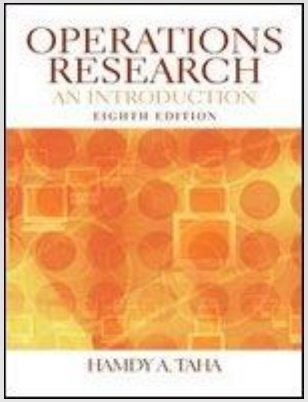
Chapter 6:
Network Models



Hamdy A. Taha, Operations Research: An introduction, 8th Edition


AT&T 3G
9:42 AM
Tuesday
9
SMS
Text
Calendar
Photos
Camera
280
73°
YouTube
Weather
Clock
Calculator
Notes
Settings
iTunes
App Store
Phone
Mail
Safari
iPod

Mute ur call

Network Models

- There is a many of operation research situation is modeled and solved as network (nodes can connected by branches)

- There are five network models algorithms

  1- Minimal spanning tree

  2- shortest-route algorithms

  3- maximum-flow algorithms

  4- minimum cost capacitated network algorithms

  5- Critical path(CPM) algorithms

Network Models (CONT.)

1- Design of an offshore gas pipeline network connecting wellheads in gulf of Mexico to an inshore delivery points.; the objective of the model is minimize the cost constructing the pipeline.

- The situation represented as ***Minimal spanning tree***.

2- Determination of the shortest route between two cities in a network of roads.

- This situation is ***shortest-route algorithms***

Network Models (CONT.)

3- determination the maximum capacity (in ton per year) of a coal slurry pipeline network

- This situation is ***maximum flow algorithms***

4- determination of the minimum-cost flow schedule from oil field to refineries through a pipeline network.

- This situation ***is minimum-cost capacitated network algorithms***

Network Models (CONT.)

5- determination the time schdule (start and completion date) for activities

- This situation is ***(CPM) algorithms***

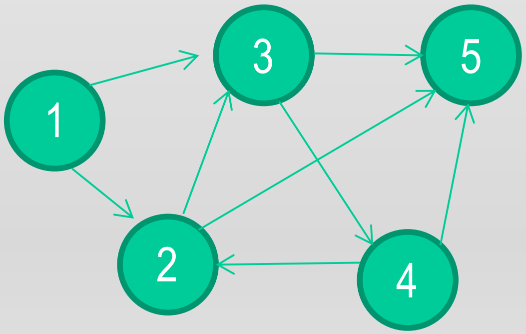
Network definitions

- A network consist of set of **nodes** linked by **arcs** (or **branches**)

- The notion for describing a network is (N, A) where:
  - N is set of nodes
  - A set of arc

Network definitions (cont.)

- Example



N ={ 1,2,3,4,5}

A={(1,2), (1,3),(2,3),(2,5),(3,4),(3,5),(4,2),(4,5)}

- *Flow* : the amount sent from node i to node j, over an arc that connects them.

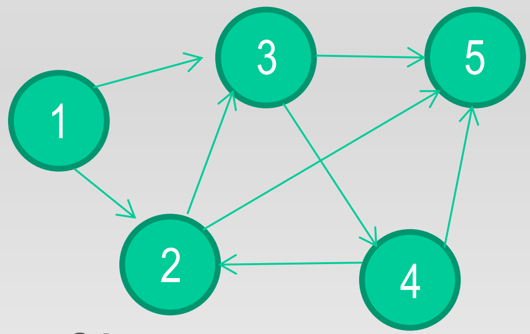
Network definitions (cont.)

- *Directed/undirected arcs* :
  - when flow is allowed in one direction the arc is **directed;** (that means allow positive flow in one direction and zero flow in the opposite direction)
  - When flow is allowed in two directions, the arc is **undirected.**

- *Path* : sequence of distinct arcs that join two nodes through other nodes regardless of the direction of flow in each arcs
- The nodes are said to be **connected** if there is a path between them.

Network definitions (cont.)

- *Cycle* : a path starting at a certain node and returning to the same node without using any arc twice. (or connects a node to itself through other nodes)



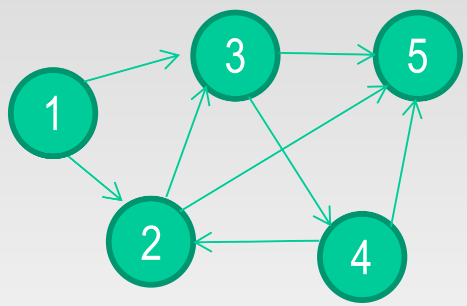
Example:

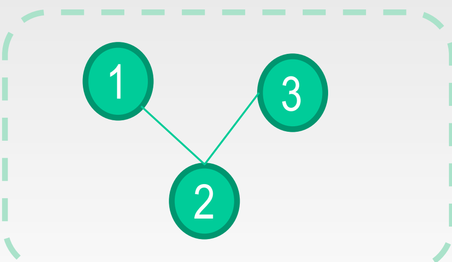
- (2,3),(3,5),(5.2) form of loop
- Cycle is directed if it consists of directed path (2,3),(3,4) and(4,2)

Network definitions (cont.)

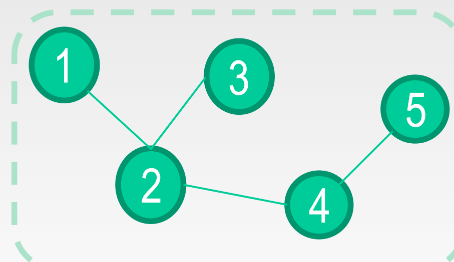
- *Tree* : is connected network that may involve only a subset of all nodes of network without cycle.

- *Spanning tree* : a tree that connects all the nodes in a network with no cycle(it consists of n -1 arcs).





Tree



Spanning Tree

Minimal Spanning tree

- It deals with linking the nodes of network, directly or indirectly, using shortest length of connecting branches.
- The typical application occurs in construction of paved roads that link several towns.

Minimal Spanning tree

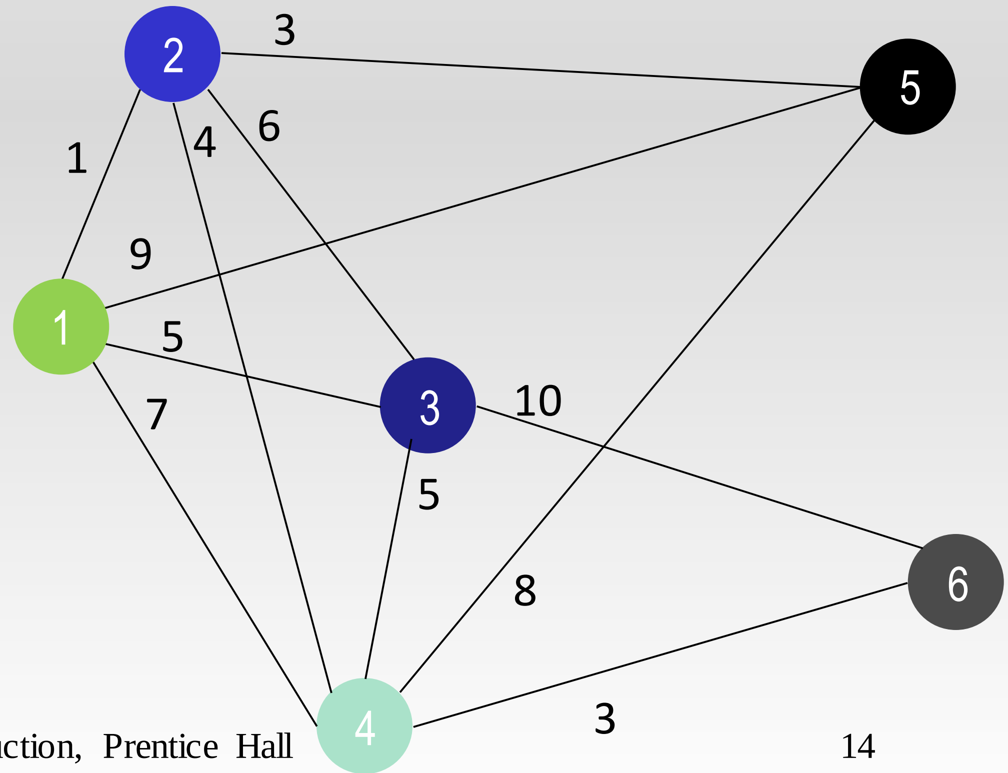
- The step of procedure are given as follows:
  - Let N={ 1,2,…n} set of nodes
  - Ck= set of nodes that have been permanently connected at iteration K
  - Ck`= set of nodes as yet to be connected permanently.
- Step 0: set C0= 0, C0`=N
- Step 1: start with any node I; set C1={i}, C1`=N-{i}
- General step: selected node j in unconnected set

  Ck-1` that yield in shortest arcs to a node in the connected set . Link j permanently to Ck-1 and remove it from Ck-1`

- If the set of unconnected nodes is empty stop. Otherwise set k=K+1 and repeat the step

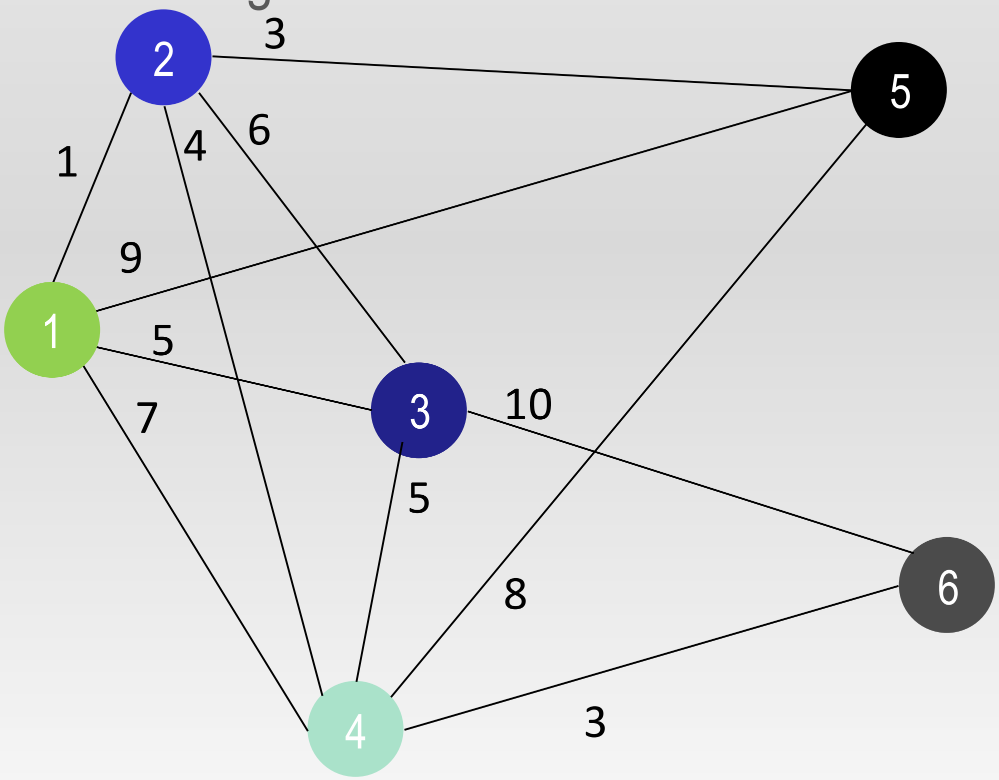
Example (cont.)

- Midwest TV cable company is in the process of providing cable service to five new housing development service areas.



