Topic: 1-C: Introduction to Linear Programming, Applications in Transportation Models

## (1) Mathematical Program

“Programming” has been used traditionally by planners to describe the process of *operations planning and resource allocation*.

xi: decision variables🡪 could be x(i,j) for link (i,j), x(k,i,j) for vehicle k selecting link (I,j), x(k,I,j,t,t’) vehicle k selecting link (I,j) departing from time t at node I and arrive at time t’ at node j,

binary: positive -> nonpositive >=0, range, lower bound and upper bound

Minimize / maximize f(x1, x2, …, xn) (objective function)

Minimize cost: transportation cost (travel time, emissions, variability), construction cost, crew operating cost

maximize profit: user fees collected, maximum utility (user benefits)

≥

Subject to: gi(x1, x2, …, xn) = bi (structural/technological constraints)

≤

xi ≥ 0 i=1,2, …, n (nonnegativity constraints)

## Linear Programming Standard Form

m= numbers of different types of resources

n = number of different activities

xj = level of activity j

cj = increase in objective function that would result from each unit increase in xj

bi = amount of resource i

aij= amount of resource i consumed by each unit of activity j (technological coefficient)

column vector: **decision vector** x = (x1, x2, …, xn)T, **right-hand-side vector** b = (b1, b2, …, bm)T

row vector: **cost vector** c = (c1, c2, …, cn)

constraint matrix: A = 

min Z = cx

1×n×n×1

s.t. A x = b

m×n×n×1 = m×1

x ≥ 0

## (2) Linear Programming Axioms

1. Proportionality : consider activity j, contribution to obj: cjxj

usage of resource i: aijxj

both are proportional to the level of activity j

1. Additivity: no “cross-term” e.g. ½ x1 x2 in the objective or constraints

Travel time = mean travel time + standard deviation of travel time

Mean travel time (MTT)= (T1+T2+T3)/3

Variance = (T1-MTT)2+(T2-MTT)2+(T2-MTT)2

Standard deviation = sqrt ((T1-MTT)2+(T2-MTT)2+(T2-MTT)2)

1. Divisibility: fractional values for decision variables are permitted

Integer solutions:

1. Certainty: A, b and c are known with certainty

Travel time, demand, capacity,

Remarks:

By assumptions (1) and (2), the linearity in the objection function and constraints is guaranteed.

Successive linear approximation can be used to solve nonlinear programming problems.

e.g. minimize travel time variances

Integer programming is used to deal with discrete decision variables.

X = 0 or 1

Stochastic programming is used to deal with programs with uncertain A, b and c.

T/F questions:

1. A linear programming model can consider the setup cost for each bus route.

Setup cost if a fixed cost, needs to use a binary variable

CX+fC\*Y

Y is a binary

1. If only one variable is restricted to be an integer value, then the problem is still a LP model.

mp

1. All the coefficients in the cost function and constraints should be known with certainty in a liner programming.
2. We cannot consider constraint x1/(x1+x2+x3) >=0.5 in a LP model.
3. In the standard form of LP, each variable should be positive.

* Nonnegative

1. In LP, the number of constraints should be less than the number of variables.

You can have as many constraints as you want

**Conversion to Standard Form**

(Transformation to standard form will be useful when we begin talking about the simplex method)

Example:

Max z = 3x1- 5 x2 + x3

s.t. x1 + x2 + x3≤ 7

- x1 – 2 x2 + x3 ≤ - 4

x2 ≤ 0, x1 and x3 unrestricted in sign

1. maximization -> minimization
2. b ≥ 0
3. inequality constraints -> equality constraints (slack and surplus variables)
4. x ≥0

**Definitions**

Feasible Region {x: Ax=b, x ≥0}

Feasible Solution  satisfying 

Optimal Solution x\* satisfying (i) x\* is a feasible solution (ii) cx\*≤ cx 

## (3) Examples of LP Formulations

**Transportation Problem**

Plant i has a supply of Si widgets

Market j has a demand of Dj widgets

cij = unit cost of shipping from i to j

Matrix model Network flow model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Cost C(I,j) | Market 1 | Market 2 | Market 3 | Supply |
| Plant 1 | 2 | 3 | M | 5 |
| Plant 2 | 5 | 2 | M | 4 |
| Plant 3 | M | 3 | 3 | 3 |
| Demand | 7 | 3 | 5 |  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Decision X(i,j) | Market 1 | Market 2 | Market 3 | Supply |
| Plant 1 |  |  |  | 5 |
| Plant 2 |  |  |  | 4 |
| D Plant 3 |  |  |  | 3 |
| Demand | 7 | 3 | 5 |  |

Decision variable: xij = amount to ship from plant i to market j

Objective function:

Constraints:

Remarks:

* The transportation problem is a special case of a more general class of network flow problems.
* (**Totally unimodular matrix**) All the coefficients are 1 and every variable appears in exactly two constraints.
* If the supplies and demands are integer, every basic solution and the optimal solution have integer values.

