OPERATIONAL RESEARCH II

Agustina Eunike, ST., MT., MBA.
Industrial Engineering – University of Brawijaya

MINIMUM-COST FLOW PROBLEM

MCFP APPLICATION

Kind of Application	Supply Nodes	Transshipment Nodes	Demand Nodes
Operation of a distribution network	Sources of goods	Intermediate storage facilities	Customers
Solid waste management	Sources of solid waste	Processing facilities	Landfill locations
Operation of a supply network	Vendors	Intermediate warehouses	Processing facilities
Coordinating product mixes at plants	Plants	Production of a specific product	Market for a specific product
Cash flow management	Sources of cash at a specific time	Short-term investment options	Needs for cash at a specific time

Minimum Cost Flow Problem

$$\min \sum_{\text{all arcs}} c_{ij}x_{ij}$$

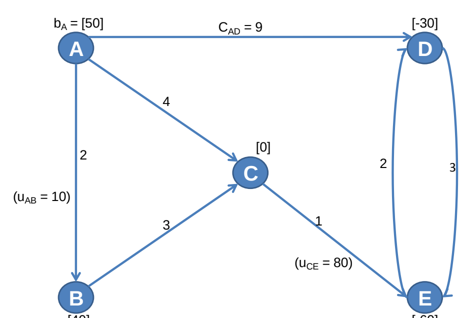
$$\text{s.t.} \quad \sum_j x_{ij} - \sum_k x_{ki} = b_i \qquad \text{(for each node } i \text{ in the network)}$$

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

Minimum Cost Flow Problem

$b_i > 0$ if node i is a supply node,
$b_i < 0$ if node i is a demand node,
$b_i = 0$ if node i is a transshipment node.

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 there is no lower bound, let $L_{ij} = 0$)
U_{ij} = upper bound on flow through arc (i, j) (if there is no upper bound, let $U_{ij} = \infty$)



Linear Programming

Minimize $Z = 2X_{AB} + 4X_{AC} + 9X_{AD} + 3X_{BC} + X_{CE} + 3X_{DE} + 2X_{ED}$;

subject to

$X_{AB} + X_{AC} + X_{AD} = 50$

$-X_{AB} + X_{BC} = 40$

$-X_{AC} - X_{BC} + X_{CE} = 0$

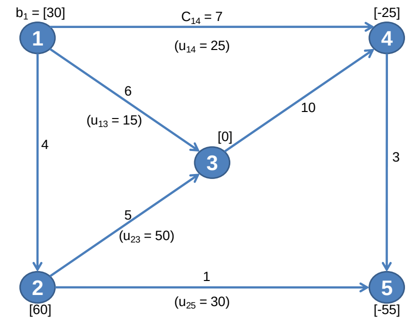
$-X_{AD} + X_{DE} - X_{ED} = -30$

$-X_{CE} - X_{DE} + X_{ED} = -60$

$X_{AB} \le 10$

$X_{CE} \le 80$

$X_{ij} \ge 0$; untuk semua i dan j



Some Assumptions

1. All data is integral. (Needed for some proofs, and some running time analysis).
2. The network is directed and connected
3. $\sum_{i=1 \text{ to } n} b(i) = 0$.
   (Otherwise, there cannot be a feasible solution.)

Minimum Cost Flow Problem

- Special Cases of MCFP:
  - Transportation
  - Transhipment
  - Assignment
  - Shortest Path
  - Maximum Flow
  - CPM
  - ...

Minimum Cost Flow Problem

- Transportation:

$$\text{Minimize} \quad Z = \sum_{i=1}^{m} \sum_{j=1}^{n} c_{ij} x_{ij},$$

subject to

$$\sum_{j=1}^{n} x_{ij} = s_i \quad \text{for } i = 1, 2, \ldots, m,$$

$$\sum_{i=1}^{m} x_{ij} = d_j \quad \text{for } j = 1, 2, \ldots, n,$$

and

$$x_{ij} \ge 0, \quad \text{for all } i \text{ and } j.$$

- Assignment:

$$\text{Minimize} \quad Z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij},$$

subject to

$$\sum_{j=1}^{n} x_{ij} = 1 \quad \text{for } i = 1, 2, \ldots, n,$$

$$\sum_{i=1}^{n} x_{ij} = 1 \quad \text{for } j = 1, 2, \ldots, n,$$

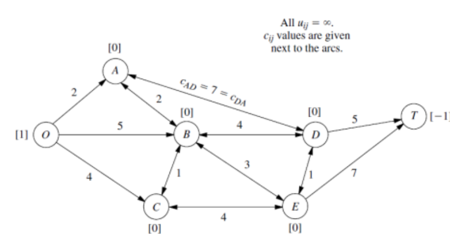
and

$$x_{ij} \ge 0, \quad \text{for all } i \text{ and } j$$
$$(x_{ij} \text{ binary}, \quad \text{for all } i \text{ and } j).$$

$$x_{ij} = \begin{cases} 1 & \text{if assignee } i \text{ performs task } j, \\ 0 & \text{if not.} \end{cases}$$