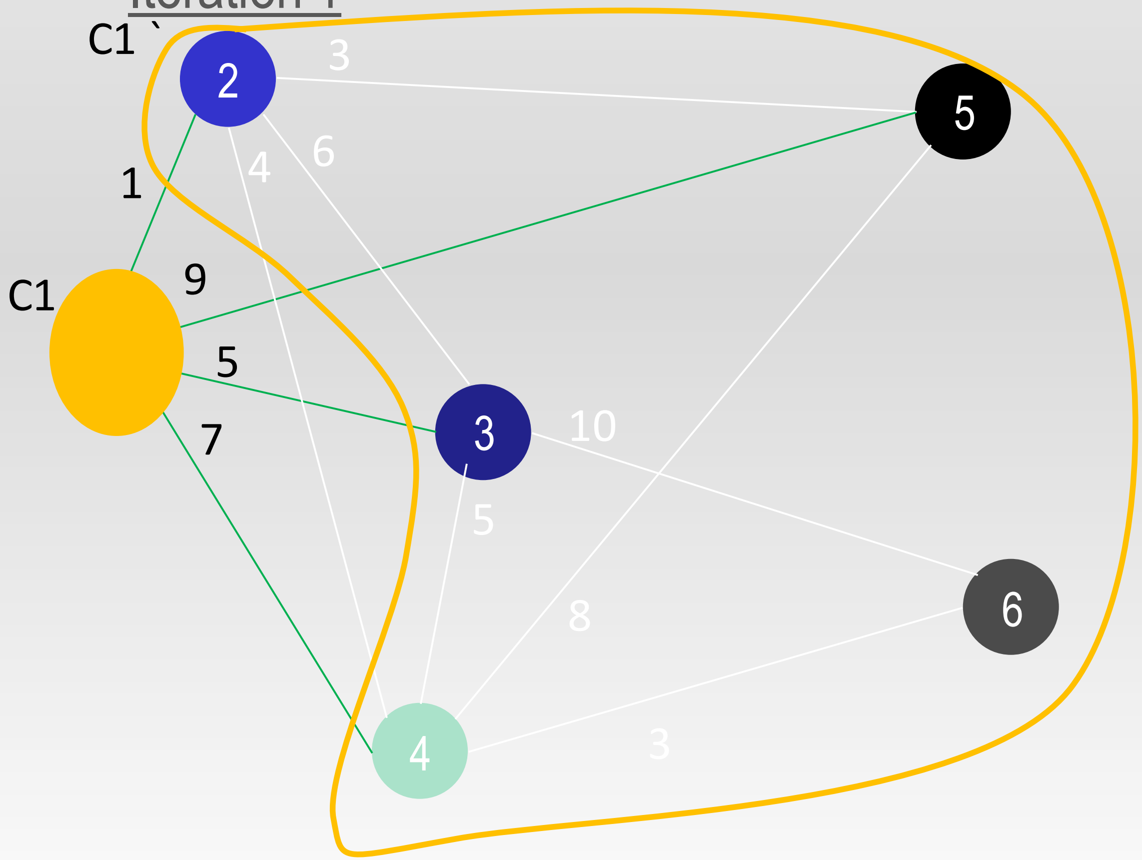
Example (cont.)

- The algorithms start at node 1

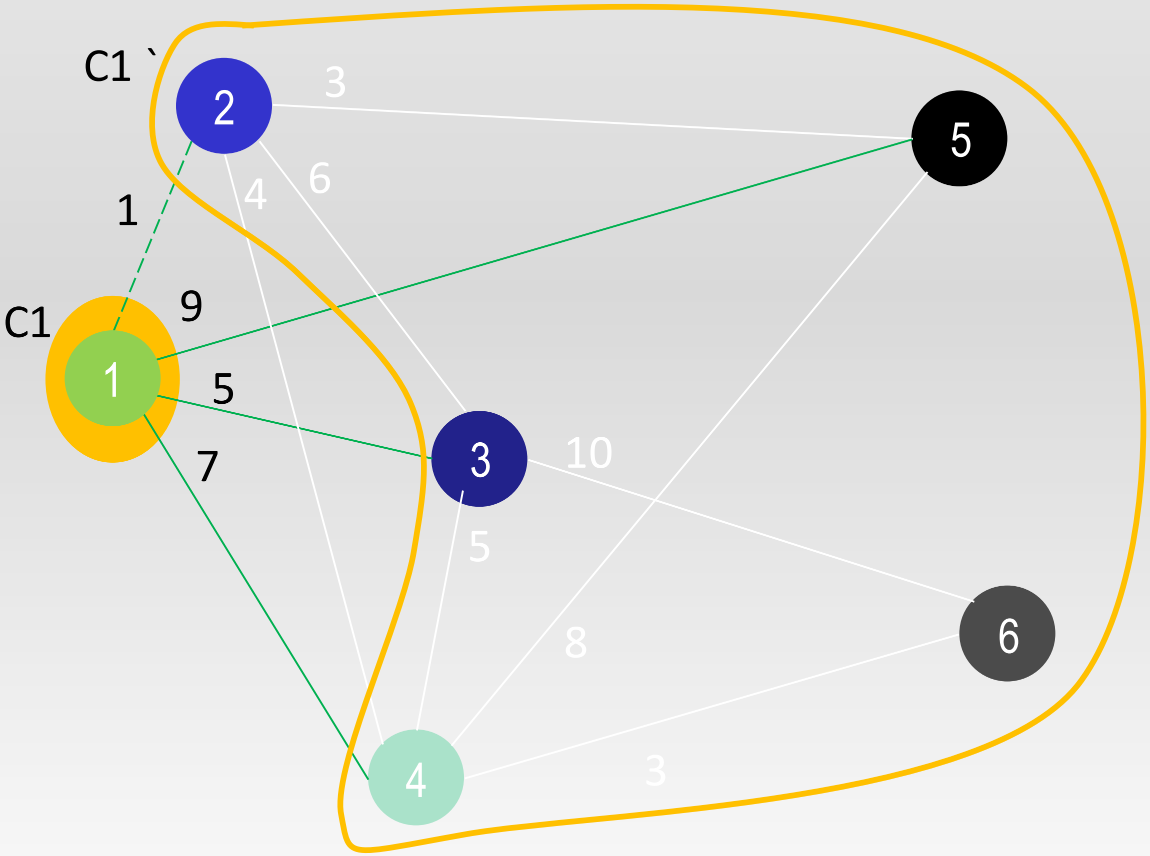


Example (cont.)

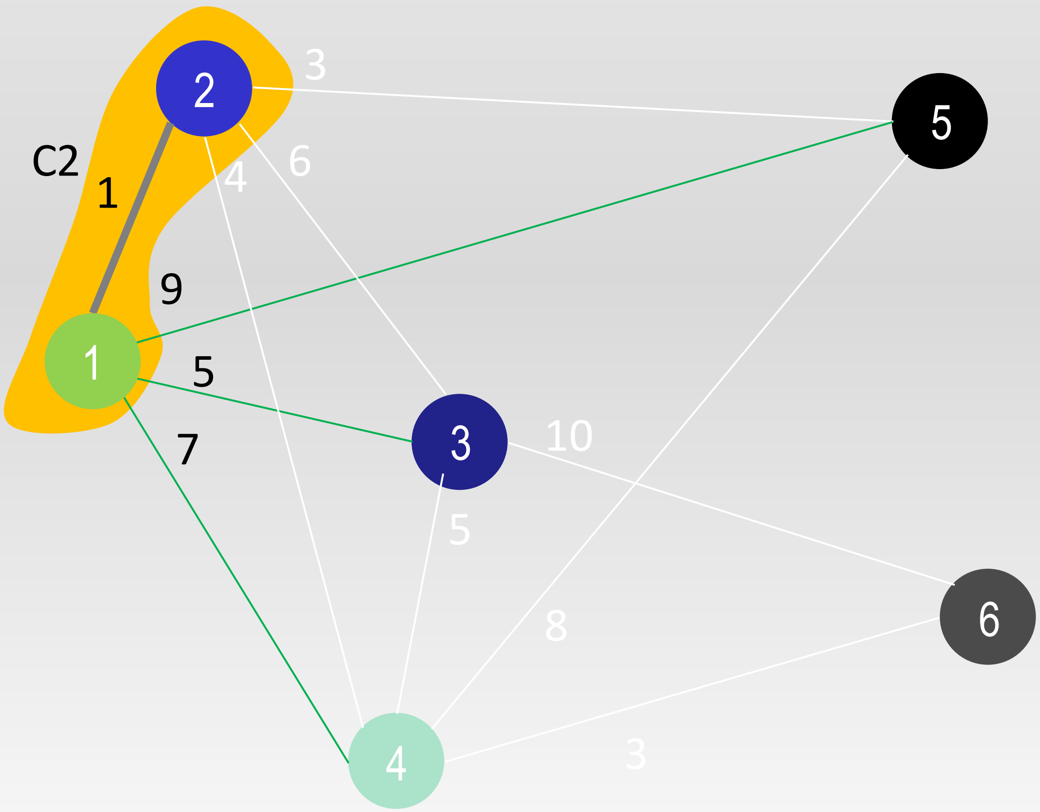
- Iteration 1



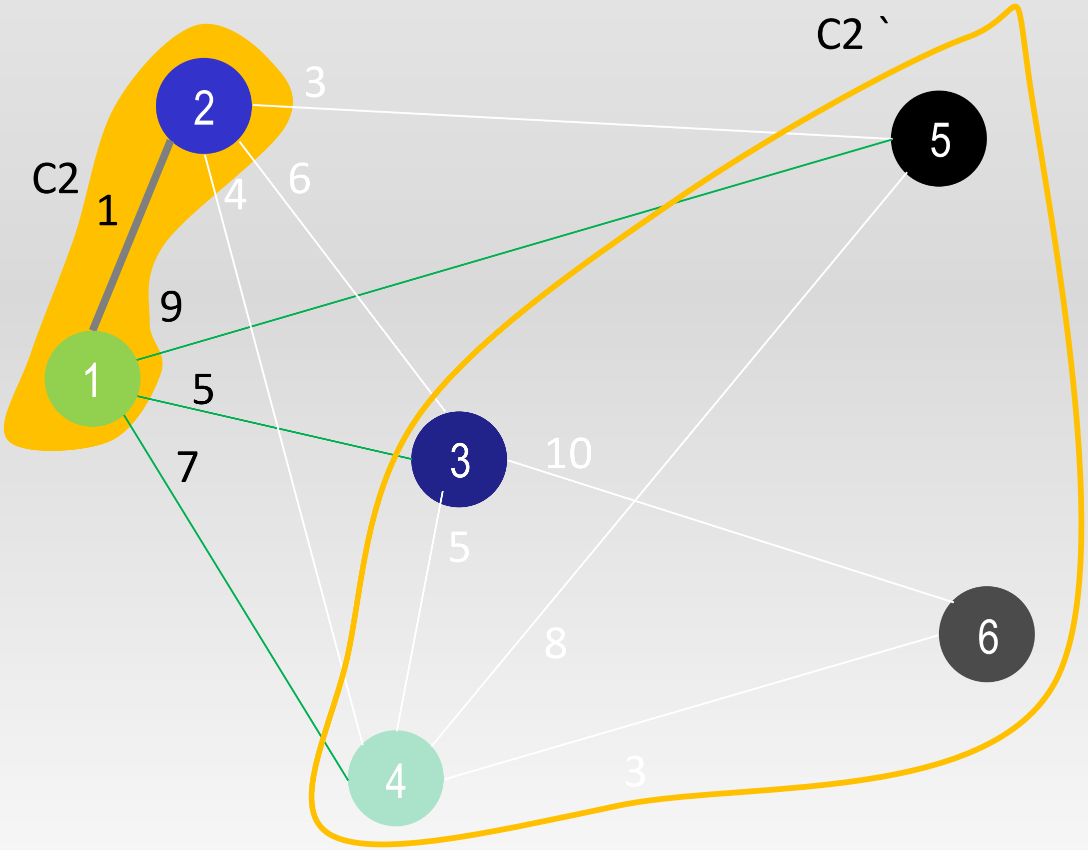
Example (cont.)

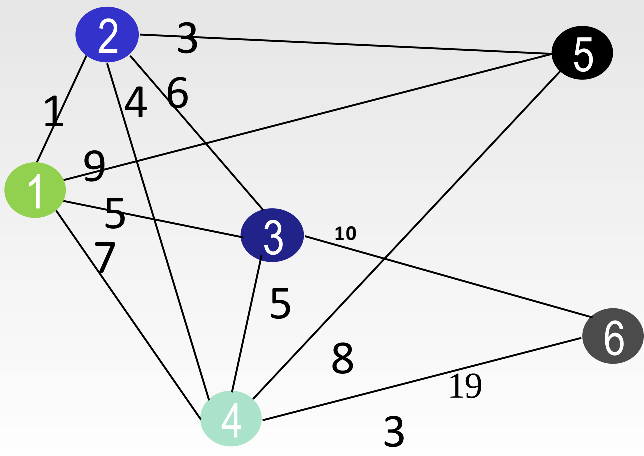


Example (cont.)



Example (cont.)

