OPERATIONAL RESEARCH II

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

Introduction

- Operational Research ⇛ research on operations
  - Operations / activities within an organization
  - Research means the scientific method is used to investigate the problem of concern
- Operational Research (Management Science) is characterized by the use of *mathematical models* in providing guidelines to managers for making effective decisions within the state of the current information, or in seeking further information if current knowledge is insufficient to reach a proper decision.

Introduction

- Operational Research ⇛ more than just mathematics
- Phases of an OR Study:
  1. Recognize the problem
  2. Formulate the problem
  3. Construct a model
  4. Find a solution
  5. Testing the solution
  6. Implement the solution

Introduction

- Some techniques are used in OR:
  - Linear programming, for models with linear objective and constraint functions
  - Integer programming, in which the variables assume integer values
  - Dynamic programming, in which the original model can be decomposed into more manageable subproblem
  - Network programming, in which the problem can be modeled as a network
  - Nonlinear programming, in which function of the model are nonlinear

Introduction

Analytical and Simulation Models Classification

	Strategy evaluation	*Strategy generation*
Certainty	Deterministic simulation Econometric models Systems of simultaneous equations Input-output models	Linear programming Network models Integer and mixed-integer programming Nonlinear programming Control theory
Uncertainty	Monte Carlo simulation Econometric models Stochastic processes Queueing theory Reliability theory	Decision theory Dynamic programming Inventory theory Stochastic programming Stochastic control theory

Statistics and subjective assessment are used in all models to determine values for parameters of the models and limits on the alternatives.

Note:

- The classification above is not rigid, since strategy evaluation models are used for improving decisions by trying different alternatives until one determined that appears "best".
- For strategy evaluation models, the user must first choose and construct the alternative and then evaluate it with the aid of the model
- For strategy generation models, the alternative is not completely determined by the user, rather, the class of alternatives is determined by establishing contraints on the decisions, and then an algorithmic procedure is used to automatically generate the "best" alternative within the class

Introduction

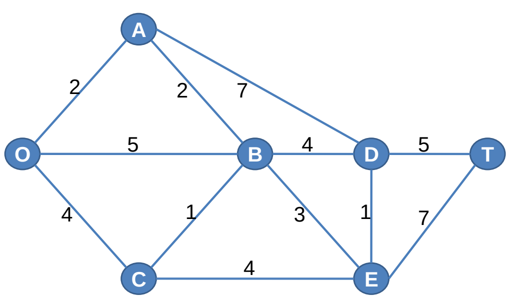
- Solutions of most OR techniques are not generally obtained in (formulalike) closed forms.
- Instead, they are determined by **algorithms**.
  - Algorithm: fixed computational rules that are applied repetitively to the problem, with each repetition (iteration) moving the solution closer to the optimum.
- Some mathematical models may be so complex that it is impossible to solve them by any of the available optimization algorithms. In such cases, it may be necessary to abondan the search for optimal solution and simply seek a *good* solution using **heuristics** approach.

Network Problem

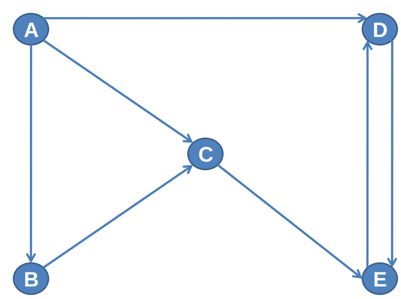
The Terminology of Networks

- Nodes
- Arcs
- Directed Arcs
- Undirected Arcs (Links)
- Directed Network
- Undirected Network
- Path
- Directed Path
- Undirected Path
- Cycle
- Connected Network
- Tree
- Spanning Tree
- Arc capacity
- Supply node
- Demand node
- Transhipment node

