IE 472 Operations Research II

UIC Mechanical & Industrial Engineering Department

Spring 2003

Instructor: Dr. Wei Chen

Course Objective

- Continue to introduce the methods of Operations Research
- Emphasize the mathematical procedures of nonlinear programming search techniques
- Introduce advanced topics such as probabilistic models (Markov chain & queuing theory) and dynamic programming
- Relate the course material to research activities (graduate students)

What is Operations Research (OR)?

A scientific approach to decision making, which seeks to determine how **best** to design and operate a system, usually under conditions requiring the **allocation of scarce resources**.

(Back to World War II, the application of mathematics and the scientific method to military operations was called operations research).

Optimization

- Optimization is derived from the Latin word "optimus", **the best**.
- Optimization characterizes the activities involved to find "the best".
- People have been "optimizing" forever, but the roots for modern day optimization can be traced to the Second World War (the born of Operations Research).

Optimization is a process of achieving the *best* outcome of given objectives while *satisfying certain restrictions*.

Basic Formulation of Optimization

Find: The optimal solution x

Subject to: Constraints $g(x) \leq 0$

or $g(x)$ ($\leq$, $=$, or $\geq$) b

Bounds $x \in R^n$ or $x^l \leq x \leq x^u$

Min or Max: objective function $z = f(x)$

Example Formulations

1. Design a beer can that can hold at least 400 cm^3 volume of beer. The objective is to reduce the cost of manufacturing the can.
2. If *K* units of capital and *L* units of labor are used, a company can produce *K L* units of a manufactured good. Capital can be purchased at \$4/unit and labor can be purchased at \$1/unit. A total of \$8 is available to purchase capital and labor. How can the firm maximize the quantity of the goods that can be manufactured?

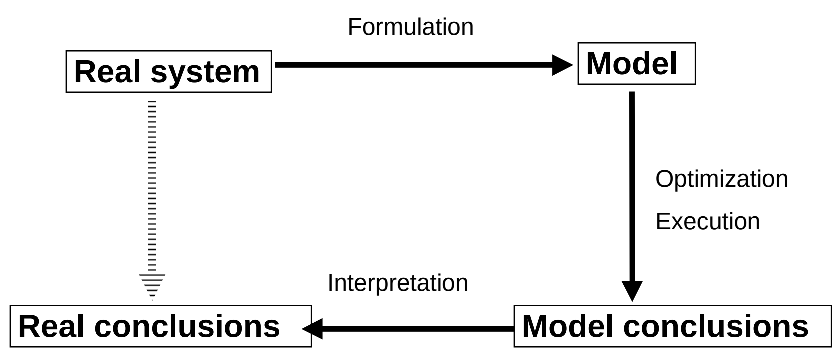
Optimization Theory

A body of ***mathematical results*** and ***numerical methods*** for finding and identifying the best candidate from a collection of alternatives without having to explicitly enumerate and evaluate all possible alternatives.

Major activities in an optimization process include:

- **Modeling**: formulating the problem using mathematical model.
- **Solution search**: searching for the optimal solution in an efficient manner.
- **Validation**: acquiring the conviction that a model actually works.

The Role of Mathematical Models



The Phrase “Programming”

- mathematical programming
- linear programming
- nonlinear programming
- mixed integer programming

In the early days, a set of values which represented a solution to a problem was referred to as a "program".

LP vs. NLP

	LP	NLP
FEASIBLE REGION	Convex	Convex or nonconvex
OPTIMUM SOLUTION	Extreme points	Not necessary extreme points or at boundary
ADDITIVITY PROPERTION-ALITY	yes	May or may not

Classifications of Optimization Problems

- **By model**: single variable or multiple variable, constrained or unconstrained, linear or nonlinear problem
- **By solution method**: Linear programming, Sequential quadratic programming, integer programming, golden section, steepest descent, geometry programming, dynamic programming, etc.
- **By derivatives**: gradient-based method, nongradient based (random, simplex method, Genetic Algorithms) etc.
- **By nature of information**: deterministic vs. probabilistic models

Advanced Topics

- Geometry Programming - Graphical representation is used to help creating the mathematical model (Transportation & Network problems).
- Markov Chains- Modeling of stochastic process which includes random variables that evolves over (discrete) time (finance, marketing, production models).
- Queuing Theory- modeling arrival-service ("waiting in lines") (bank teller service, pizza delivery).
- Dynamic Programming- a technique to solve optimization problems by breaking large problems into smaller tractable problems (network, inventory problems).