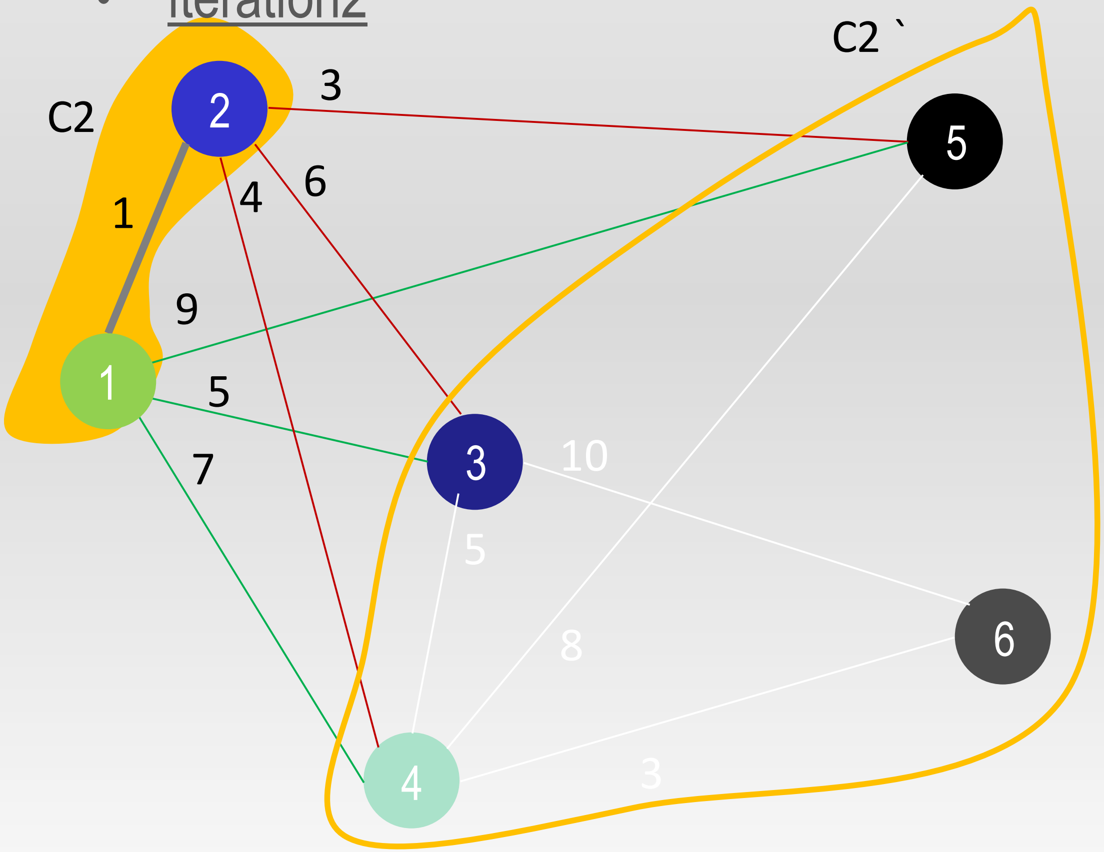




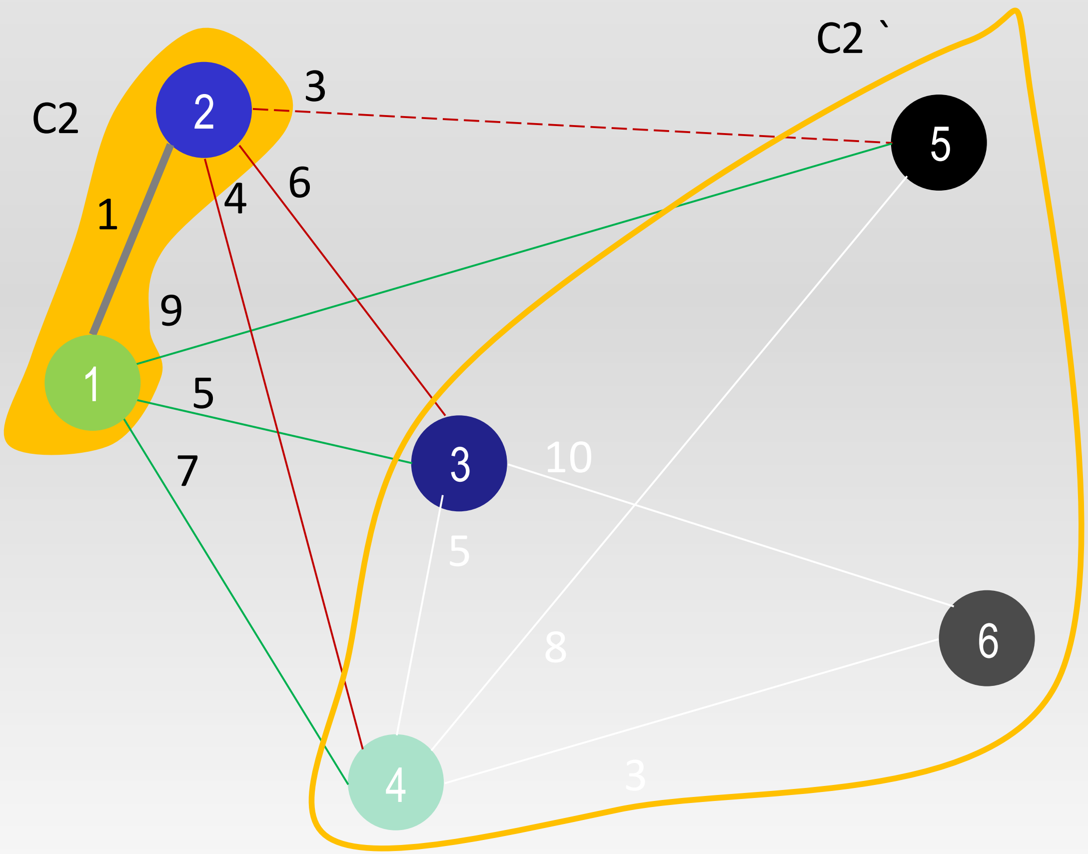
Hamdy A. Taha, Operations Research: An introduction, Prentice Hall

Example (cont.)

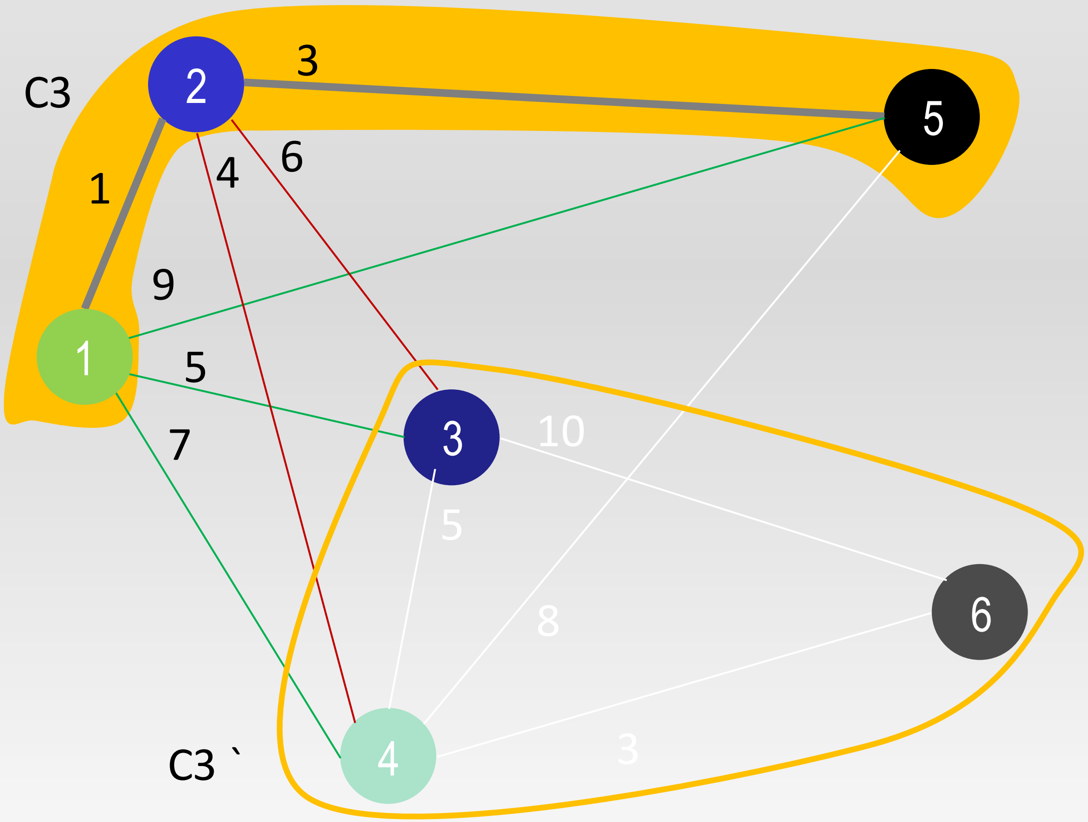
- iteration2

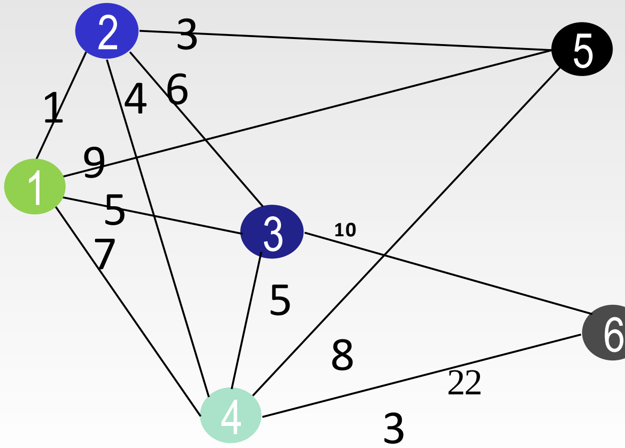


Example (cont.)



Example (cont.)