Minimum Cost Flow Problem

- Shortest Path



Minimum Cost Flow Problem

- Shortest Path

x_{ij} = amount of flow in arc (i, j)

$= \begin{cases} 1, \text{ if arc } (i, j) \text{ is on the shortest route} \\ 0, \text{ otherwise} \end{cases}$

c_{ij} = length of arc (i, j)

Thus, the objective function of the linear program becomes

$$\text{Minimize } z = \sum_{\substack{\text{all defined} \\ \text{arcs } (i, j)}} c_{ij} x_{ij}$$

The constraints represent the *conservation-of-flow equation* at each node:

Total input flow = Total output flow

Mathematically, this translates for node j to

$$\begin{pmatrix}\text{External input} \\ \text{into node } j\end{pmatrix} + \sum_{\substack{i \\ \text{all defined} \\ \text{arcs } (i, j)}} x_{ij} = \begin{pmatrix}\text{External output} \\ \text{from node } j\end{pmatrix} + \sum_{\substack{k \\ \text{all defined} \\ \text{arcs } (j, k)}} x_{jk}$$

Minimum Cost Flow Problem

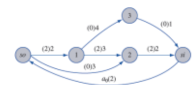
- Maximum Flow

For a flow to be feasible, it must have two characteristics:

$$0 \leq \text{flow through each arc} \leq \text{arc capacity}$$

and

$$\text{Flow into node } i = \text{flow out of node } i$$



$$\max z = x_0$$

s.t.

$x_{so,1} \leq 2$	(Arc capacity constraints)
$x_{so,2} \leq 3$	
$x_{12} \leq 3$	
$x_{2,si} \leq 2$	
$x_{13} \leq 4$	
$x_{3,si} \leq 1$	
$x_0 = x_{so,1} + x_{so,2}$	(Node *so* flow constraint)
$x_{so,1} = x_{12} + x_{13}$	(Node 1 flow constraint)
$x_{so,2} + x_{12} = x_{2,si}$	(Node 2 flow constraint)
$x_{13} = x_{3,si}$	(Node 3 flow constraint)
$x_{3,si} + x_{2,si} = x_0$	(Node *si* flow constraint)
$x_{ij} \geq 0$	

Minimum Cost Flow Problem

- Maximum Flow

min $z = x_0$

$x_{so,1}$	$x_{so,2}$	x_{13}	x_{12}	$x_{3,si}$	$x_{2,si}$	x_0		rhs	Constraint
1	1	0	0	0	0	−1	=	0	Node *so*
−1	0	1	1	0	0	0	=	0	Node 1
0	−1	0	−1	0	1	0	=	0	Node 2
0	0	−1	0	1	0	0	=	0	Node 3
0	0	0	0	−1	−1	1	=	0	Node *si*
1	0	0	0	0	0	0	≤	2	Arc (*so*, 1)
0	1	0	0	0	0	0	≤	3	Arc (*so*, 2)
0	0	1	0	0	0	0	≤	4	Arc (1, 3)
0	0	0	1	0	0	0	≤	3	Arc (1, 2)
0	0	0	0	1	0	0	≤	1	Arc (3, *si*)
0	0	0	0	0	1	0	≤	2	Arc (2, *si*)

All variables nonnegative

Each variable x_{ij} has a coefficient of +1 in the node i flow balance equation, a coefficient of −1 in the node j flow balance equation, and a coefficient of 0 in all other flow balance equations.

Practices...

- United Airlines has seven daily flights from BOS to SFO, every two hours, starting at 7am.
- Capacities are **100**, **100**, **100**, **150**, **150**, **150**, and ∞.
- Passengers suffering from overbooking are diverted to later flights.
- Delayed passengers get $**200** plus $**20** for every hour of delay.
- Suppose that today the first six flighs have **110**, **160**, **103**, **149**, **175**, and **140** confirmed reservations.

Determine the most economical passenger routing strategy!

THE NETWORK SIMPLEX METHOD

NETWORK SIMPLEX METHOD

- Comparison Running Time

Algorithm	Running Time (sec)	# Iterations
Standard Simplex	334.59	42759
Network Simplex	7.37	23306
Ratio	2.2 %	54 %

Average over **5** random instances with **10, 000** nodes and **25, 000** arcs each.

Summary

1. Network simplex is extremely fast in practice.

2. Relying on network data structures, rather than matrix algebra, causes the speedups. It leads to simple rules for selecting the entering and exiting variables.

3. A good pivot rule can dramatically reduce running time in practice.

References

- Frederick Hillier and Gerald J.Lieberman, Introduction to Operations Research, Holden Day Ltd, San Fransisco, 1997
- Taha, Hamdy, Operation Research : An Introduction, Macmillan Publishing Company., New York, 1997