} \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $\mathbf{b} \in \mathbb{R}^m$, and $\mathbf{c} \in \mathbb{R}^n$.

1.2 Duality

Definition 1.2. The **dual** of the primal problem:

$$\begin{aligned} \max \quad & \mathbf{c}^T \mathbf{x} && (4) \\ \text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} && (5) \\ & \mathbf{x} \geq \mathbf{0} && (6) \end{aligned}$$

is:

$$\begin{aligned} \min \quad & \mathbf{b}^T \mathbf{y} && (7) \\ \text{subject to} \quad & A^T\mathbf{y} \geq \mathbf{c} && (8) \\ & \mathbf{y} \geq \mathbf{0} && (9) \end{aligned}$$

1.3 Duality Theorems

Theorem 1.1 (Weak Duality)**.** If $\mathbf{x}$ is feasible for the primal and $\mathbf{y}$ is feasible for the dual, then $\mathbf{c}^T\mathbf{x} \leq \mathbf{b}^T\mathbf{y}$.

Theorem 1.2 (Strong Duality)**.** If either the primal or dual has an optimal solution, then both have optimal solutions and their optimal values are equal.

Theorem 1.3 (Complementary Slackness)**.** If $\mathbf{x}^*$ and $\mathbf{y}^*$ are optimal solutions to the primal and dual respectively, then:

$$x_i^*(A^T\mathbf{y}^* - \mathbf{c})_i = 0 \quad \text{and} \quad y_j^*(\mathbf{b} - A\mathbf{x}^*)_j = 0$$

1.4 Simplex Method

Definition 1.3. The **simplex method** is an iterative algorithm that moves from vertex to vertex of the feasible region to find the optimal solution.

1.5 Sensitivity Analysis

Definition 1.4. Sensitivity analysis studies how changes in the problem parameters affect the optimal solution.

2 Integer Programming

2.1 Integer Linear Programming

Definition 2.1. An **integer linear programming problem** is:

$$\begin{aligned}
\max \quad & \mathbf{c}^T\mathbf{x} && (10)\\
\text{subject to} \quad & A\mathbf{x} \leq \mathbf{b} && (11)\\
& \mathbf{x} \geq \mathbf{0} && (12)\\
& \mathbf{x} \in \mathbb{Z}^n && (13)
\end{aligned}$$

2.2 Branch and Bound

Definition 2.2. Branch and bound is a method for solving integer programming problems by systematically exploring the solution space.

2.3 Cutting Plane Method

Definition 2.3. The **cutting plane method** adds linear inequalities (cuts) to eliminate fractional solutions.

2.4 Gomory Cuts

Definition 2.4. Gomory cuts are cutting planes derived from the simplex tableau for integer programming.

3 Network Optimization

3.1 Minimum Cost Flow

Definition 3.1. The **minimum cost flow problem** is:

$$\begin{aligned}
\min \quad & \sum_{(i,j)\in E} c_{ij}x_{ij} && (14)\\
\text{subject to} \quad & \sum_{j:(i,j)\in E} x_{ij} - \sum_{j:(j,i)\in E} x_{ji} = b_i \quad \forall i \in V && (15)\\
& l_{ij} \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in E && (16)
\end{aligned}$$

where $G = (V, E)$ is a directed graph, c_{ij} are costs, b_i are supplies/demands, and l_{ij}, u_{ij} are lower and upper bounds.

3.2 Shortest Path Problem

Definition 3.2. The **shortest path problem** finds the minimum cost path from a source vertex to a destination vertex.

3.3 Dijkstra's Algorithm

Theorem 3.1 (Dijkstra's Algorithm)**.** For non-negative edge weights, Dijkstra's algorithm finds shortest paths in $O(|V|^2)$ time.

3.4 Maximum Flow Problem

Definition 3.3. The **maximum flow problem** is:

$$\max \quad \sum_{j:(s,j)\in E} x_{sj} \tag{17}$$

$$\text{subject to} \quad \sum_{j:(i,j)\in E} x_{ij} = \sum_{j:(j,i)\in E} x_{ji} \quad \forall i \in V \setminus \{s,t\} \tag{18}$$

$$0 \leq x_{ij} \leq u_{ij} \quad \forall (i,j) \in E \tag{19}$$

where s is the source and t is the sink.

3.5 Ford-Fulkerson Algorithm

Theorem 3.2 (Max-Flow Min-Cut Theorem)**.** The maximum flow value equals the minimum cut capacity.