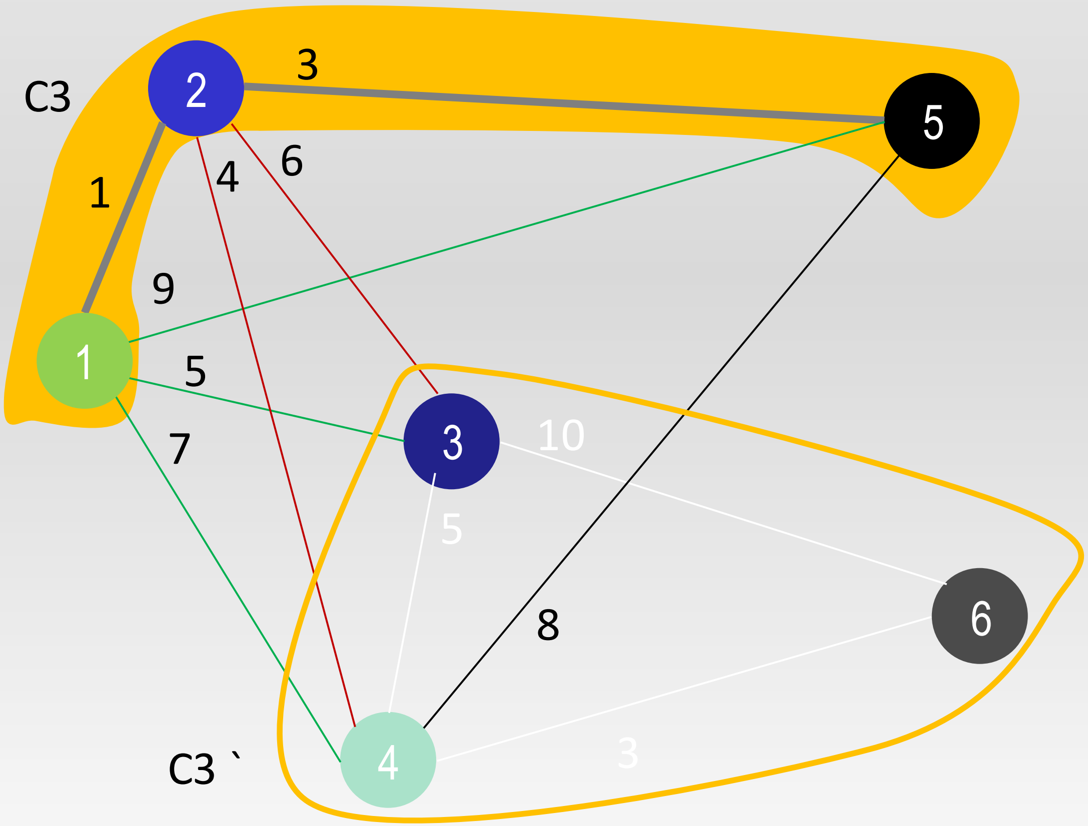




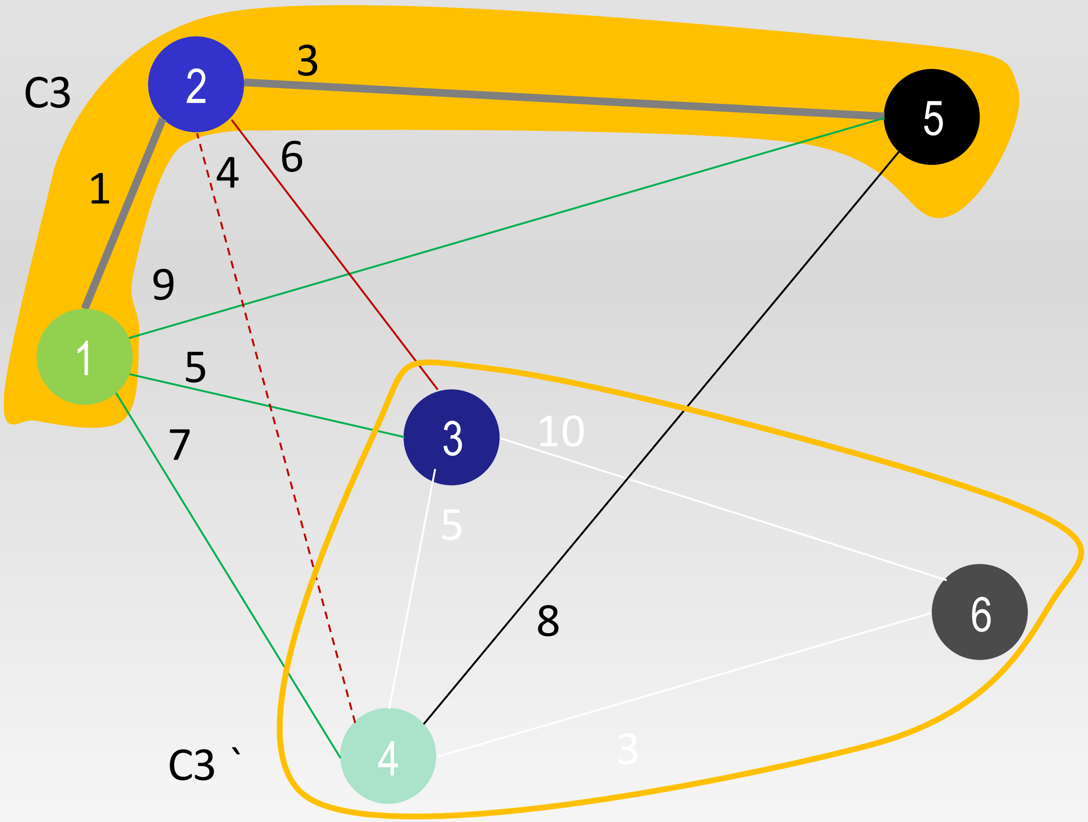
Hamdy A. Taha, Operations Research: An introduction, Prentice Hall

Example (cont.)

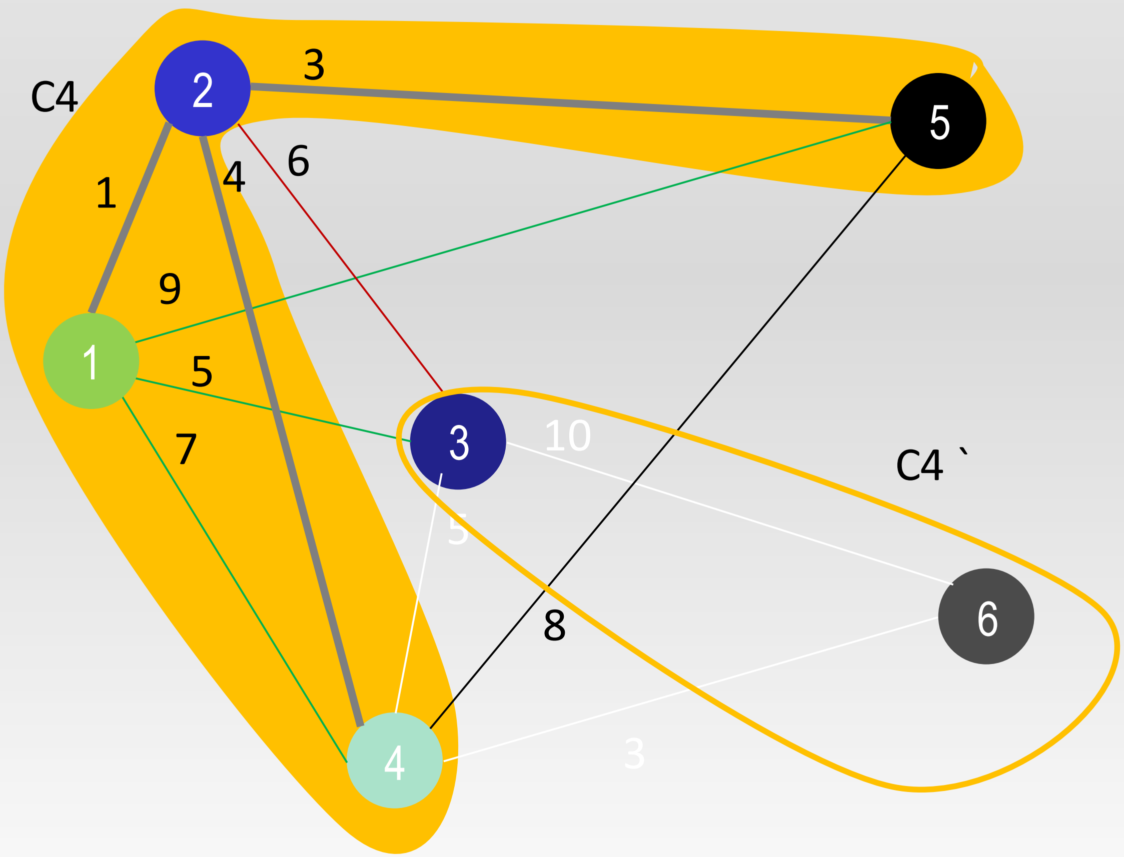
- iteration3



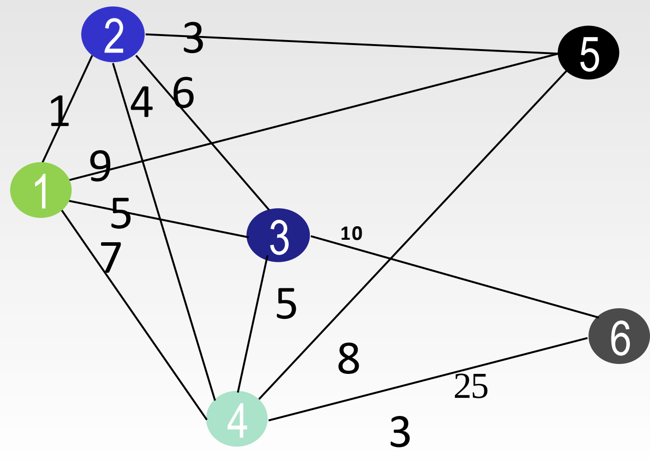
Example (cont.)



Example (cont.)

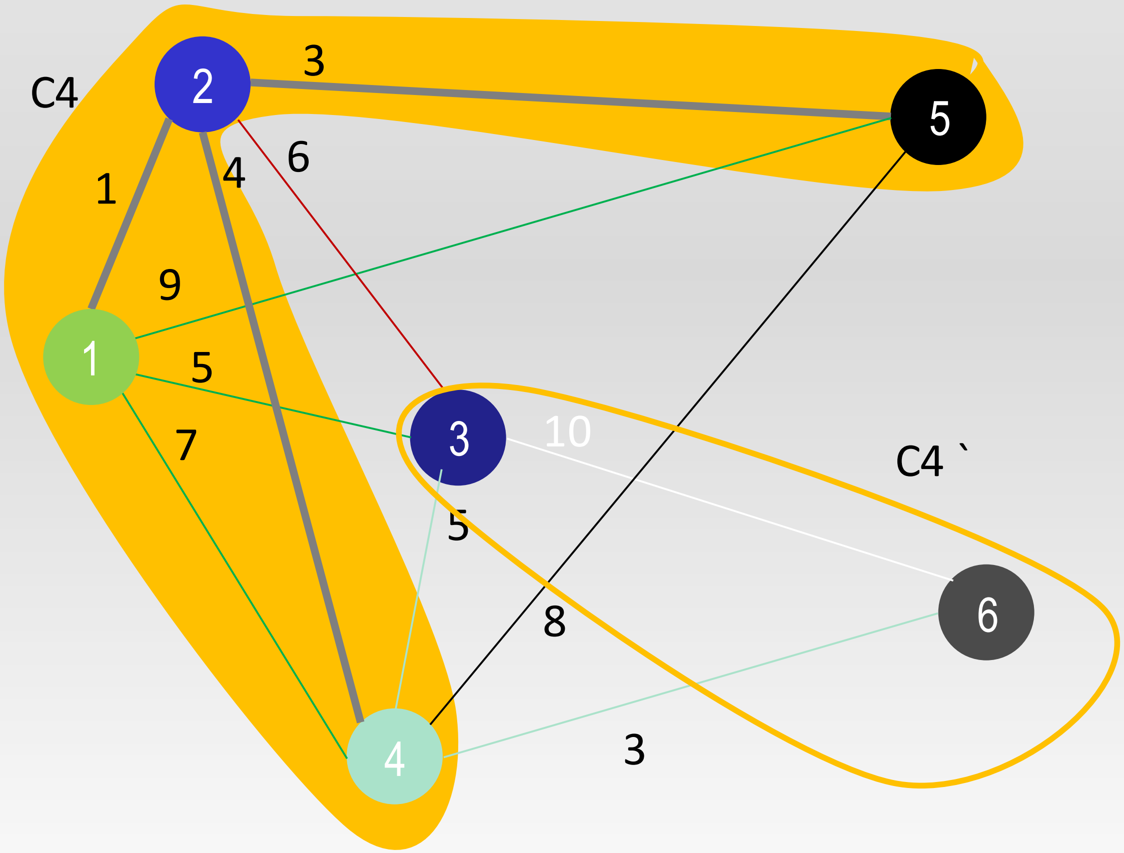


Hamdy A. Taha, Operations Research: An introduction, Prentice Hall

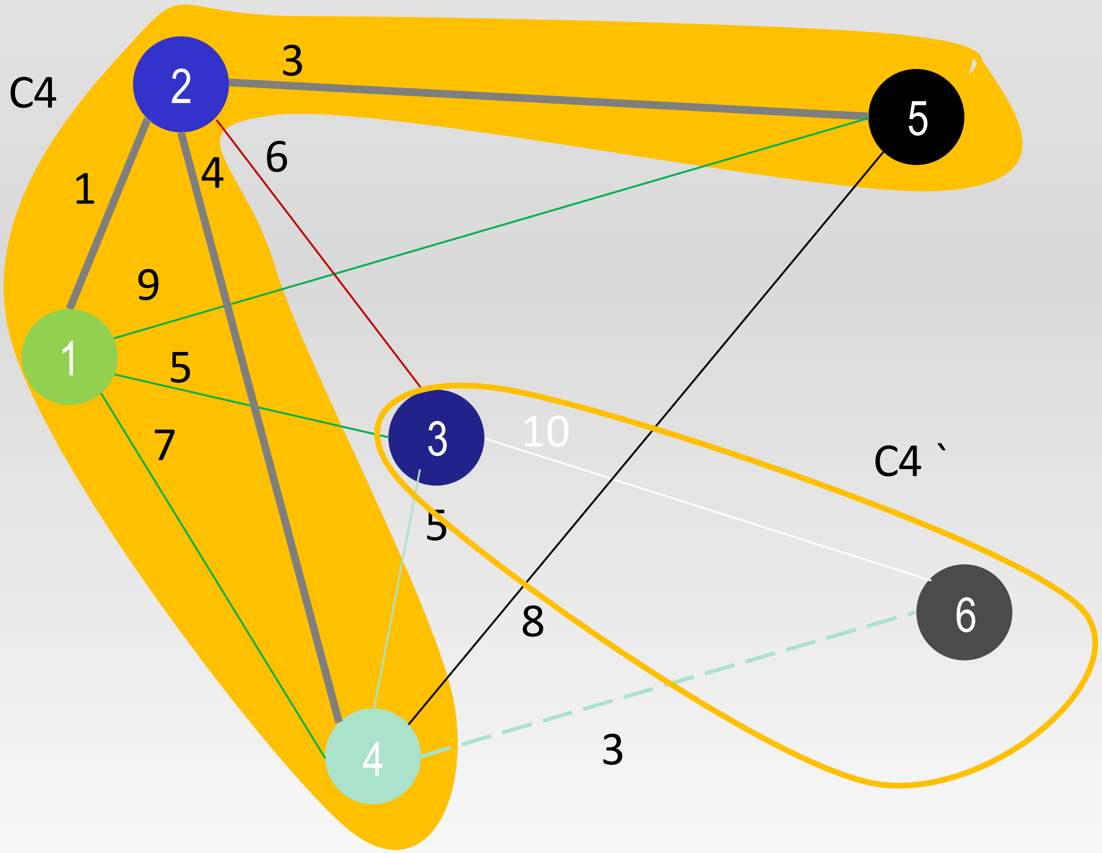


Example (cont.)