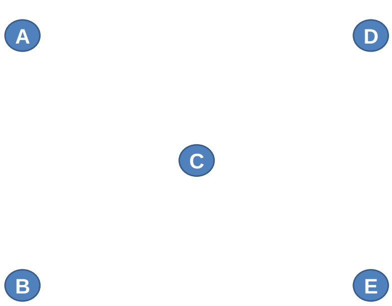
Example



Directed Networks

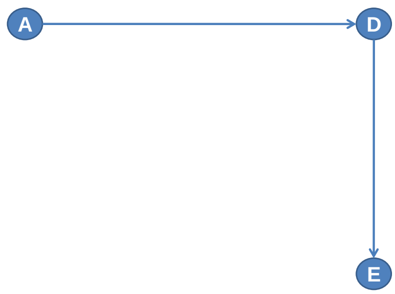


The nodes without arcs

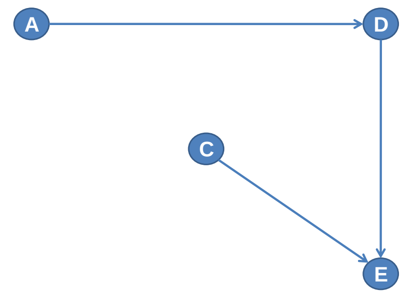


A tree with one arc

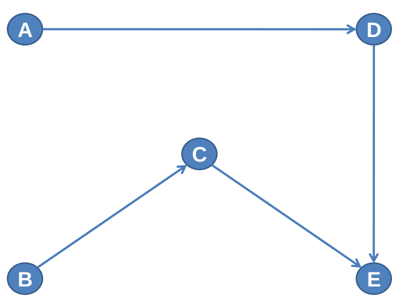
A tree with two arcs



A tree with three arcs



A spanning tree



Components of typical networks

Nodes	Arcs	Flow
Intersection	Roads	Vehicles
Airpots	Air lanes	Aircraft
Switching points	Wires, channels	Messages
Pumping stations	Pipes	Fluids
Work centers	Material-handling routes	Jobs

Network Models

- The shortest-route problem
- The minimal spanning tree problem
- The maximum flow problem
- Minimum cost flow problem
- Critical path (CPM) algorithm

Note:
Some network models are special condition of linear programming (transportation, assignment problem)

Network Models

- The shortest-route problem
  - The determination of the shortest route between two cities in an existing network of roads
- The minimal spanning tree problem
  - Design of telecomunication networks (fiber-optic networks, computer network, cable telivison network, etc)
- The maximum flow problem
  - Maximize the flow through a company's supply network from its vendors to its factories
- Minimum cost flow problem
  - The determination of the minimum cost-flow schedule for shipping goods from factory to storage facilities and then on to customers
- Critical path (CPM) algorithm
  - The determination of the time schedule (start and completion dates) for the activities of a construction project

THE SHORTEST-ROUTE PROBLEM

Shortest Route Problem

- The shortest-route problem is concerned with finding the shortest path in a network from one node (or set of nodes) to another node (or set of nodes).
- The criterion to be minimized in the shortest-route problem is not limited to distance even though the term "shortest" is used in describing the procedure. Other criteria include time and cost. (Neither time nor cost are necessarily linearly related to distance.)
- The Shortest-route problem can be formulated as a linear program.

Shortest-route Algorithm

If all arcs in the network have non-negative values then a labeling algorithm can be used to find the shortest paths from a particular node to all other nodes in the network.

Note: we use the notation [ ] to represent a permanent label and ( ) to represent a tentative label.

- Step 1: assign node 1 the permanent label [0,*S*]. The first number is the distance from node 1; the second number is the preceding node. Since node 1 has no preceding node, it is labeled *S* for the starting node.

Shortest-route Algorithm

- Step 2: compute tentative labels, (*d*,*n*), for the nodes that can be reached directly from node 1. *D* = the direct distance from node 1 to the node in question — this is called the distance value. *N* indicates the preceding node on the route from node 1 — this is called the preceding node value. (All nodes labeled in this step have *n* = 1.)
- Step 3: identify the tentatively labeled node with the smallest distance value. Suppose it is node *k*. Node *k* is now permanently labeled (using [,] brackets). If all nodes are permanently labeled, GO TO STEP 5.

Shortest-route Algorithm

- Step 4: consider all nodes without permanent labels that can be reached directly from the node *k* identified in step 3. For each, calculate the quantity *t*, where

T = (arc distance from node k to node i)
+ (Distance value at node k).