**Shortest path problem**



Basic program



**Extension 1:**

Let tij = bus fare on link (i,j) (unit: dollar)

Please find a shortest path that costs less than T dollars



**Extension 2:**

Please find the shortest paths from origin 1 to all the other nodes 2, .., 6.



**Work Scheduling Problem**

A transit authority requires different numbers of full-time employees on different days of the week. The number of full-time employees required on each day is given by the following table:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Day 1 = Monday | Day 2 = Tuesday | Day 3 = Wednesday | Day 4 = Thursday | Day 5 = Friday | Day 6 = Saturday | Day 7 = Sunday |
| 11 | 10 | 13 | 13 | 16 | 10 | 10 |

Union rules state that each full-time employee must work five consecutive days and then receive two days off. For example, an employee who works Monday to Friday must be off on Saturday and Sunday. The authority wants to meet its daily requirements using full-time employees and no more than 10 part-time employees. The problem is to find the minimal number of full-time employees that must be hired.

Approach 1:

xi= number of employees that start their work on day i

yi= number of employees that start their work on day i

min z = x1+ x2 + x3+ x4+x5 +x6+x7

(day1) x1+ x4+x5 +x6+x7 + y1>=11

(day2) x1+ x2+ x5 +x6+x7 + y2>=10

(day3) x1+ x2 + x3 + +x6+x7 + y3 >=13

(day4).

(day7)

y1+y2+ ….+ y7 <= 10

x1, …, x7 >= 0

**Example: OD Demand Estimation**

Consider a simple metro network with 6 nodes, and each line represents a service line.

passenger counts are collected on links 1->5, 2->5 and 6->4 through “point-check”.

There are four major OD pairs 1->3, 1->4, 2->3 and 2->4 served in this network.

Please estimate OD demand along the four OD pairs. A unit of volume = 1000 passengers.



Decision variables: OD flows x13, x14, x23, x24

Linear system

x13+ x14 = 3

x23+ x24 = 2

x13+ x23 = 3

x13, x14, x23, x24 >=0

Questions: 1) Please find at least one feasible solution.

2) Do we have a unique solution?

3) Could we have more information if we also know the volumes on links 5->6 and 6->4?

4) Through a previous survey, the metro system has historical OD demand information

1-> 3: 1

1-> 4: 1.5

2-> 3: 0.7

2-> 4: 1

Please utilize the above information to estimate a most likely OD demand volume.

Approach 1: Minimizing least squared errors

Min (x13-1)2 + (x14-1.5)2+ (x23-0.7)2 + (x24-1)2

Approach 2: Linear programming

Hints: Define additional constraint

x13+x13+ - x13- = 1

Where x13+ , x13- >= 0

x13+ , x13  can be viewed as positive and negative deviations, repectively.

Min x13+ + x13- …..

Quiz:

**Knapsack Problem**

Jason is going for an overnight hike. There are four items Jansen is considering taking along the trip. The weight of each item and the benefit Jansen feels he would obtain from each item:

|  |  |  |
| --- | --- | --- |
|  | Weight | Benefit |
| Flashlight | 12 | 25 |
| Energy snack | 5 | 20 |
| Map | 3 | 8 |
| Camera | 7 | 12 |

* Jansen’s knapsack can hold up to a total weight of 14
* Maximize benefit

**Knapsack problem**

We are given a knapsack of capacity *c* and a set of *n* objects numbered *1,2,…,n*. Each object *i* has weight *wi* and profit *pi*.

Let *x =* [*x1, x2,…, xn*] be a solution vector in which *xi = 0* if object *i* is not in the knapsack, and *xi = 1* if it is in the knapsack.

The goal is to find a subset of objects to put into the knapsack so that

We maximize value of chosen objects subject to weight not exceeding capacity (that is, the objects fit into the knapsack)



Generalization:

• Reward is product of demand, reward scale factor

• Time / labor / physical materials / network capacity

## (4) Linear Programming Geometry

**Graphical Solution Method**

(can be used to solve LP’s with 2 decision variables)

Example: max z = 6 x1 + 4 x2

s.t. x1+x2 ≤ 6 (1)

2 x1+x2 ≤ 9 (2)

2 x1+3x2 ≤ 16 (3)

x1 ≥ 0 (4)

x2 ≥ 0 (5)



|  |  |  |  |
| --- | --- | --- | --- |
| **Point** | **Tight constraints** | **Extreme point** | **Objective** |
| A | (1) and (3) | (2,4) | 28 |
| B | (1) and (2) | (3,3) | 30 |
| D | (4) and (5) | (0,0) | 0 |
| E | (2) and (5) | (4.5,0) | 27 |
| F | (3) and (4) | (0,16/3) | 21.3 |

**Solution procedure**

Step 1 Plot the feasible region

Step 2 Draw an isoprofit line z=cx

Step 3 Move the isoprofit line in the direction of vector c.