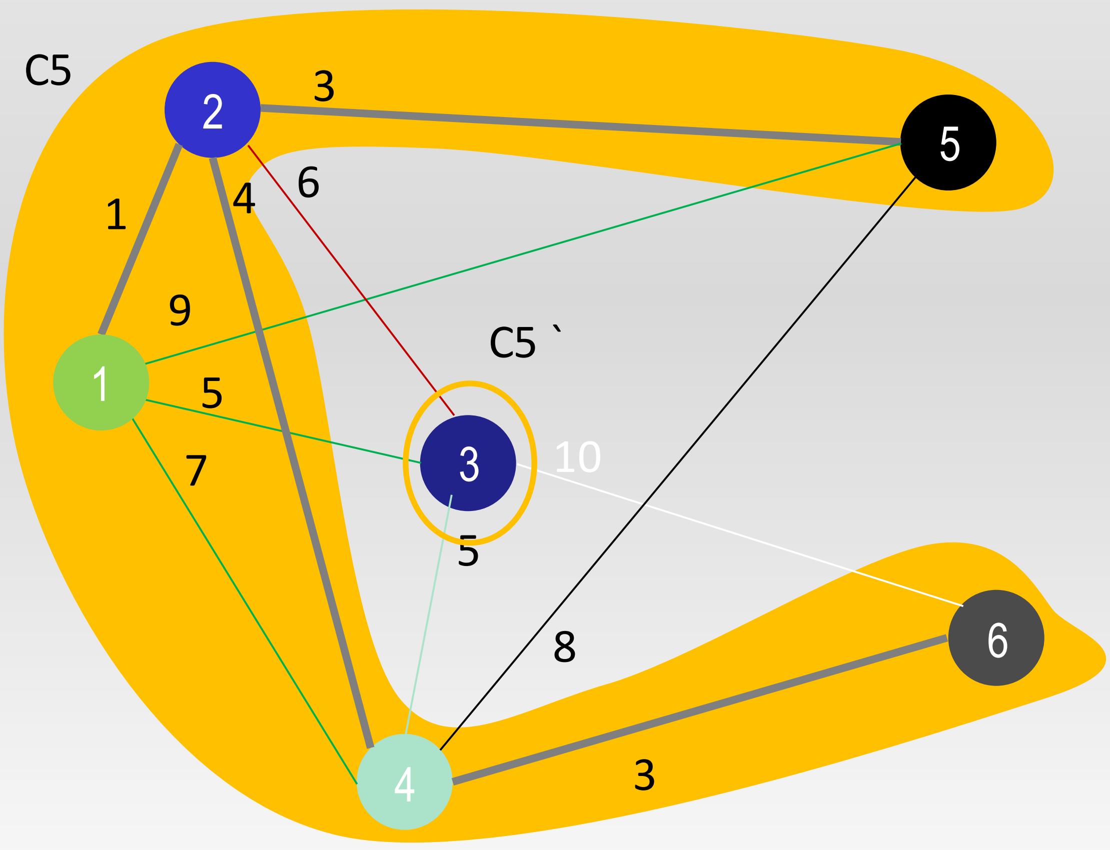
- iteration4

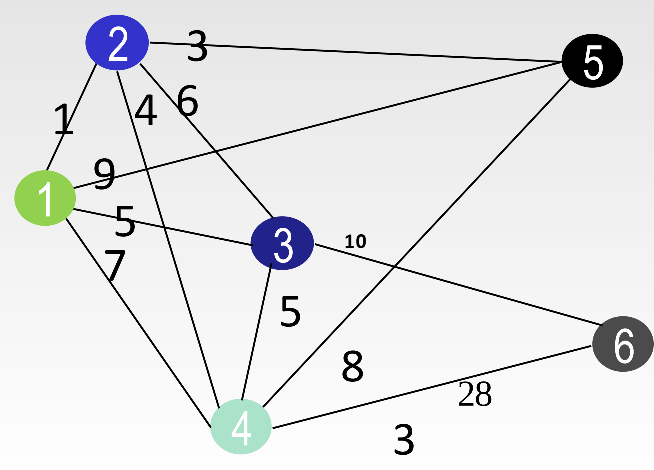


Example (cont.)



Example (cont.)

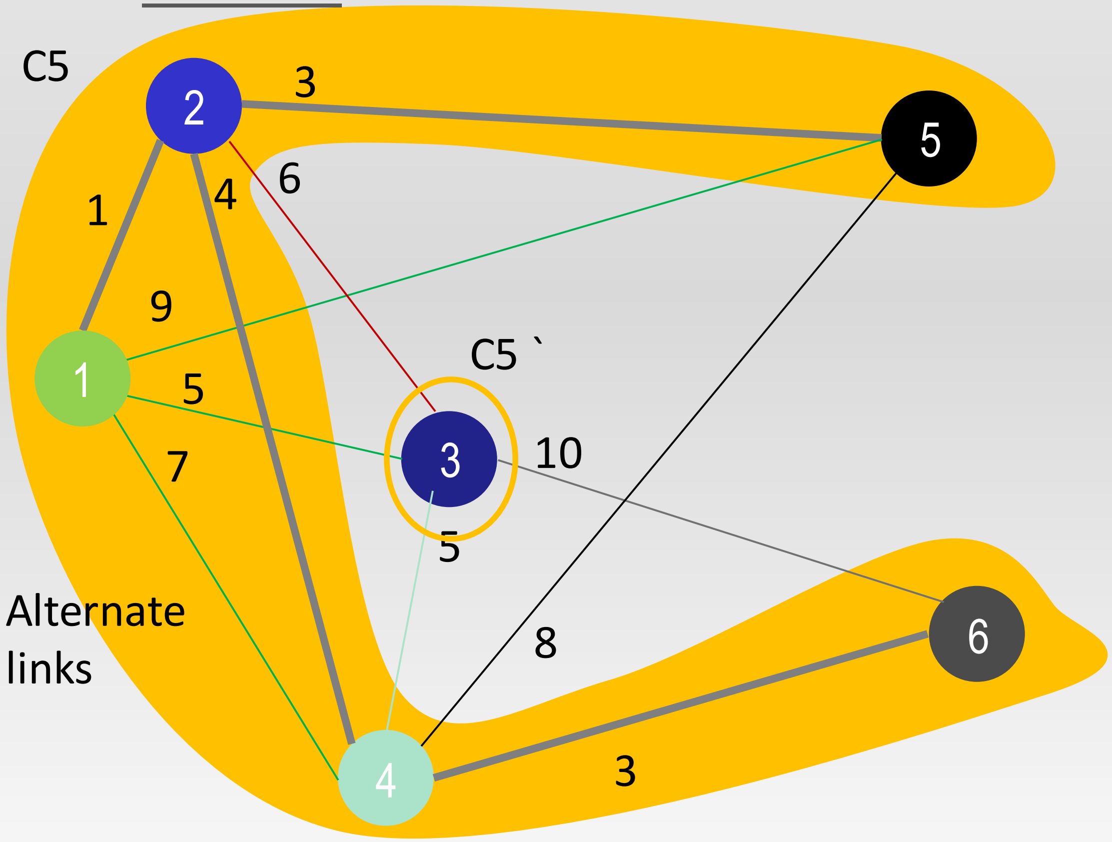




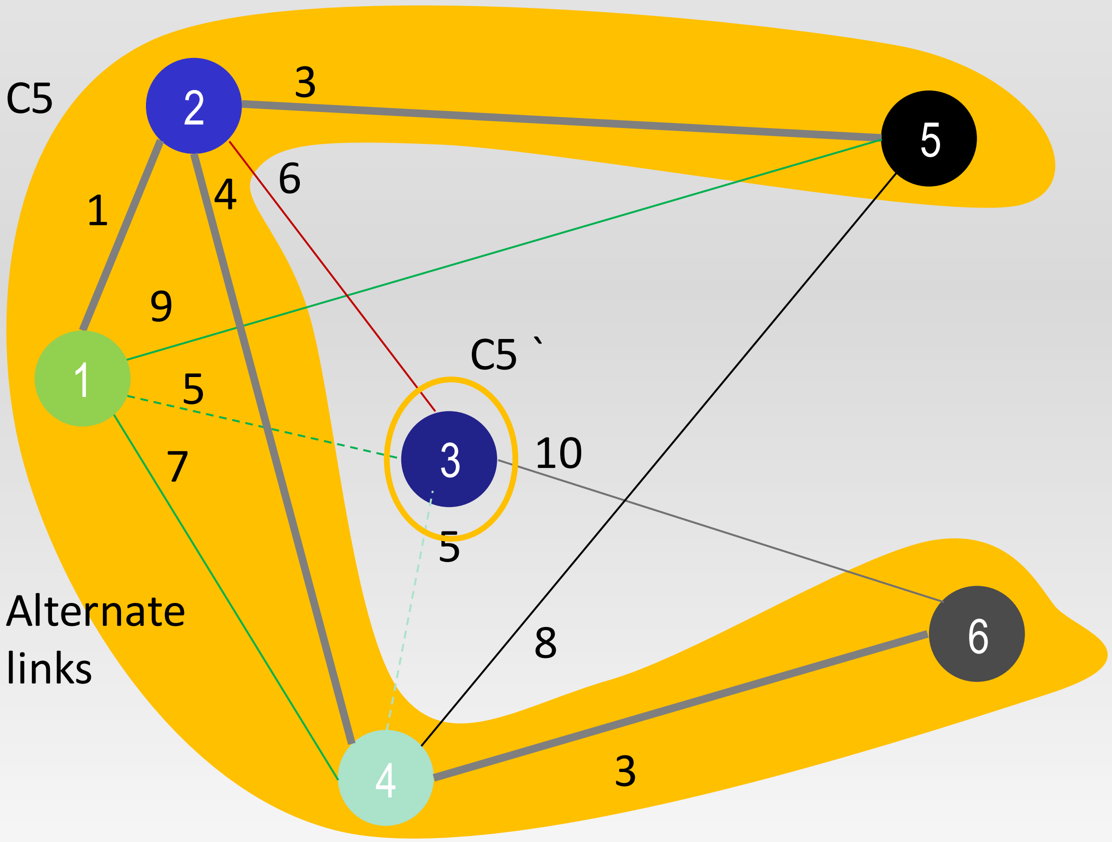
Hamdy A. Taha, Operations Research: An introduction, Prentice Hall

Example (cont.)