(Continued on next slide . . .)

Shortest-route Algorithm

– If the non-permanently labeled node has a tentative label, compare *t* with the current distance value at the tentatively labeled node in question.
  - If $t <$ distance value of the tentatively labeled node, replace the tentative label in question with (t,k).
  - If $t \geq$ distance value of the tentatively labeled node, keep the current tentative label.

– If the non-permanently labeled node does not have a tentative label, create a tentative label of (t,k) for the node in question.

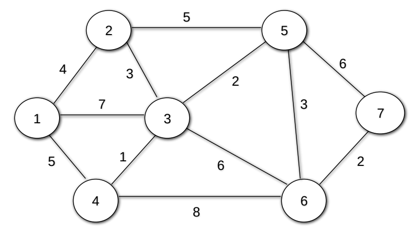
In either case, now GO TO STEP 3.

Shortest-route Algorithm

- Step 5:
  - The permanent labels identify the shortest distance from node 1 to each node as well as the preceding node on the shortest route.
  - The shortest route to a given node can be found by working backwards by starting at the given node and moving to its preceding node.
  - Continuing this procedure from the preceding node will provide the shortest route from node 1 to the node in question.

Example: Shortest Route

- Find the shortest route from node 1 to all other nodes in the network:

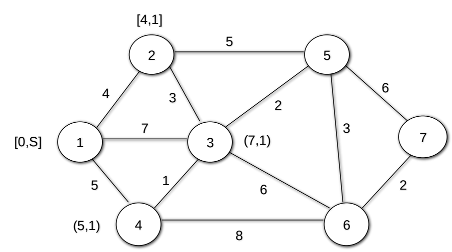


Example: Shortest Route

- Iteration 1
  - Step 1: assign node 1 the permanent label [0,S].
  - Step 2: since nodes 2, 3, and 4 are directly connected to node 1, assign the tentative labels of (4,1) to node 2; (7,1) to node 3; and (5,1) to node 4.
  - Step 3: node 2 is the tentatively labeled node with the smallest distance (4) , and hence becomes the new permanently labeled node.

Example: Shortest Route

- Tentative labels shown

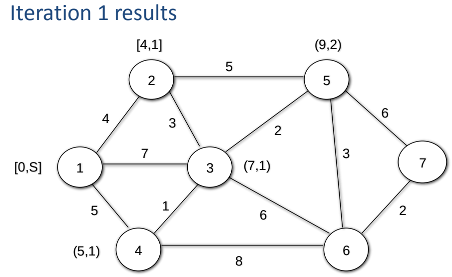


Example: Shortest Route

- Iteration 1
  - Step 4: For each node with a tentative label which is connected to node 2 by just one arc, compute the sum of its arc length plus the distance value of node 2 (which is 4).
    - Node 3: 3 + 4 = 7 (not smaller than current label; do not change.)
    - Node 5: 5 + 4 = 9 (assign tentative label to node 5 of (9,2) since node 5 had no label.)

Example: Shortest Route

- Iteration 1 results

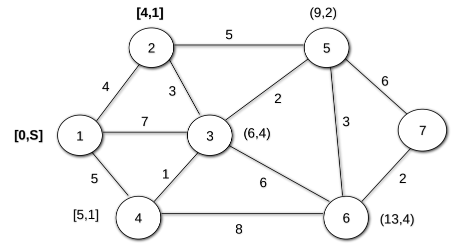


Example: Shortest Route

- Iteration 2
  - Step 3: node 4 has the smallest tentative label distance (5). It now becomes the new permanently labeled node.
  - Step 4: For each node with a tentative label which is connected to node 4 by just one arc, compute the sum of its arc length plus the distance value of node 4 (which is 5).
    - Node 3: 1 + 5 = 6 (replace the tentative label of node 3 by (6,4) since 6 < 7, the current distance.)
    - Node 6: 8 + 5 = 13 (assign tentative label to node 6 of (13,4) since node 6 had no label.)