**Definition**

A set of points S is a **convex set** if the line segment joining any pair of points in S is wholly contained in S.

For any convex set, a point P in S is an **extreme point** if each line segment that lies completely in S and contains the point P has P as an endpoint of the line segment.

**Remarks**

* Linear system Ax=b is a convex set.
* The feasible region of any linear programming problem is a convex set.
* If a linear programming in standard form has a finite optimal solution, then it has an optimal corner (or extreme) point solution.
* In general, in a problem with n decision variables and m structural constraints, we have  sets of equations to solve.
* Possible solution method: enumerating possible extreme point and eliminate infeasible points, finally select the feasible solution with the best objective values.

**Possible outcomes of an LP**

1. Infeasible: feasible region is empty
2. Unbounded e.g. max 15x1 s.t. x1≥0.
3. Multiple optimum e.g. max 3x1 + 3x2 s.t. x1 + x2≤1, x1 x2≥0.
4. Unique optimal solution

# of optimal solutions = 0, 1 and infinity

## (5) Linear Algebra Review

The following statements are equivalent

1. A is nonsingular

2. There is a unique x satisfying Ax=b

3. The columns of A are linearly independent

4. det (A) ≠ 0.

Consider a system Ax=b of m linear equations in n variables (assuming n≥m)

A **basic solution** of Ax = b is a solution that only uses linearly independent columns of A.

A basic solution to Ax=b is obtained by setting n-m variables equal to zero, and then solving for the remaining variables

Consider Ax=b s.t. x≥0

Suppose we pick an index set B with m elements and solve ABxB = b and it turns out that xB≥0. Then x = (xB ,xN) is called a **basic feasible solution** (i.e. nonnegative basic solution)

xB basic variables, xN nonbasic variables

AB basic matrix (or simply basis), AN nonbasic variables



**Optimality Condition**



If rank(B)=m then B is invertible xB=B-1(b─NxN)

z = cx = cBxB+ cNxN = cB(B-1(b ─ NxN))+ cNxN = cBB-1b+ (cN ─ cB B-1 N) xN

If we select xN to be zero, then xB= B-1b and z= cBB-1b

Define reduced cost 

**Optimality condition: (minimization problem) **

**(maximization problem) **

**Feasibility condition: xB= B-1b≥0**

Consider min z= -2x1-3x2-7x3

s.t. 2x1+x2+2x3=4

3x1-x2-2x3=1

x1, x2, x3 ≥0

Can x2 and x3 form a basis?

Suppose the current basic variables in an iteration are x1 and x2



Verify

xB= B-1b=(1,2)T,  z = cBxB=-8

Is the current xB optimum?

****

**Geometric and algebraic interpretation of the simplex method**

|  |  |  |
| --- | --- | --- |
| Method Sequence | Geometric Interpretation | Algebraic Interpretation |
| Initialization | Choose (0,0) to be the initial feasible corner point (constraint boundaries 4&5). | Choose x1 and x2 to be the nonbasic variables (=0) for the initial basic feasible solution (BFS): (0,0,6, 9,16). |
| Optimality test | Not optimal, because moving along either edge from (0,0) increases Z. | No optimal, because increasing either nonbasic variable (x1 and x2) increases Z. |
| Iteration 1 |  |  |
| Step 1 | Move up the edge lying on the x1 axis. | Increase x1 while adjusting other variable values to satisfy the system of equations. |
| Step 2 | Stop when the first new constraint boundary (2) is reached. | Stop when the first basic variable (x3, x4, or x5) drops to zero (x4). |
| Step 3 | Find the intersection of the new pair of constraint boundaries (2&5): (4.5,0) is the new feasible corner point. | With x1 now a basic variable and x4 now a nonbasic variable, solve the system of equations: (4.5,0,1.5,0,7) is the new BFS. |
| Optimality test | Not optimal, because moving along the edge from (4.5,0) to the right increases Z. | Not optimal, because increasing one nonbasic variable (x2) increases Z. |
| Iteration 2 |  |  |
| Step 1 | Move along the edge to the right | Increase x2 while adjusting other variable values to satisfy the system of equations. |
| Step 2 | Stop when the first new constraint boundary (1) is reached. | Stop when the first basic variable (x1,x3 or x5) drops to zero (x3). |
| Step 3 | Find the intersection of the new pair of constraint boundaries (1&2): (3,3) is the new feasible corner point. | With x2 now a basic variable and x3 now a nonbasic variable, solve the system of equations: (3,3,0,0,1) is the new BFS. |
| Optimality test | (3,3) is optimal, because moving along either edge from (3,3) decreases Z. | (3,3,0,0,1) is optimal, because increasing either nonbasic variable (x3 or x4) decreases Z. |