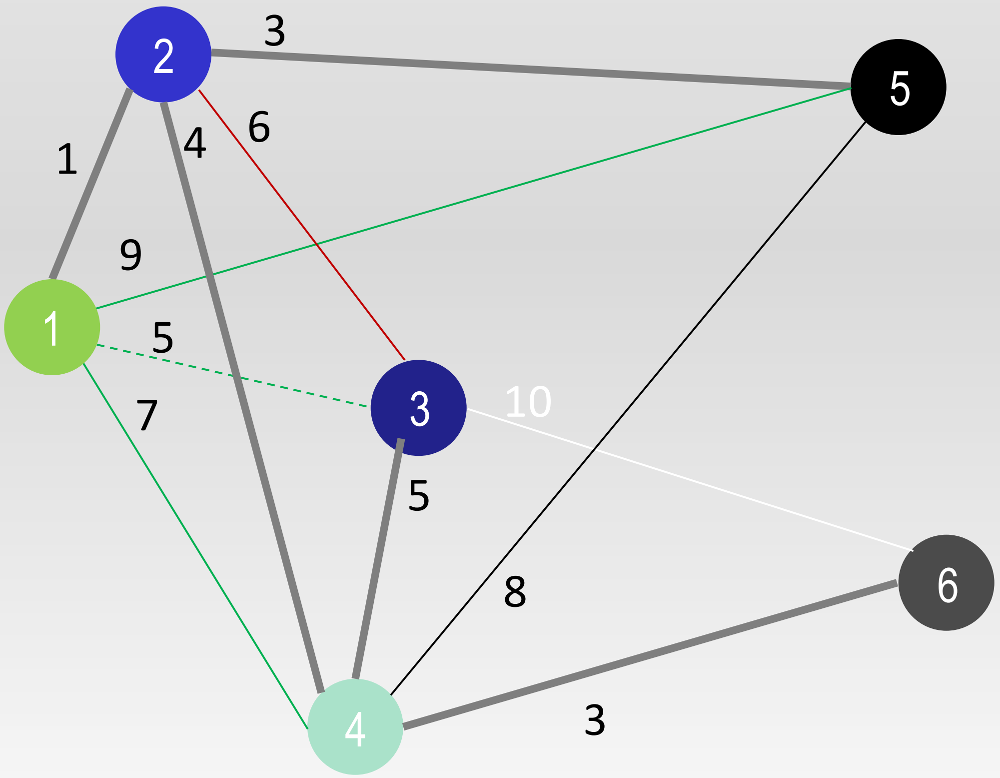
- iteration5



Example (cont.)



Example (cont.)



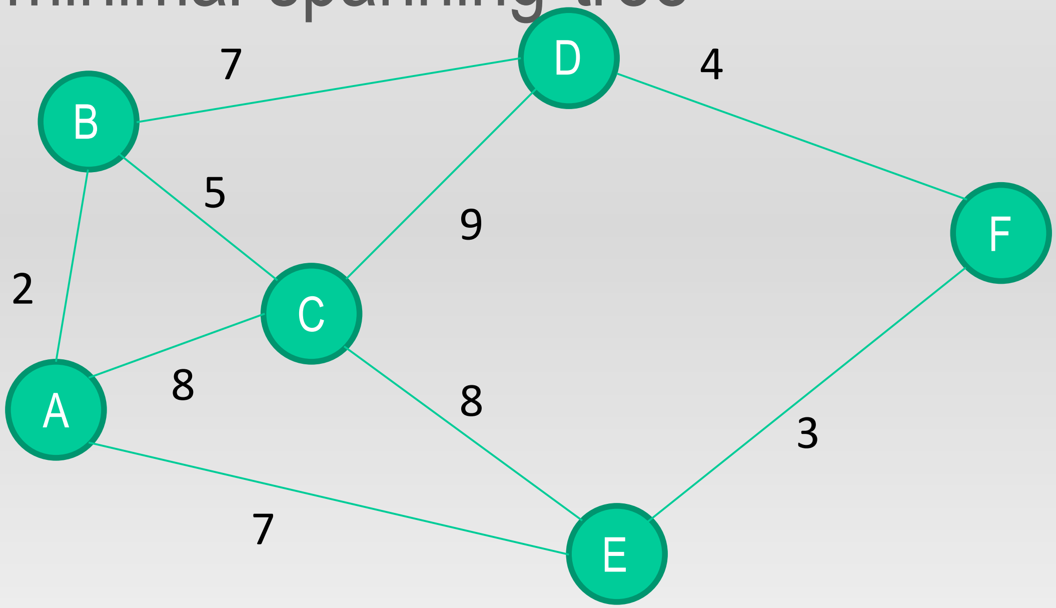
Example (cont.)

- Summery of solution

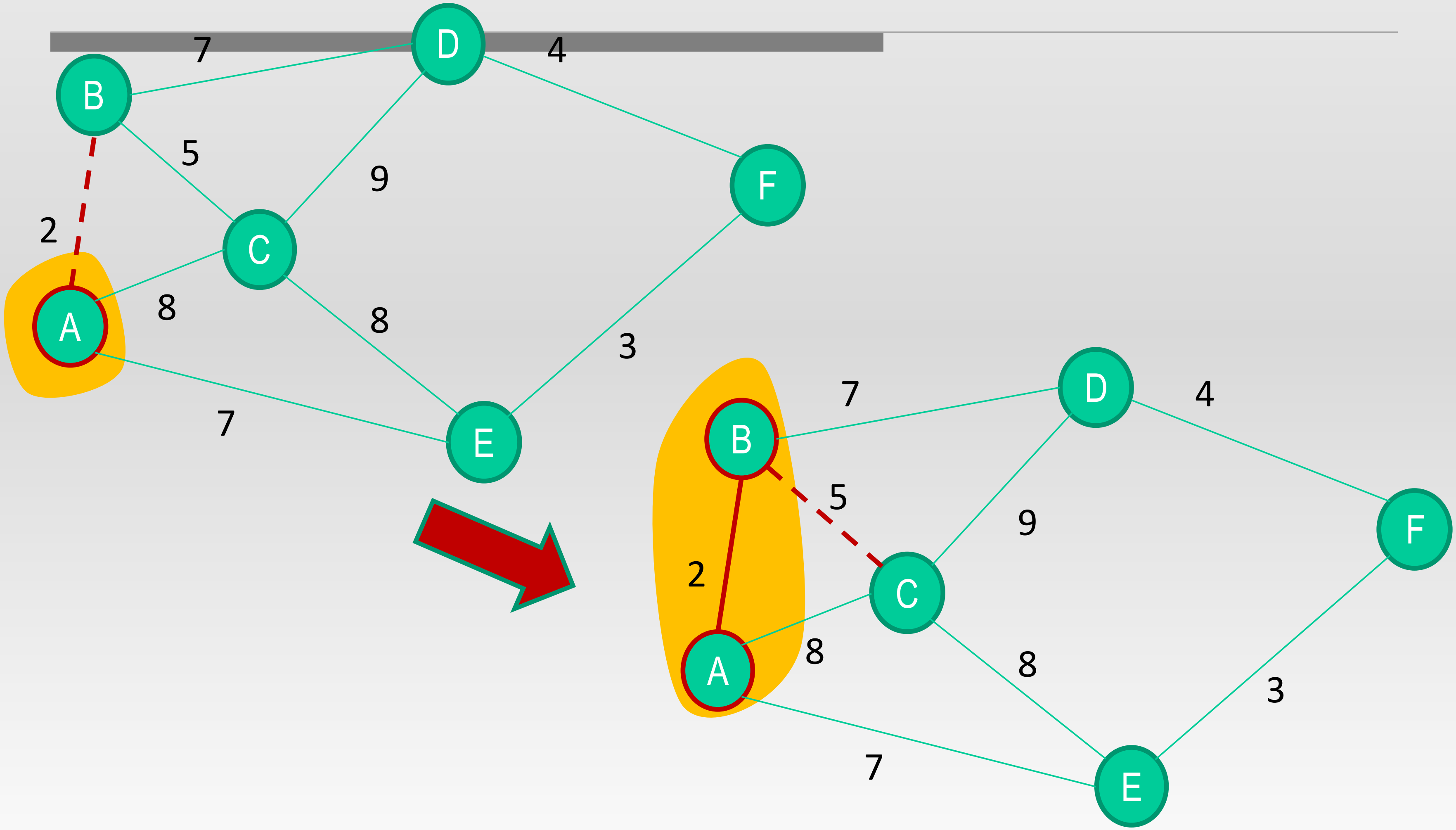
iteration	Minimum distance connecting arc	distance	Add arc to tree?	Cumulative tree distance
1	(1,2)	1	yes	1
2	(2,5)	3	yes	4
3	(2,4)	4	yes	8
4	(4,6)	3	yes	11
5	(4,3)	5	yes	16