Mathematical Concepts for NLP

- Local & global optimums
- Convex & nonconvex sets
- Convex, concave, nonconvex (or nonconcave) functions
- Theorem: Suppose the feasible region *S* for NLP is a *convex* set. If f(x) is *convex/concave* on *S*, then any local *minimum/maximum* is an (global) optimal solution to this NLP.

When is f(x) convex/concave?

- Single variable x, for all x in S
  - convex, if $f''(x) \geq 0$
  - concave, If $f''(x) \leq 0$
  - Otherwise, nonconvex (nonconcave)
- Multiple variables x, for all x in S
  - convex, if Hessian of f(x) is positive definite or positive semidefinite.
  - Concave, if -f(x) is convex, or Hessian of f(x) is negative definite or negative semidefinite
  - otherwise, nonconvex

Theorems 3 and 3' (p.657)

- Theorem 3: f(x) is a convex function if and only if for all x, all principal minors of H are nonnegative.
- Theorem 3': f(x) is a concave function if and only if for all x and k = 1, 2, …n, all nonzero principal minors have the same sign as $(-1)^k$.

Definitions of Principal Minors

- The *i*th **principal minor** of an *n* x *n* matrix is the determinant of any *i* x *i* matrix obtained by deleting *n* - *i* rows and the corresponding columns of the matrix.
- The *k*th **leading principal minor** of an *n* x *n* matrix is the determinant of the *k* x *k* matrix obtained by deleting the **last** *n* - *k* rows and columns of the matrix.

Conditions for Positive Definite Hessian (H)

- All the eigenvalues are positive.
- Since H is symmetric, we could judge by whether
  - All diagonal elements are positive.
  - The leading principal minors (determinates) are positive.
- Positive semidefinite
  - All diagonal elements are nonnegative.
  - All the principal determinants are nonnegative.
  - Some of the eigenvalues are zeros.

Conditions for Negative Definite Hessian (H)

- Negative definite:
  - All diagonal elements are negative.
  - The leading principal minors (determinates) have the sign of $(-1)^k$, k stands for kth leading principle minor.
  - Or test the negative of that matrix for positive definite.
- Indefinite
  - at least two of its diagonal elements are of the opposite sign.

Single Variable Unconstrained NLP

- Case 1: $f'(x) = 0$, $a < x < b$
- Case 2: $f'(x_0)$ does not exist
- Case 3: Endpoints of a and b of [a, b]

Conditions for Optimality (Case 1)

- If $f'(x_0) = 0$
  - $f''(x_0) < 0$, local maximum
  - $f''(x_0) > 0$, local minimum
- If $f'(x_0) = 0$ and $f''(x_0) = 0$, first nonzero high derivative denoted by n
  - If n is odd, x_0 is a point of inflection.
  - If n is even
    - if $f^n(x_0) < 0$, local maximum
    - if $f^n(x_0) > 0$, local minimum

Region Elimination Method

Search methods that successively eliminating subintervals so as to reduce the remaining interval of search.

- Or called "line search techniques" which are in essence optimization algorithms for one-dimensional minimization problems.
- They are often regarded as the backbones of nonlinear optimization algorithms.
- Typically, two phases: Bounding and Interval Refinement.
- Often, unimodality is assumed.

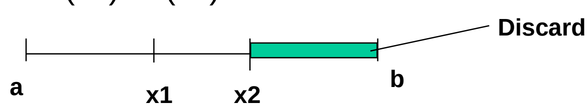
- **Advantages**
  - No derivative calculation, only function evaluation
- **Disadvantages**
  - Large number of function evaluations

Concept of Golden Section

Based on the golden section ratio that commonly occurs in nature

– First find bounds [a,b] on the minimum, then pick two interior points such that the bounds are reduced as rapidly as possible.

If $f(x1) > f(x2)$

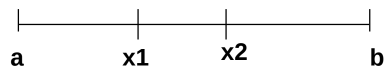


$x_2 - a = r(b-a)$; $b - x_1 = r(b-a)$ $r=0.618$

New interval [a, x2), x_1 becomes new right-hand point

Golden Section Algorithm (MAX Problem)

```
Initialize:
x1 = a + (b-a)*0.382
x2 = a + (b-a)*0.618
f1 = ƒ(x1)
f2 = ƒ(x2)
Loop:
if f1 < f2 eliminate [a, x1], new interval (x1, b]
    a = x1; x1 = x2; f1 = f2
    x2 = a + (b-a)*0.618
    f2 = ƒ(x2)
else  (f1 ≥ f2)
eliminate [x2, b], new interval [a, x2]
    a=a; b = x2; x2 = x1; f2 = f1
    x1 = a + (b-a)*0.382
    f1 = ƒ(x1)
endif
```



Note: a, b, x1, x2, f1, f2 are updated in each loop.