Example: Shortest Route

- Iteration 2 results

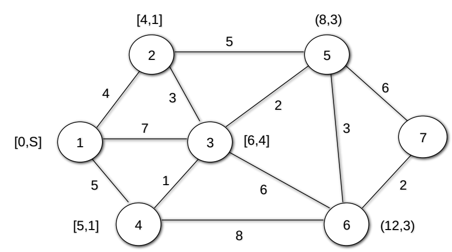


Example: Shortest Route

- Iteration 3
  - Step 3: node 3 has the smallest tentative distance label (6). It now becomes the new permanently labeled node.
  - Step 4: For each node with a tentative label which is connected to node 3 by just one arc, compute the sum of its arc length plus the distance to node 3 (which is 6).
    - Node 5: 2 + 6 = 8 (replace the tentative label of node 5 with (8,3) since 8 < 9, the current distance)
    - Node 6: 6 + 6 = 12 (replace the tentative label of node 6 with (12,3) since 12 < 13, the current distance)

Example: Shortest Route

- Iteration 3 results

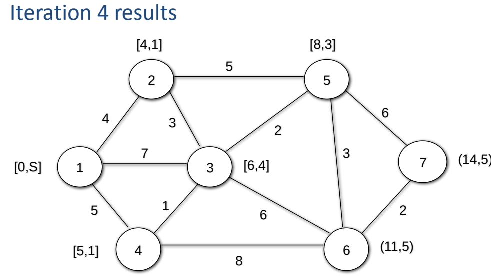


Example: Shortest Route

- Iteration 4
  - Step 3: node 5 has the smallest tentative label distance (8). It now becomes the new permanently labeled node.
  - Step 4: For each node with a tentative label which is connected to node 5 by just one arc, compute the sum of its arc length plus the distance value of node 5 (which is 8).
    - Node 6: 3 + 8 = 11 (replace the tentative label of node 6 with (11,5) since 11 < 12, the current distance.)
    - Node 7: 6 + 8 = 14 (assign tentative label to node 7 of (14,5) since node 7 had no label.)

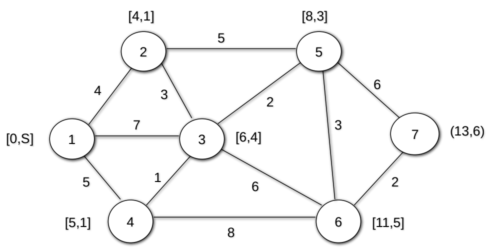
Example: Shortest Route

- Iteration 4 results



Example: Shortest Route

- Iteration 5 results



Example: Shortest Route

- Iteration 6
  - Step 3: node 7 becomes permanently labeled, and hence all nodes are now permanently labeled. Thus proceed to summarize in step 5.
  - Step 5: summarize by tracing the shortest routes backwards through the permanent labels.

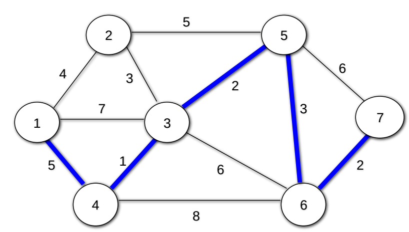
Example: Shortest Route

- Solution summary

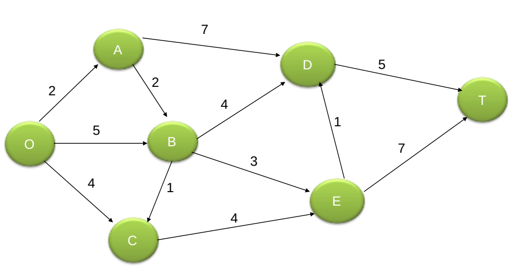
Node	minimum distance	shortest route
2	4	1-2
3	6	1-4-3
4	5	1-4
5	8	1-4-3-5
6	11	1-4-3-5-6
7	13	1-4-3-5-6-7

Example: Shortest Route

- Find the shortest route from node 1 to all other nodes in the network:

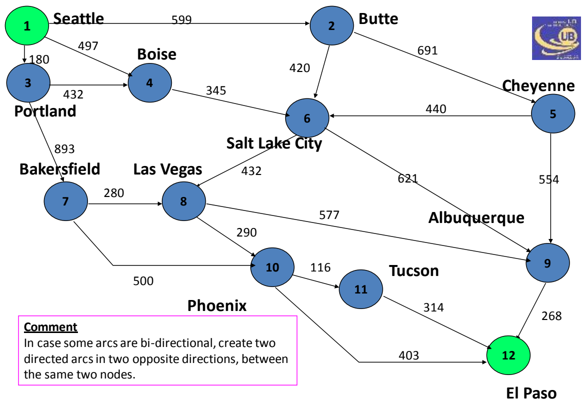


Example 2 : The road system at Servaada Park



FAIRWAY VAN LINES

Determine the shortest route from Seattle to El Paso over the following network of highways.



Comment
In case some arcs are bi-directional, create two directed arcs in two opposite directions, between the same two nodes.

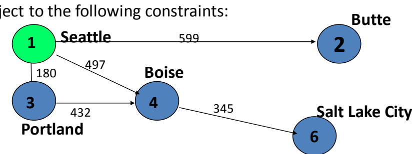
FAIRWAY VAN LINES – The Linear Programming Model

Decision variables

$$X_{ij} = \begin{cases} 1 & \text{if a truck travels on the highway from city } i \text{ to city } j \\ 0 & \text{otherwise} \end{cases}$$

Objective = Minimize $\Sigma\, d_{ij}X_{ij}$

Subject to the following constraints:



[The number of highways traveled out of Seattle (the start node)] = 1
$X_{12} + X_{13} + X_{14} = 1$

In a similar manner:
[The number of highways traveled into El Paso (terminal node)] = 1
$X_{9,12} + X_{10,12} + X_{11,12} = 1$

[The number of highways used to travel into a city] = [The number of highways traveled leaving the city].
For example, in Boise (node 4):
$X_{14} + X_{34} = X_{47}$.

Non-negativity constraints

FAIRWAY VAN LINES – spreadsheet

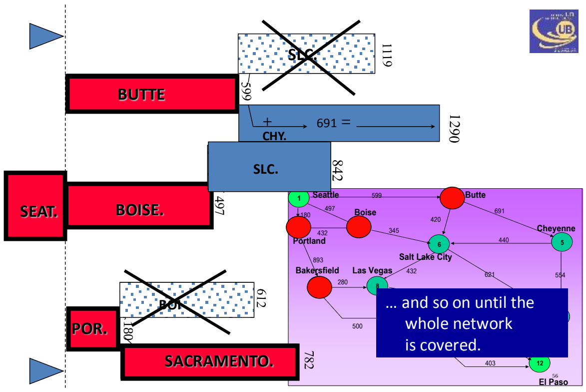
NODE INPUT		ARC INPUT			SOLUTION	TOTAL DISTANCE=	1731
NODE NAME	NODE #	FROM	TO	DISTANCE	FROM	TO	FLOW
Seattle	1	1	2	599	1	2	
Butte	2	1	3	180	1	3	
Portland	3	1	4	497	1	4	1
Boise	4	2	5	691	2	5	
Cheyenne	5	2	6	420	2	6	
Salt Lake City	6	3	4	432	3	4	
Bakersfield	7	3	7	893	3	7	
Las Vegas	8	4	6	345	4	6	1
Albuquerque	9	5	6	440	5	6	
Phoenix	10	5	9	554	5	9	
Tucson	11	6	8	432	6	8	
El Paso	12	6	9	621	6	9	1
		7	8	280	7	8	
		7	10	500	7	10	
		8	9	577	8	9	
		8	10	290	8	10	
		9	12	268	9	12	1
		10	11	116	10	11	
		10	12	403	10	12	
		11	12	314	11	12	

FAIRWAY VAN LINES – The Network Model

The Dijkstra's algorithm:

- Find the shortest distance from the "START" to each other node, in the order of the closet nodes to the "START".
- Once the shortest route to the m closest node is determined, the (m+1) closest can be easily determined.
- This algorithm finds the shortest route from the start to all the nodes in the network.

An illustration of the Dijkstra's algorithm



Dijkstra's algorithm - continued

- When all the network is covered, the shortest route from START to every other node can be identified.
- Trace the path that leads to each node by backtracking from each node toward node START.