Example 2

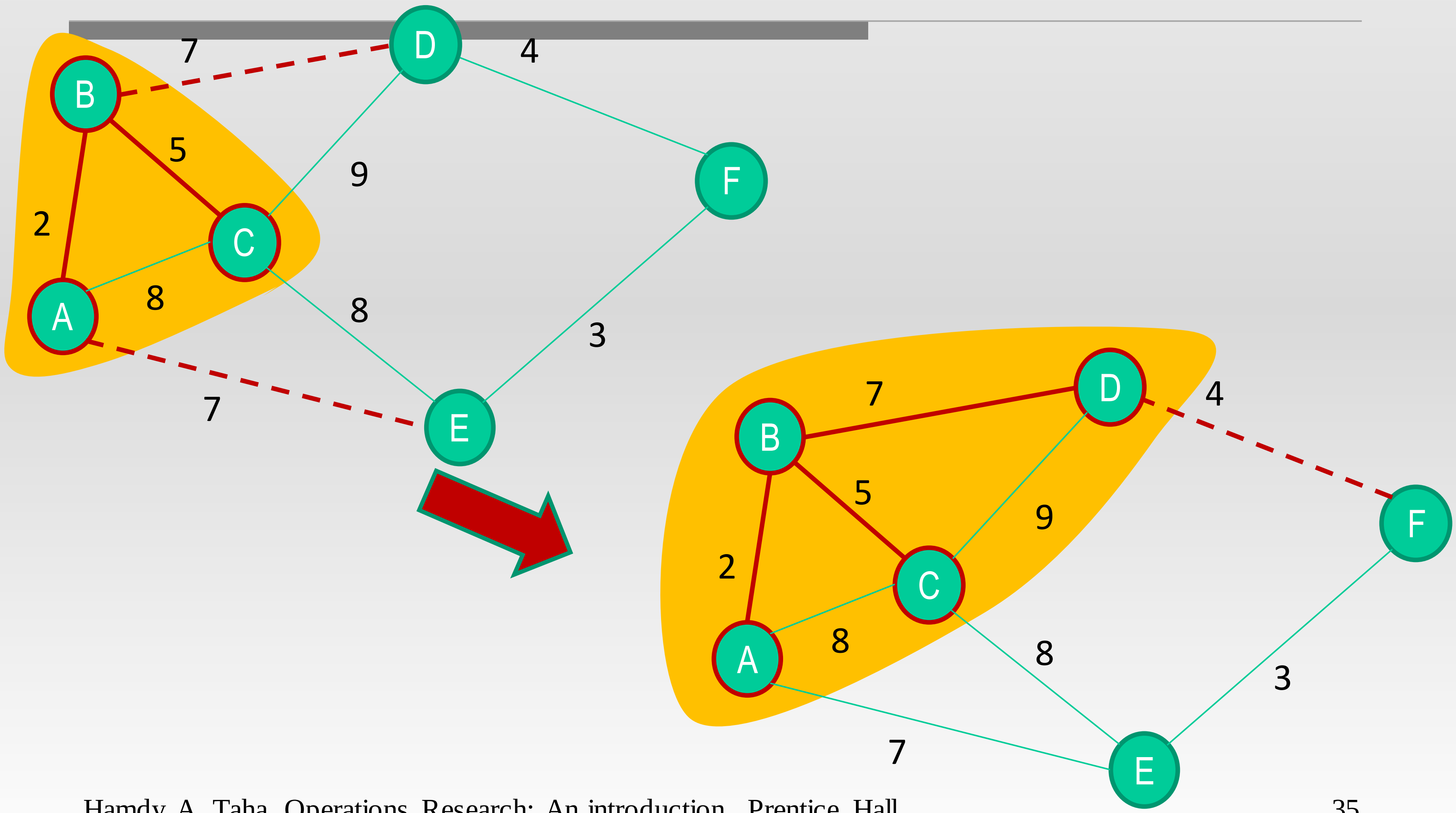
- Apply minimal spanning tree



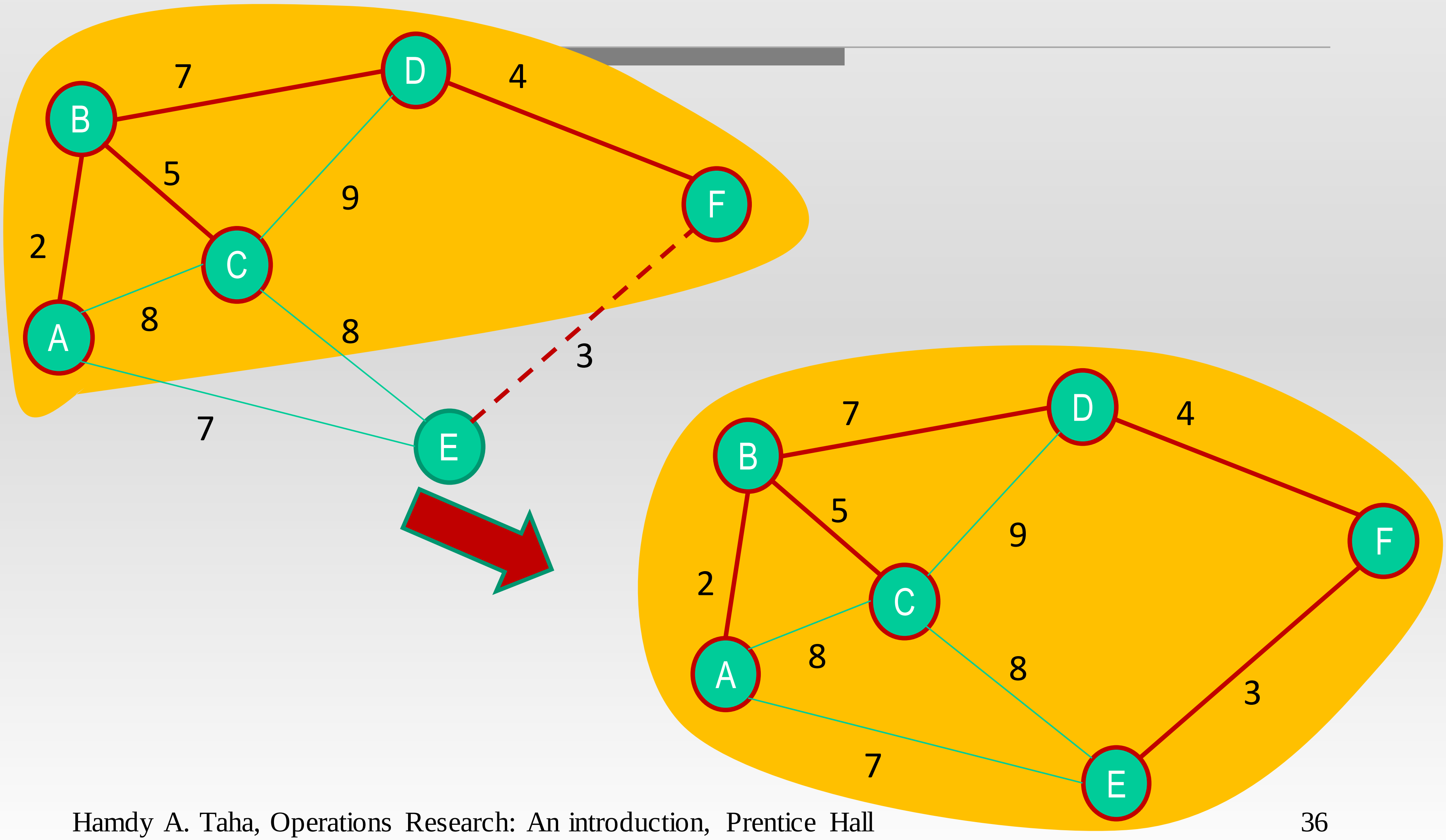
Solution



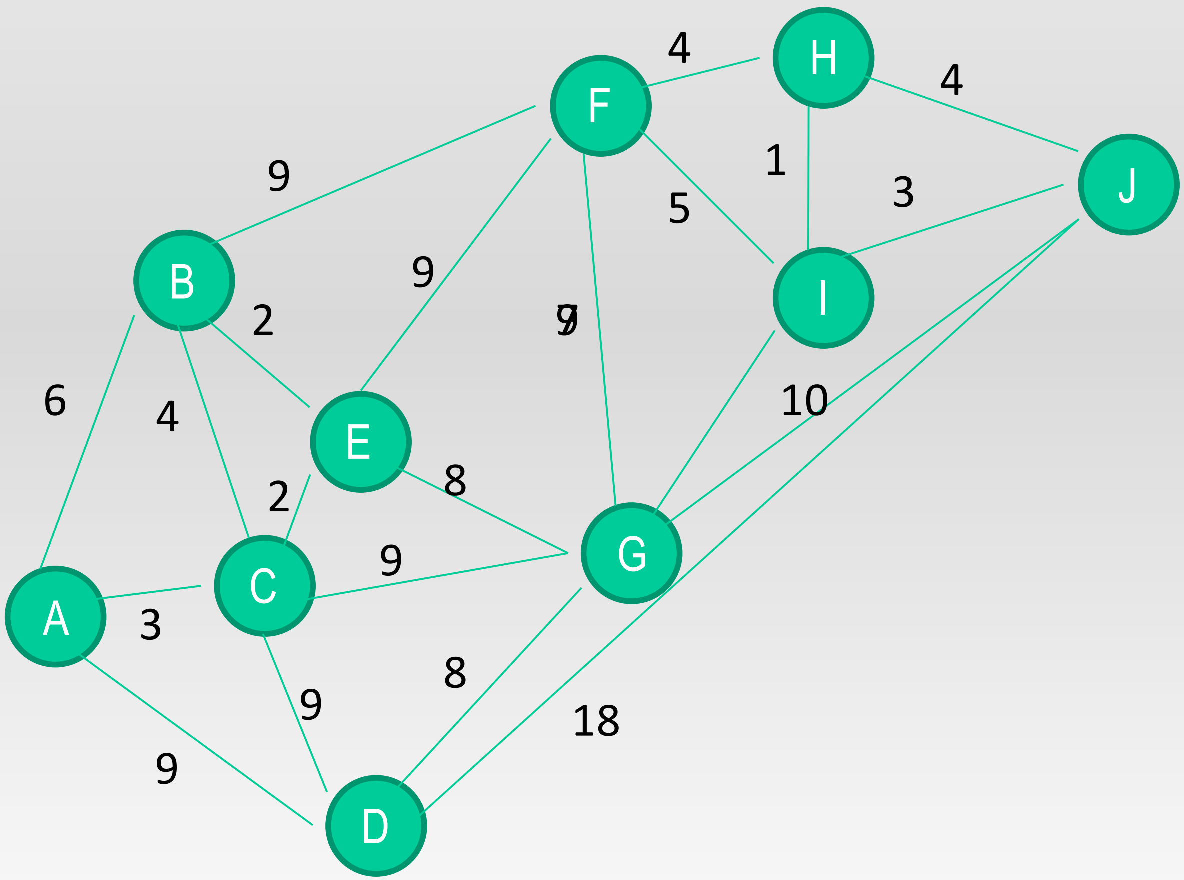
Solution (cont.)



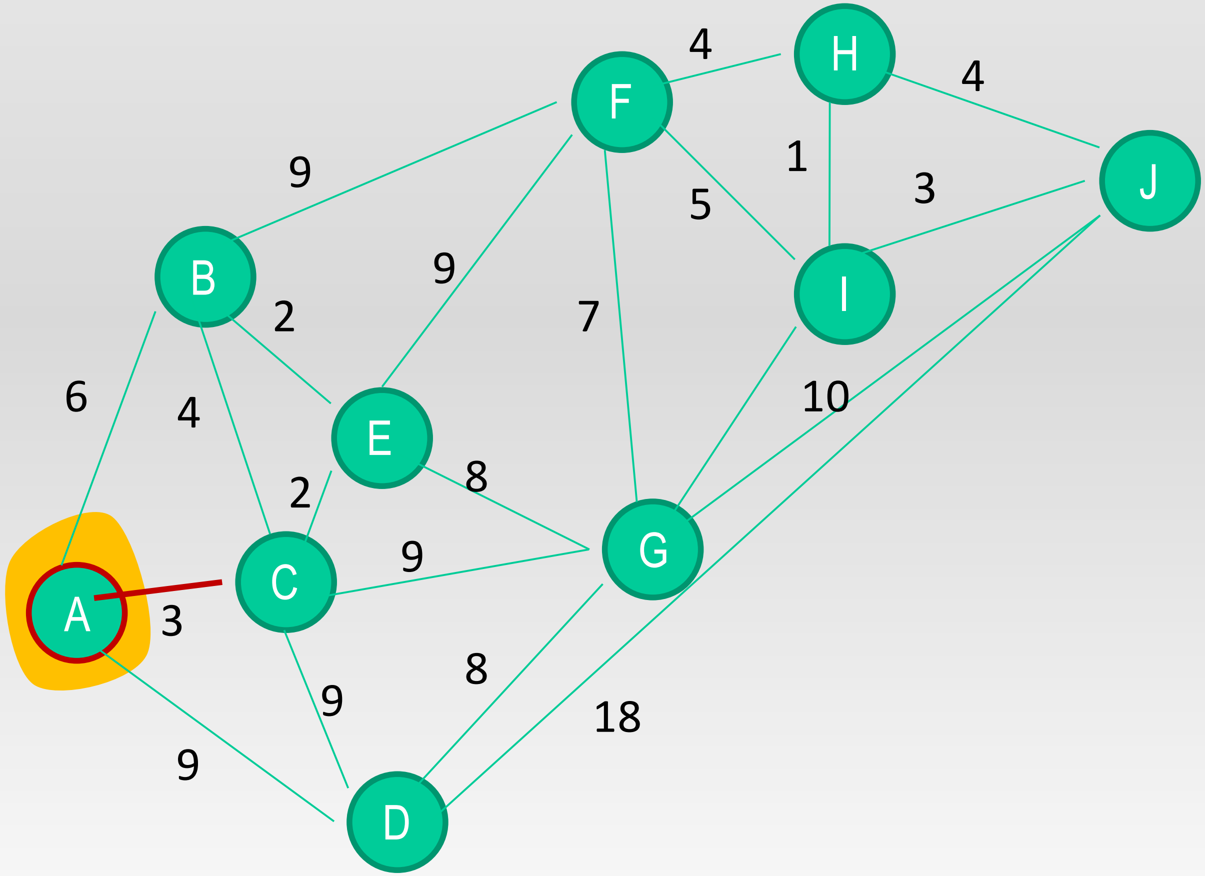
Solution (cont.)