Search requires only one new point at each loop.

Stop when interval is small enough

Multiple Variable Unconstrained NLP-Optimality Condition

- $\nabla f(x^*) = 0$ (x^* called a stationary point)
  - H(x*) negative definition, local maximum
  - H(x*) positive definite, local minimum
  - H(x*) indefinite, saddle point, no optimum.

Definition of Leading Principle Minor

- For a Hessian matrix with dimension of *n x n*, *k*th leading principal minor $H_k(x)$ stands for the determinant of *k x k* maxtrix obtained by deleting the **last** *n-k* rows and columns of the matrix

Check the Condition by the Leading Principle Minor

For $\nabla f(x^*) = 0$

- If $H_k(x^*) > 0$, k=1,2,…n, then x* is a local minimum (positive definite).
- If for k=1,2,…n, $H_k(x^*)$ is nonzero and has the same sign as $(-1)^k$, x* is a local maximum (negative definite).
- If $H_n(x^*) \neq 0$, and the above two conditions don't hold, x* is not a local extremum (saddle point).
- If $H_n(x^*) = 0$, x* may be a local min, max or saddle.

General Procedure of Optimization Algorithms

- Optimization seeks to find a perturbation to an existing solution which will lead to an improvement.
- Most optimization algorithms apply a two-step process
  $\mathbf{X}^{k+1} = \mathbf{X}^k + \delta^k \mathbf{d}^k$ (k - kth iteration)
  - $\mathbf{d}^k$ is the **search direction** that will improve the current solution.
  - δ^k represents the **step size** in the search direction.
- The users always need to choose an initial design X^0 to start.
- The process will iterate (k=1, 2.. N) until convergence.

Properties of Surface Gradient

- $\nabla f(x^0)$ is orthogonal (normal) to the tangent plane for the surface $f(x) = C$, where $f(x^0)$ is a point on this surface.
- Gradient represents a direction of maximum rate of increase for the function f(x) at the point x^0.
  - The function increases most rapidly along the gradient.

Algorithm of “Steepest Ascent” (For Maximization)

- Basic Steps in each iteration k
  - calculate the gradient
  - set the search direction as $d^k = \nabla f(x^k)$
  - Search the best step size δ^k in the direction of d^k which *maximizes* the function $f(\delta^k)$.
- For minimization, the algorithm is called “steepest descent”
  - $d^k = -\nabla f(x^k)$
  - δ^k is sought to *minimize* the function.

Features of “Steepest Ascent”

- Successive directions are orthogonal to each other.
- The method is locally convergent by assuring $f(X^{k+1}) > f(X^k)$.
- Rate of converge is slow (one reason is because only the first-order information about the function is used). The rate is related to the condition number of Hessian at the optimum.
- Information calculated at the previous iterations is not used, which is inefficient.
- f is substantially increased in the first several steps.
- The direction of $\nabla f(x^k)$ is good in a local sense, but not in a global sense.

Measures for a Good Algorithm

- **Robustness –**must be reliable and be able to converge to the solution point starting from any given starting point.
- **Generality –** should not impose restrictions on the model's constraints and objective functions.
- **Accuracy –** ability to converge to precise mathematical optimum point is important, though it may not be required in practice.
- **Ease of use –** by both experienced and inexperienced users. Should not have problem dependent tuning parameters.
- **Efficiency –**1) a faster rate of convergence requiring fewer iterations, and 2) least number of calculations within one iteration.

Fletcher-Reeves Conjugate Gradient Method (for MAX)

1. Set iteration counter, k=1
2. Calculate the gradient, $\nabla f(x^k)$
3. If k=1 then $d^k = \nabla f(x^k)$

   Else $\beta = \dfrac{|\nabla f(x^k)|^2}{|\nabla f(x^{k-1})|^2}$ $\quad d^k = \nabla f(x^k) + \beta d^{k-1}$

4. Search the best δ in the direction of d^k
5. Set k=k+1
6. Repeat from step 2 until convergence to the optimum.

Features of the Conjugate Gradient Method

Successive directions are not orthogonal to each other. Directions tend to cut diagonally through the orthogonal steepest ascent directions.

Easy to Program

Converges in N or fewer iterations for quadratic problems

Simple addition to the steepest ascent method but has dramatically better performance

A Constrained Problem

A company is planning to spend \$10,000 on advertising. It costs \$3000 per minute to advertise on television and \$1000 per minute to advertise on radio. If the firm buys x minutes of television advertising and y minutes of radio advertising, its revenue in thousands of dollars is given by $f(x, y) = -2x^2 - y^2 + xy + 8x + 3y$. How can the firm maximize its revenue? (Ex. 22, p.687).