3.6 Assignment Problem

Definition 3.4. The **assignment problem** assigns n tasks to n workers to minimize total cost:

$$\min \quad \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \tag{20}$$

$$\text{subject to} \quad \sum_{j=1}^{n} x_{ij} = 1 \quad \forall i \tag{21}$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \forall j \tag{22}$$

$$x_{ij} \in \{0,1\} \tag{23}$$

3.7 Hungarian Algorithm

Definition 3.5. The **Hungarian algorithm** solves the assignment problem in $O(n^3)$ time.

4 Queuing Theory

4.1 Kendall Notation

Definition 4.1. The **Kendall notation** $A/B/c/K/m/Z$ describes a queuing system where:

- A: arrival process
- B: service time distribution
- c: number of servers
- K: system capacity
- m: population size
- Z: service discipline

4.2 Poisson Process

Definition 4.2. A **Poisson process** with rate λ has:

- Inter-arrival times are exponentially distributed with mean $1/\lambda$
- Number of arrivals in time t is Poisson distributed with mean λt

4.3 M/M/1 Queue

Definition 4.3. An **M/M/1 queue** has Poisson arrivals, exponential service times, and one server.

For M/M/1 with arrival rate λ and service rate μ:

- Traffic intensity: $\rho = \lambda/\mu$
- Steady-state probability of n customers: $p_n = \rho^n(1-\rho)$
- Average number in system: $L = \rho/(1-\rho)$
- Average waiting time: $W = 1/(\mu - \lambda)$

4.4 Little's Law

Theorem 4.1 (Little's Law)**.** For a stable queuing system:

$$L = \lambda W$$

where L is the average number of customers, λ is the arrival rate, and W is the average time in the system.

4.5 M/M/c Queue

Definition 4.4. An **M/M/c queue** has c servers with Poisson arrivals and exponential service times.

4.6 Erlang's Loss Formula

Theorem 4.2 (Erlang's Loss Formula)**.** For M/M/c/c (loss system):

$$P(\text{loss}) = \frac{(\lambda/\mu)^c/c!}{\sum_{k=0}^{c}(\lambda/\mu)^k/k!}$$

5 Dynamic Programming

5.1 Principle of Optimality

Definition 5.1. The **principle of optimality** states that an optimal policy has the property that whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision.

5.2 Bellman Equation

Definition 5.2. The **Bellman equation** for dynamic programming is:

$$V(s) = \max_a \left\{ R(s,a) + \gamma \sum_{s'} P(s'|s,a)V(s') \right\}$$

where $V(s)$ is the value function, $R(s,a)$ is the reward, and γ is the discount factor.

5.3 Inventory Management

Definition 5.3. The **economic order quantity (EOQ)** model minimizes total inventory costs:

$$Q^* = \sqrt{\frac{2DS}{H}}$$

where D is demand rate, S is setup cost, and H is holding cost per unit per time.

5.4 Newsvendor Problem

Definition 5.4. The **newsvendor problem** determines optimal order quantity when demand is uncertain:

$$F(Q^*) = \frac{p-c}{p-s}$$

where p is selling price, c is cost, s is salvage value, and F is the demand distribution function.

6 Game Theory

6.1 Normal Form Games

Definition 6.1. A **normal form game** consists of:

- Players: $N = \{1, 2, \dots, n\}$
- Strategy sets: S_i for each player i
- Payoff functions: $u_i : S_1 \times S_2 \times \cdots \times S_n \to \mathbb{R}$

6.2 Nash Equilibrium

Definition 6.2. A **Nash equilibrium** is a strategy profile $(s_1^*, s_2^*, \dots, s_n^*)$ such that for each player i:

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i \in S_i$$

6.3 Zero-Sum Games

Definition 6.3. A **zero-sum game** satisfies $\sum_{i=1}^n u_i(s) = 0$ for all strategy profiles s.

6.4 Minimax Theorem

Theorem 6.1 (Minimax Theorem)**.** In a zero-sum game, the minimax value equals the maximin value:

$$\min_y \max_x x^T A y = \max_x \min_y x^T A y$$

7 Stochastic Programming

7.1 Two-Stage Stochastic Programming

Definition 7.1. A **two-stage stochastic program** is:

$$\begin{aligned} \min \quad & \mathbf{c}^T\mathbf{x} + \mathbb{E}[Q(\mathbf{x}, \boldsymbol{\xi})] && (24) \\ \text{subject to} \quad & A\mathbf{x} = \mathbf{b} && (25) \\ & \mathbf{x} \geq \mathbf{0} && (26) \end{aligned}$$

where $Q(\mathbf{x}, \boldsymbol{\xi}) = \min\{\mathbf{q}^T\mathbf{y} : T\mathbf{x} + W\mathbf{y} = \mathbf{h}(\boldsymbol{\xi}), \mathbf{y} \geq \mathbf{0}\}$.