Optimal Condition for Equality-Constrained Problem

Max $f(x)$

s.t. $g_i(x) = b_i$, $i = 1, m$

Stationary Condition:

Let $L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i [b_i - g_i(x)]$

$$\frac{\partial L}{\partial x} = 0 \quad \text{i.e.,} \quad \nabla f(x^*) - \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) = 0$$

and $g_i(x^*) = b_i$, $i = 1, m$

Optimality Condition:

Hessian H_L, positive definite, local minimum.

negative definite, local maximum.

Derived Optimality Conditions

- For maximization problem, if f(x) is a concave function and each $g_i(x)$ is a linear function, then the stationary point will yield an optimal solution (Theorem 8).
- For minimization problem, if f(x) is a convex function and each $g_i(x)$ is a linear function, then the stationary point will yield an optimal solution (Theorem 8').

Interpretation of Lagrange Multipliers

- Geometrical Interpretation
  - Function gradient is a linear combination of the constraints' gradients at the optimal point.

$$\nabla f(x^*) = \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*)$$

- Economic Interpretation
  - The rate of change of the optimal value f w.r.t. to b_i is given by the optimal value of λ_i (shadow price).

$$\frac{\partial f^o}{\partial b_i} = \lambda_i^o$$

Kuhn-Tucker Conditions

Max f(x)

s.t. $g_i(x) \le b_i$, i = 1, m

Stationary Condition (K-T conditions):

$$L(x,\lambda) = f(x) + \sum_{i=1}^{m} \lambda_i [b_i - g_i(x)]$$

$$\frac{\partial L}{\partial x} = 0 \quad \text{i.e.,} \quad \nabla f(x^*) - \sum_{i=1}^{m} \lambda_i \nabla g_i(x^*) = 0$$

and $\lambda_i[b_i - g_i(x^*)] = 0$, i = 1, m; $\lambda_i \ge 0$; $g_i(x^*) \le b_i$.

Optimality Condition:

Hessian H_L, positive definite, local minimum.

negative definite, local maximum.

Derived Optimality Conditions

- For maximization problem, if f(x) is a *concave* function and each $g_i(x)$ is a *conve*x function, then those points satisfy K-T condition is optimal (Theorem 11).
- For minimization problem, if f(x) is a *convex* function and each $g_i(x)$ is a *conve*x function, then those points satisfy K-T condition is optimal (Theorem 11').

Example 25 (p.698)

A monopolist can purchase up to 17.25 oz of a chemical for $10/oz. At a cost of $3/oz, the chemical can be processed into an ounce of product 1, or at a cost of $5/oz, the chemical can be processed into an ounce of product 2. If x1 oz of product 1 are produced, it sells for a price of $30-x1 per ounce. If x2 oz of product 2 are produced, it sells for price of $50-2x2 per ounce. Determine how the monopolist can maximize the profit.

Interpretation of Lagrange Multipliers (K-T condition)

- Economic Interpretation
  - The rate of change of the optimal value *f* w.r.t. to b_i of *an active constraint* is given by the optimal value of λ_i (shadow price). $\frac{\partial f^o}{\partial b_i} = \lambda_i^o$
- Geometrical Interpretation
  - Function gradient is a linear combination of the *active* constraints' gradients at the optimal point.

$$\nabla f(x^*) = \sum_{i=1}^{k} \lambda_i \nabla g_i(x^*)$$ k - # of active constraints

Special Issues

- Minimization instead of maximization

$$L(x,\lambda) = f(x) - \sum_{i=1}^{m} \lambda_i [b_i - g_i(x)]$$

- NLP in which variables must be nonnegative
  - $x_i \geq 0$.
- Constraint qualification - regularity condition
  - K-T condition may fail at an irregular point, though the point may be optimal!
  - It happens when the gradients of all the active constraints at x* are linearly independent.

Extended K-T conditions

Max $f(x)$

s.t. $h_k(x) = b_k$

$g_i(x) \le b_i$

$$L(x,\lambda) = f(x) + \sum \lambda_k [b_k - h_k(x)] + \sum u_i [b_i - g_i(x)]$$

$$\frac{\partial L}{\partial x} = 0 \qquad \nabla f(x^*) - \sum \lambda_k \nabla h_k(x^*) - \sum u_i \nabla g_i(x^*) = 0$$

and $u_i[b_i - g_i(x^*)] = 0$;

$u_i \ge 0$; (no restriction of the sign of λ)

$h_k(x^*) = b_k$;

$g_i(x^*) \le b_i$;

Notes on the use of K-T conditions

- K-T conditions can be used to check or find *candidate* optimal points.
  - Need to check all equality and inequality constraints for *feasibility*.
  - Calculate all Lagrange multipliers and ensure those for inequality constraints are *nonnegative*.
- K-T condition may fail at an irregular point, though the point could be optimal.
- Whether it is truly maximum or minimum depends on H_L (convexity of f , g, and h).

Classification of Algorithms for NLP

- **Transformation methods** – convert a constrained optimization problem to a sequence of unconstrained optimization problems, e.g., **penalty function** method.

- **Primal Method** – a search method that works on the original problem directly by searching through the feasible region for the optimal solution.

  e.g., feasible direction method, generalized reduced gradient (GRG) method, sequential linear (or) quadratic programming, etc.

Penalty Function Method

Penalty methods use a mathematical function that will increase the objective for any given constrained violation.

For minimize f(x), s.t. $g(x) \leq 0$ and $h(x) = 0$

General transformation of constrained problem into an unconstrained problem

$$\mathbf{T(x, R) = f(x) + P(R, g(x), h(x))}$$

where

f(x) is the original function of the constrained problem.

R is a penalty scalar

P is a penalty function that imposes penalty for feasibility

T(x,R) is a transformed objective function.

A typical penalty function:

$$\mathbf{P(R,\ g(x),\ h(x)) = R\left\{\sum_{i=1}^{p}[h_i(x)]^2 + \sum_{i=1}^{m}[g_i^{+}(x)]^2\right\}}$$

where $g_i^+(x) = \max(0, g_i(x))$

Feasible Direction Method

Iterations: $\mathbf{X}^{k+1} = \mathbf{X}^k + \delta^k \mathbf{d}^k$

Emphasis: determining a search direction which will rapidly reduce the objective function while maintaining a feasible direction.

Compared to steepest ascent method: constraints must be considered when determining both δ^k (step size) and $\mathbf{d}^k$ (search direction).

Convergence Criteria:

1. Function must increase at each iteration

   $f(\mathbf{X}^{k+1}) \geq f(\mathbf{X}^k)$

2. New point ($\mathbf{X}^{k+1}$) is feasible.

Feasible Direction Method for NLP with Linear Constraints

Max $z = f(x)$

s.t. $A x \leq b;\ x \geq 0$

1. Pick a starting point $\mathbf{X}^0$ which is feasible
2. In k iteration, move away from $\mathbf{X}^k$ in the direction of ($\mathbf{d}^k$ - $\mathbf{X}^k$), d^k (d) is identified by solving the LP problem

   Max $z = \nabla f(x^k) \bullet d$

   s.t. $A\, d \leq b;\ d \geq 0$

   (The above problem is used to identify a feasible direction that achieves the maximum local increase of the objective function).

   $\mathbf{X}^{k+1} = \mathbf{X}^k + t\,(\mathbf{d}^k - \mathbf{X}^k)$. Step size t is identified by solving

   Max $f(\mathbf{X}^k + t\,(\mathbf{d}^k - \mathbf{X}^k))$

   s.t., $0 \leq t \leq 1$
3. Repeat the iterations until the difference $\mathbf{X}^{k+1} - \mathbf{X}^k$ is very small.

Quadratic Programming

- The type of problem
  - NLP whose constraints are linear and each term in objective having a degree of 2, 1, or 0.
- The K-T conditions can be solved by linear programming.
- When using Simplex Algorithm, never perform a pivot that makes the two variables in complementary slackness conditions both positive.

K-T Necessary and Sufficient Conditions

- K-T necessity theorem:
  - Assume the regularity condition is satisfied, if x* is an optimal solution, then the K-T condition must be satisfied at this point.
- K-T sufficiency theorem:
  - Theorem 11 and 11'. The optimum is a global optimum.
- Second-order optimality condition:
  - Let x* satisfy the first order K-T necessary conditions. Let there be nonzero feasible directions ($d\neq 0$) satisfying
    - $\nabla h_i^T d = 0$ (for all equality constraints) and
    - $\nabla g_i^T d = 0$ (for all active inequalities).
  - Then x* is a local minimum if $Q>0$, where $Q=d^T H_L(x^*)d$, or x* is a local maximum if $Q<0$.

Network Models

- Some basic concepts
  - Nodes and arcs (ordered pair of nodes)
  - Chain and Path
    - Chain: every arc has exactly one node in common with the previous arc.
    - Path: for each arc, terminal node is identical to the initial node of next arc.
- Shortest Path Problem (SPP)
  - to minimize the length of path
  - can be modeled as a balanced transshipment problem.