7.2 Sample Average Approximation

Definition 7.2. **Sample average approximation** replaces the expected value with a sample average:

$$\mathbb{E}[Q(\mathbf{x}, \boldsymbol{\xi})] \approx \frac{1}{N}\sum_{i=1}^{N} Q(\mathbf{x}, \boldsymbol{\xi}_i)$$

8 Multi-Objective Optimization

8.1 Pareto Optimality

Definition 8.1. A solution $\mathbf{x}^*$ is **Pareto optimal** if there does not exist another solution $\mathbf{x}$ such that:

- $f_i(\mathbf{x}) \leq f_i(\mathbf{x}^*)$ for all i
- $f_j(\mathbf{x}) < f_j(\mathbf{x}^*)$ for at least one j

8.2 Weighted Sum Method

Definition 8.2. The **weighted sum method** converts multi-objective optimization to single-objective:

$$\min \sum_{i=1}^{k} w_i f_i(\mathbf{x})$$

where $w_i \geq 0$ and $\sum_{i=1}^{k} w_i = 1$.

8.3 Goal Programming

Definition 8.3. **Goal programming** minimizes deviations from target values:

$$\min \sum_{i=1}^{k} (d_i^+ + d_i^-)$$

subject to $f_i(\mathbf{x}) + d_i^- - d_i^+ = g_i$ where g_i is the goal for objective i.

9 Heuristic Methods

9.1 Genetic Algorithms

Definition 9.1. Genetic algorithms use evolutionary principles to find good solutions:

1. Initialize population
2. Evaluate fitness
3. Select parents
4. Create offspring through crossover and mutation
5. Replace population
6. Repeat until convergence

9.2 Simulated Annealing

Definition 9.2. Simulated annealing accepts worse solutions with probability $e^{-\Delta E/T}$ where ΔE is the energy difference and T is the temperature.

9.3 Tabu Search

Definition 9.3. Tabu search maintains a list of forbidden moves to avoid cycling and explore new regions of the solution space.

10 Applications

10.1 Supply Chain Management

- Facility location
- Transportation planning
- Inventory optimization
- Production scheduling

10.2 Finance

- Portfolio optimization
- Risk management
- Asset allocation
- Option pricing

10.3 Healthcare

- Resource allocation
- Appointment scheduling
- Emergency response
- Treatment planning

10.4 Transportation

- Vehicle routing
- Traffic flow optimization
- Public transit planning
- Logistics management

10.5 Energy

- Power generation scheduling
- Grid optimization
- Renewable energy integration
- Demand response

11 Important Algorithms

11.1 Linear Programming

- Simplex method
- Interior point methods
- Dual simplex method
- Revised simplex method

11.2 Network Algorithms

- Dijkstra's algorithm
- Bellman-Ford algorithm
- Floyd-Warshall algorithm
- Ford-Fulkerson algorithm
- Hungarian algorithm

11.3 Integer Programming

- Branch and bound
- Cutting plane methods
- Branch and cut
- Lagrangian relaxation

12 Key Theorems

12.1 Farkas' Lemma

Theorem 12.1 (Farkas' Lemma)**.** Exactly one of the following systems has a solution:

1. $A\mathbf{x} = \mathbf{b}$, $\mathbf{x} \geq \mathbf{0}$
2. $A^T\mathbf{y} \leq \mathbf{0}$, $\mathbf{b}^T\mathbf{y} > 0$

12.2 Carathéodory's Theorem

Theorem 12.2 (Carathéodory's Theorem)**.** If $\mathbf{x}$ is in the convex hull of $S \subset \mathbb{R}^n$, then $\mathbf{x}$ is in the convex hull of at most $n + 1$ points of S.

12.3 Separating Hyperplane Theorem

Theorem 12.3 (Separating Hyperplane Theorem)**.** If C and D are disjoint convex sets, then there exists a hyperplane that separates them.

13 Software and Tools

13.1 Commercial Solvers

- CPLEX
- Gurobi
- Xpress
- MOSEK

13.2 Open Source Solvers

- GLPK
- COIN-OR
- SCIP
- OR-Tools

13.3 Modeling Languages

- AMPL
- GAMS
- OPL
- Pyomo