Maximum Flow Problems

- Features:
  - Each arc has a capability limit
  - Objective is to transport the maximum amount of flow.
- Two sets of constraints:
  - Flow through each arc ≤ arc capability
  - flow into node i = flow out of node i.

CPM (Critical Path Method)

- What is the usage?
  - Determine the length of time to complete a project.
  - Answer the question: How long each activity can be delayed without delaying the completion of the project?
- Conditions
  - Project is considered to be completed when all the activities are completed.
  - Some activities (predecessors) must be completed before others begin.

Project Network Representation

- Representation
  - Arc: activity
  - Nodes (event): completion of a set of activities
  - AOA network (activity on arc network)
- Rules:
  - Numbering of nodes must follow the sequence of the activities.
  - An activity should not be represented by more than one arc.
  - Two nodes can be connected by at most one arc.

Example 6 (p.417)

Production of a new product

Activity	Predecessors	Duration (days)
A= train workers	--	6
B= purchase raw material	--	9
C= produce product 1	A, B	8
D= produce product 2	A, B	7
E= test product 2	D	10
F= assemble products 1&2	C, E.	12

Definitions Associated with CPM

- Early Event Time
  - Earliest time at which event corresponding to node i can occur.
  - ET (i) = maximum of [ET (j) + duration of the activity from j to i], j are the immediate predecessors of node i.
- Late Event Time
  - Latest time at which event corresponding to node i can occur without delaying the completion of the project.
  - LT (i) = minimum of [LT (j) - duration of the activity from i to j], j are the immediate successors of node i.
- If ET (i) = LT (i), any delay in i will delay the whole project.

Definitions Associated with CPM (continued)

- Total float
  - by which the duration of the activity can be increased without delaying the completion of the project.
  - TF (i, j) = LT (j) - ET(i) - t_{ij}
- Free float
  - the amount by which the starting time of the activity can be delayed without delaying the start of any later activities beyond its earliest possible staring time.
  - FF (i, j) = ET (j) - ET(i) - t_{ij}
- Critical activity: any activity with a total float of zero.

Critical Path

- Critical Path
  - A path from node 1 to the finish node that consists entirely of critical activities.
  - It is the longest path from the start node to the finish node.
- Using LP to find LP
  - x_j = the time that the event corresponding to node j occurs
  - For each activity (i, j), $x_j \geq x_i + t_{ij}$

Minimum Cost Network Flow Problems (MCNFP)

- The transportation, assignment, transshipment, shortest path, maximum flow, and CPM problems are all special cases of the MCNFP.
- MCNFP can be written as

$$\text{Min} \sum_{all-arcs} c_{ij} x_{ij}$$

s.t.

$$\sum_j x_{ij} - \sum_k x_{ki} = b_i \quad \text{(for each node i)}$$

$$L_{ij} \le x_{ij} \le U_{ij} \quad \text{(for each arc in the network)}$$

Definitions in MCNFP

- x_{ij} - number of units of flow sent from node i to node j through arc (i, j).
- b_i - net supply (outflow - inflow) at node i.
- c_{ij} - cost of transporting 1 unit of flow from node i to node j via arc (i, j).
- L_{ij} - lower bound on flow through arc (i, j). (If no lower bound, let $L_{ij} = 0$).
- U_{ij} - upper bound on flow through arc (i, j).

Concepts Related to Stochastic Process

- Random Variable x
  - there is uncertainty about x
  - defined as a function that associates a number with each point in the experiment's sample space
  - x can be discrete or continuous.
- Stochastic Process
  - study how random variable x changes with time.
- Discrete-Time Stochastic Process
  - description of the relation between x_0, x_1, .., x_i, i =t.

Markov Chain

- A discrete-time stochastic process is a Markov chain if, for t=0,1,2,… and all states:
  $P\,(X_{t+1} = i_{t+1} \mid X_t = i_t\,, X_{t-1} = i_{t-1}\,, \ldots X_0 = i_0)$
  $= P\,(X_{t+1} = i_{t+1} \mid X_t = i_t\,)$.
- Indication: probability distribution of the state at time t+1 depends only on the state at time t.
- Stationary Markov chain: $P\,(X_{t+1} = j \mid X_t = i) = P_{ij}$ is independent of t.
- Transition Probability Matrix ***P***. Its element P_{ij} is the probability that given state i at time t, it will be in a state j at time t+1.
- The sum of P_{ij} in a row of ***P*** is equal to 1, or $\sum_{j=1}^{s} P_{ij} = 1$.

n-step Transition Probabilities

- Answer the question:
  - Given a Markov Chain with a known transition probability matrix P, if a Markov chain is in state *i* at time *m*, what is the probability that *n* periods later, it will be in state *j*?
- We may write:
  - $P\,(X_{m+n} = j \mid X_m = i) = P\,(X_n = j \mid X_0 = i) = P_{ij}\,(n)$.
  - $P_{ij}(n)$ is called n-step probability.
  - $P_{ij}\,(n)$ = *ij*th element of P^n.

Classification of States in M.C.

- A <u>*path*</u> from *i* to *j* is a sequence of transitions that begins in I and ends in *j*, such that each transition has a positive *P*.
- A state *j* is <u>*reachable*</u> from state *i* if there is a path from *i* to *j*.
- Two states *i* and *j* are said to <u>communicate</u> if *j* is reachable from *i*, and *i* is reachable from *j*.
- A set of states S in M.C. in a <u>*closed set*</u> if no state outside of S is reachable from any state in S.
- A state *i* is an <u>*absorbing state*</u> if $P_{ii} = 1$.

Classification of States in M.C. (continued)

- A state *i* is a *transient state* if there exists a state *j* that is reachable from *i*, but the state *i* is not reachable from *j*.
- If a state is not transient, it is called a *recurrent state*.
- A state *i* is *periodic* with period k >1 if k is the *smallest* number such that all paths leading from state *i* back to state *i* have a length that is multiple of k. If a recurrent state is not periodic, it is referred to as *aperiodic*.
- If all states in a chain are recurrent, aperiodic, and communicate with each other, the chain is *ergodic*.

Steady-State Probabilities

- Study the long-run behavior of a M.C.
- $\lim_{n\to\infty} P^n = \begin{bmatrix} \pi_1 & \pi_2 & ... & \pi_s \\ \pi_1 & \pi_2 & ... & \pi_s \\ .. & .. & .. & .. \\ \pi_1 & \pi_2 & & \pi_s \end{bmatrix}$ Or $\lim_{n\to\infty} P_{ij}(n) = \pi_j$
- π [π_1 , π_2 , …, π_s] is called steady-state distribution or equilibrium distribution.
- Indications
  - Steady state doesn't depend on what is the initial state.
  - P^n has identical rows (after a long time, M.C. settles down)

Use of Steady-State Probabilities in Decision Making

In the cola example, suppose that each customer makes one purchase of cola during any week (52 weeks a year). Suppose there are 100 million cola customers. One selling unit of cola costs the company $1 to produce and is sold for $2. For $500 million per year, an advertising firm guarantees to decrease from 10% to 5% the fraction of cola 1 customers who switch to cola 2 after a purchase. Should the company that makes cola 1 hire the advertising firm? (Ex5, p.978).

Mean First Passage Time (MFPT)

- M_{ij} - expected number of transitions before we first reach state j from state i.
- For an ergodic chain $m_{ij} = 1 + \sum_{k \neq j} p_{ik} m_{kj}$
- $m_{ii} = \frac{1}{\pi_i}$

Absorbing Chains

- A M.C. is considered as an absorbing chain if some of the states are absorbing and the rest are transient states.
- If present in t_i (transient state), the expected # of periods will be spent in transient state t_j before absorption is the *ij*th element of $(I-Q)^{-1}$.
- If present in t_i (transient state), the probability eventually be absorbed in a_j is the *ij*th element of $(I-Q)^{-1}R$.

Work Force Planning (p.985)

The law firm of Mason and Burger employs three types of lawyers: junior lawyers, senior lawyers, and partners. During a give year, there is a 0.15 prabability that a junior lawyer will be promoted to senior lawyer and a 0.05 probability that he or she will leave the firm. Also, there is a 0.20 probability that a senior lawyer will be promoted to partner and a 0.10 proability that he or she will leave the firm. There is a 0.05 proability that a partner will leave the firm. The firm never demotes a lawyer. (1) How long does a newly hired junior lawyer stay with the firm? (2) What is the probability that a newly hired junior lawyer will leave the firm as a partner? (3) What is the average length of time that a partner spends with the firm (as the partner)?

Queuing Theory

- Study the process of waiting lines (or queues)
- Answer the questions such as
  - What is the expected # of customers in queue?
  - What is the expected time that a customer spends in queue?
  - What is the probability distribution of the expected # of customers in queue?
  - What is the probability distribution of a customer's waiting time?
  - To ensure only 1% of all customers will have to wait more than 5 minutes, how many tellers should be employed?

Queuing Terminology

- Input and Output Processes
- Assumptions on Input (Arrival) Process
  - No more than one arrival can occur at a given time. (If # of arrival > 1 at an instant, it is called "bulk arrival".
  - Arrival process is not affected by the number of customers waiting. (Exceptions: When arrivals are drawn from a small population, called "finite source models". If a customer arrives but fails to enter the system, we say the customer has "balked").
- Assumptions on Output (Service) Process
  - Service time is independent of the # of customers in waiting.
  - Two types: Servers in parallel & Servers in series.

Queuing Terminology (continued)

- Queue Discipline (order in which customers are served)
  - FCFS (First come, first served)
  - LCFS (Last come, first served)
  - SIRO (Service in random order)
  - Priority Queuing (Classify arrivals into categories with priorities, within one priority level use FCFS).
- Method Used by Arrivals to Join Queue
  - Single line
  - Multiple lines
    - Shortest line?
    - Allowed to switch?

Modeling Arrival Process

- Arrival time t_i and *i*th interarrival time T_i.
- T_i's are assumed to be independent, continuous random variables described by the random variable A.
- Each T_i is governed by the same A implies that the distribution of arrivals is independent of the time of the day.
- Exponential distribution of an arrival process.
  - λ is the arrival rate (# of arrivals/hr)
  - E (A) = $1/\lambda$, var A = $1/\lambda^2$.

No-Memory Property of the Exponential Distribution

- No-Memory Property: $P\ (A > t + h \mid A \geq t\ ) = P\ (A > h\ )$
  - *P* of no arrival during the next *h* hours provided no arrival for the last *t* hours = *P* of no arrival for the last *h* hours.
- Implications of No-Memory Property:
  - Probability doesn't depend on the value of t.
  - If we want to know the probability distribution of the time until the next arrival, it doesn't matter how long it has been since the last arrival.
  - To predict future arrival patterns, we need not keep track of how long it has been since the last arrival.

Poisson Distribution and Exponential Distribution

- Interarrival times are exponential with parameter λ if and only if the number of arrivals to occur in an interval of length *t* follows a Poisson distribution with parameter $\lambda\, t$.
- $E\ (N_t) = \text{var}\ N_t = \lambda\, t$

Modeling the Service Process

- Assume service times of different customers are independent random variables governed by S that has a density function of s(t).
- s(t) follows an exponential function.
- Is the service rate (# of services/per hour).

Modeling the Service Process

- Assume service times of different customers are independent random variables governed by S that has a density function of s(t).
- If s(t) follows an exponential function,
  - μ is the service rate (# of services/per hour).
  - Mean service time is $1/\mu$.
- Assumption of exponential service can simplify computations, e.g., for a multi-server system.

Erlang Distribution

- Density function is specified by two parameters:
  - Rate parameter R
  - Shape parameter k (k>0)

$$f(t) = \frac{R(Rt)^{k-1} e^{-Rt}}{(k-1)!}$$

- E (T) = k/R; var (T) = k/R^2
- For k=1, Erlang density looks similar to exponential distribution.
- When R = kλ, the interarrival process is equivalent to a customer going through *k* no-memory phases before arriving.

Kendall-Lee Notion for Queuing Systems

- Described by six charateristics: 1/2/3/4/5/6
- 1 is for the nature of arrival process
  - M = interarrival times are independent, identically distributed (iid), random with exponential distribution
  - D = iid and deterministic
  - E_k = iid Erlangs with shape parameter k
  - GI = iid governed by general distribution
- 2 is for the nature of service process
  - M, D, E_k, G definitions are identical to that of the arrival process

Kendall-Lee Notion for Queuing Systems (continued)

- 3 is the number of parallel servers
- 4 describes the queue discipline
  - FCFS = first come, first served
  - LCFS = last come, last served
  - SIRO = service in random order
  - GD = general queue discipline
- 5 is for the maximum allowable number of customers in the system (waiting and in service)
- 6 is the size of the population from which customers are drawn.

Birth-Death Processes

- Terminology
  - State: # of people present at time t.
  - $P_{ij}(t)$: probability that j people will be present at time t, given that at time 0, i people are present.
  - Steady state: $P_{ij}(t)$ reaches for large t.
- Laws of Motion
  - With probability $\lambda_j \Delta t+o(\Delta t)$, a birth (arrival) occurs between [t, t+ Δt]. Increases state from j to j+1. λ_j called birth rate.
  - With probability $\mu_j \Delta t+o(\Delta t)$, a death (service) occurs between [t, t+ Δt]. Decreases state from j to j-1. μ_j is called Death rate.
  - Births and deaths are independent.

Relation of Exponential Distribution to B-D Processes

- Most queuing systems with exponential interarrival times and service times may be modeled as birth-death processes.
- If either interarrival times and service times are nonexponential, the birth-death process model is not appropriate.