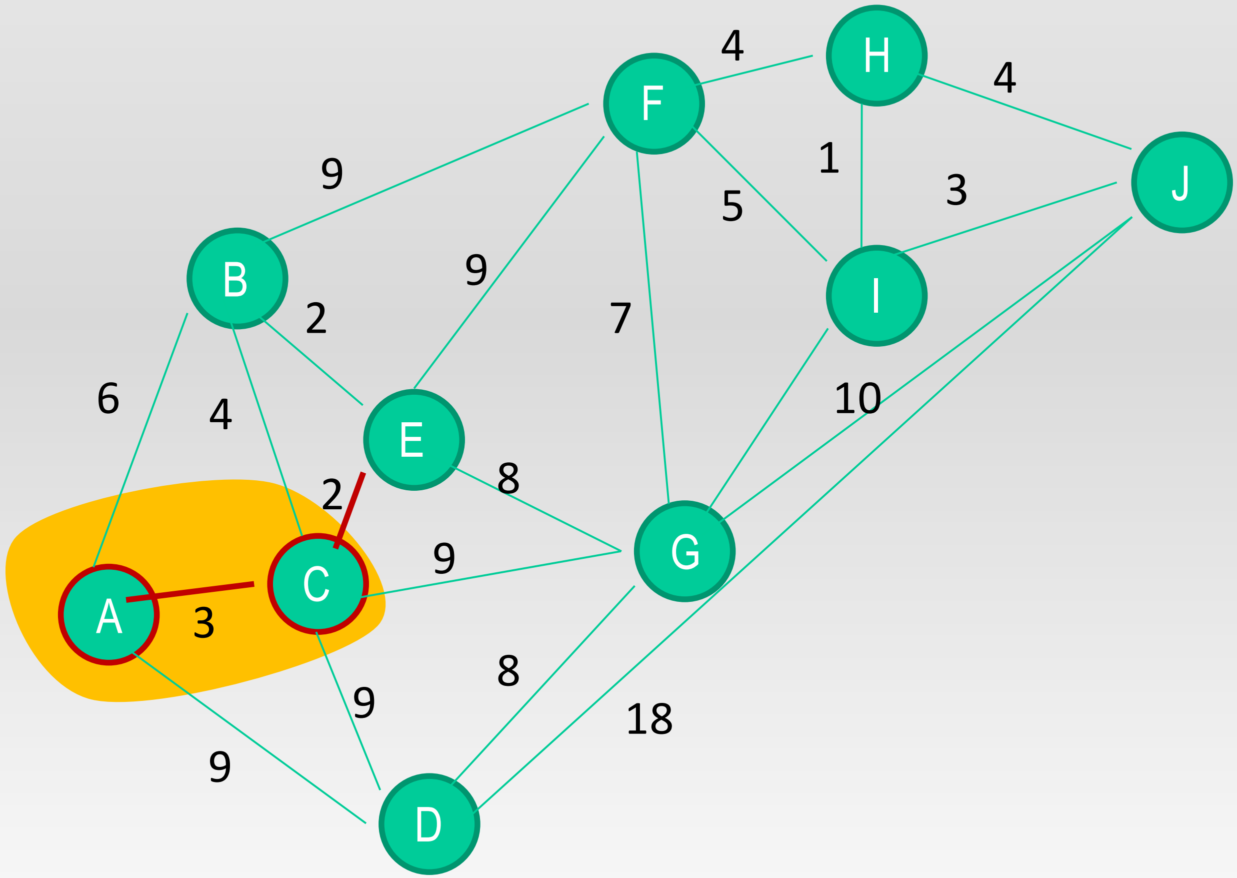
Example 3



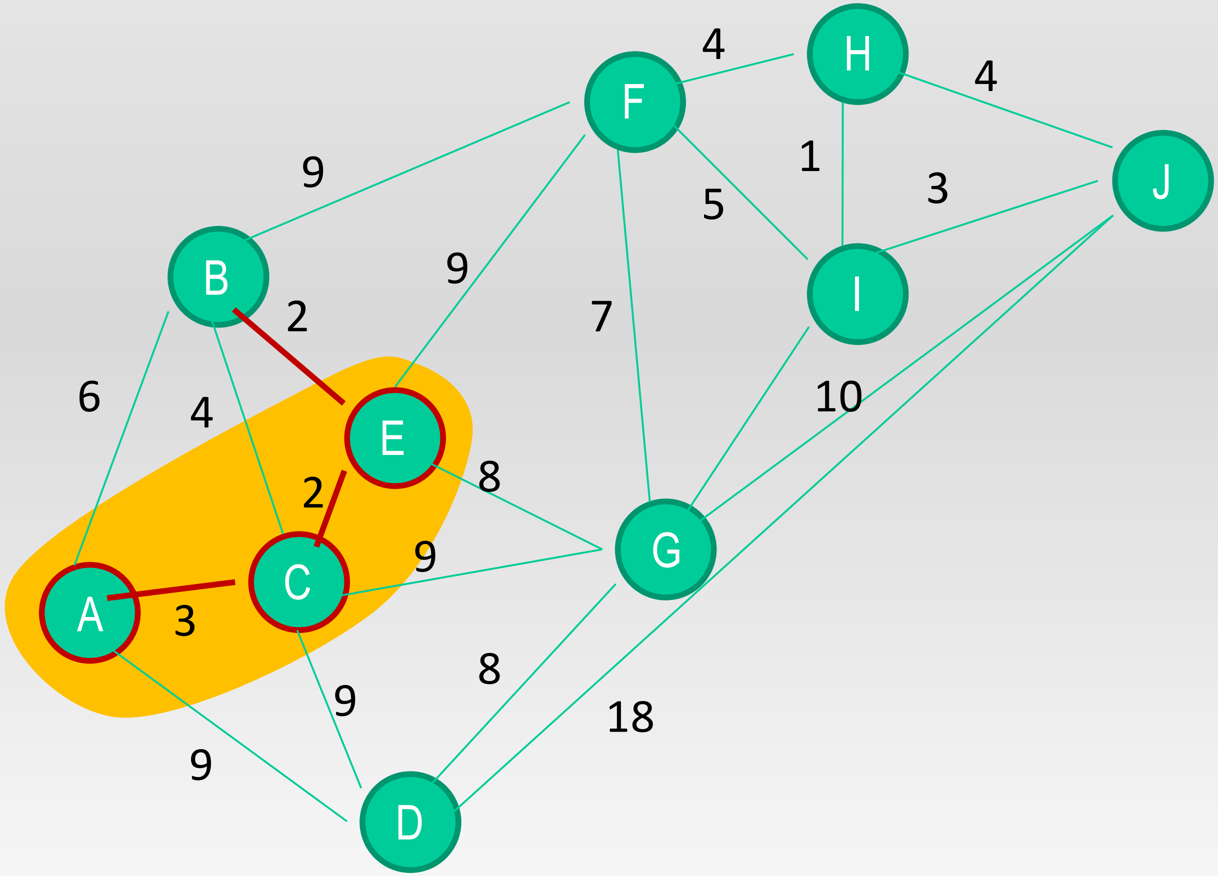
Solution



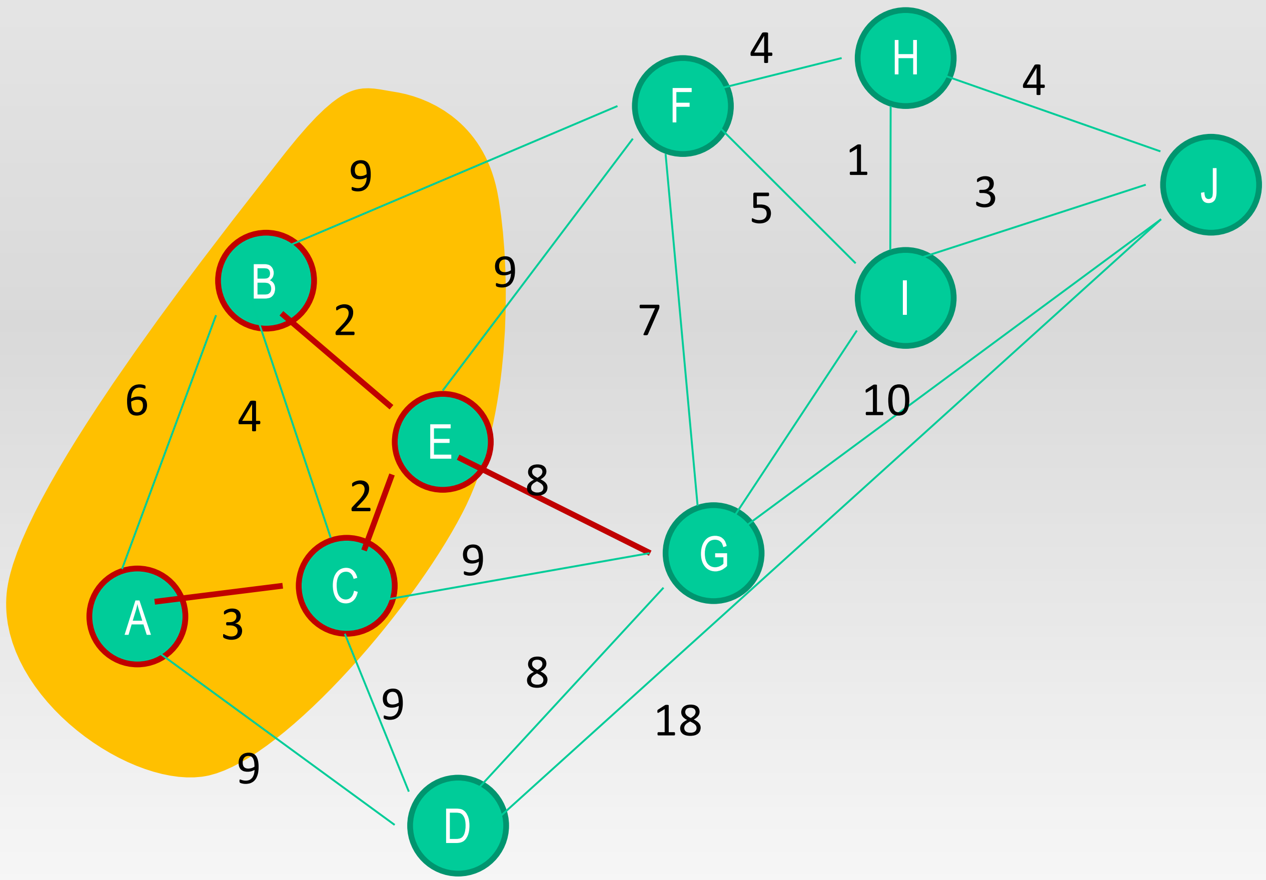
Solution (cont.)



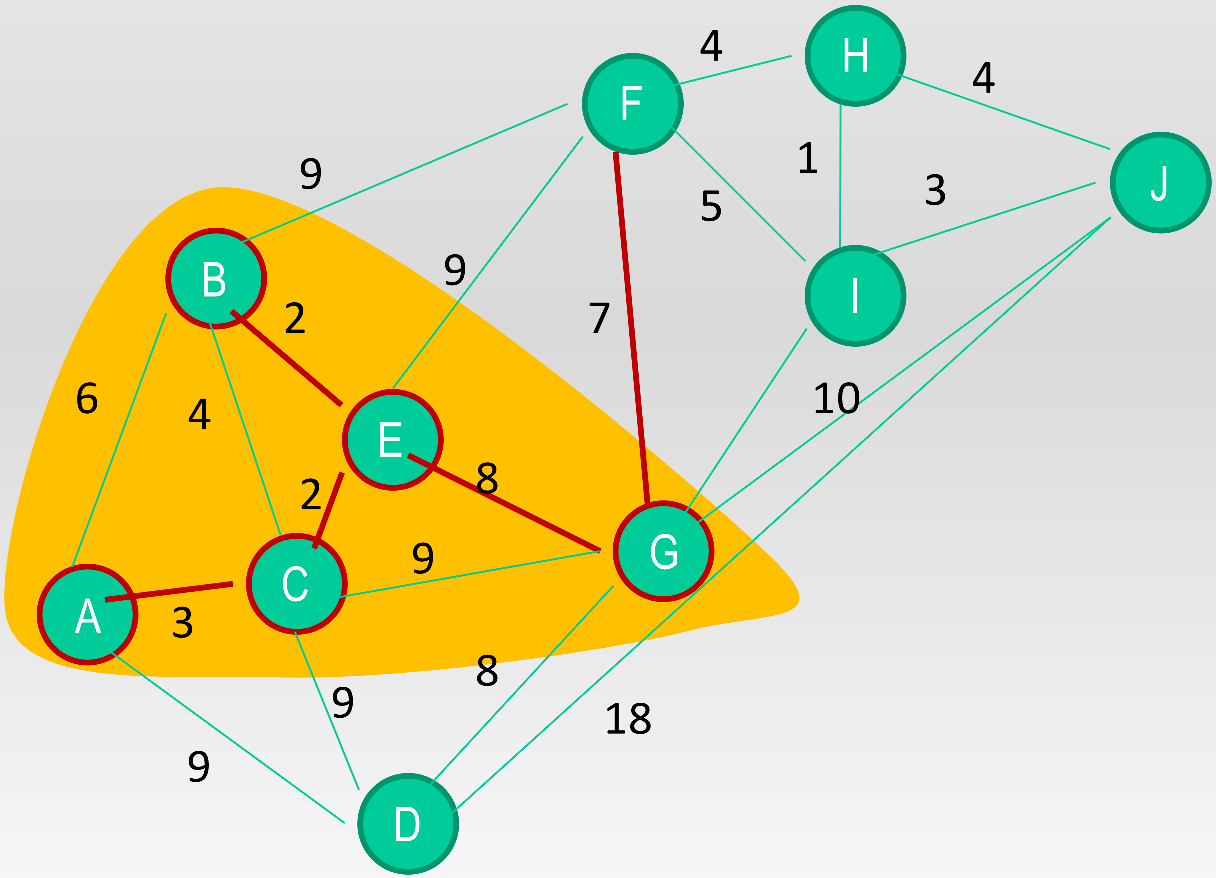
Solution (cont.)



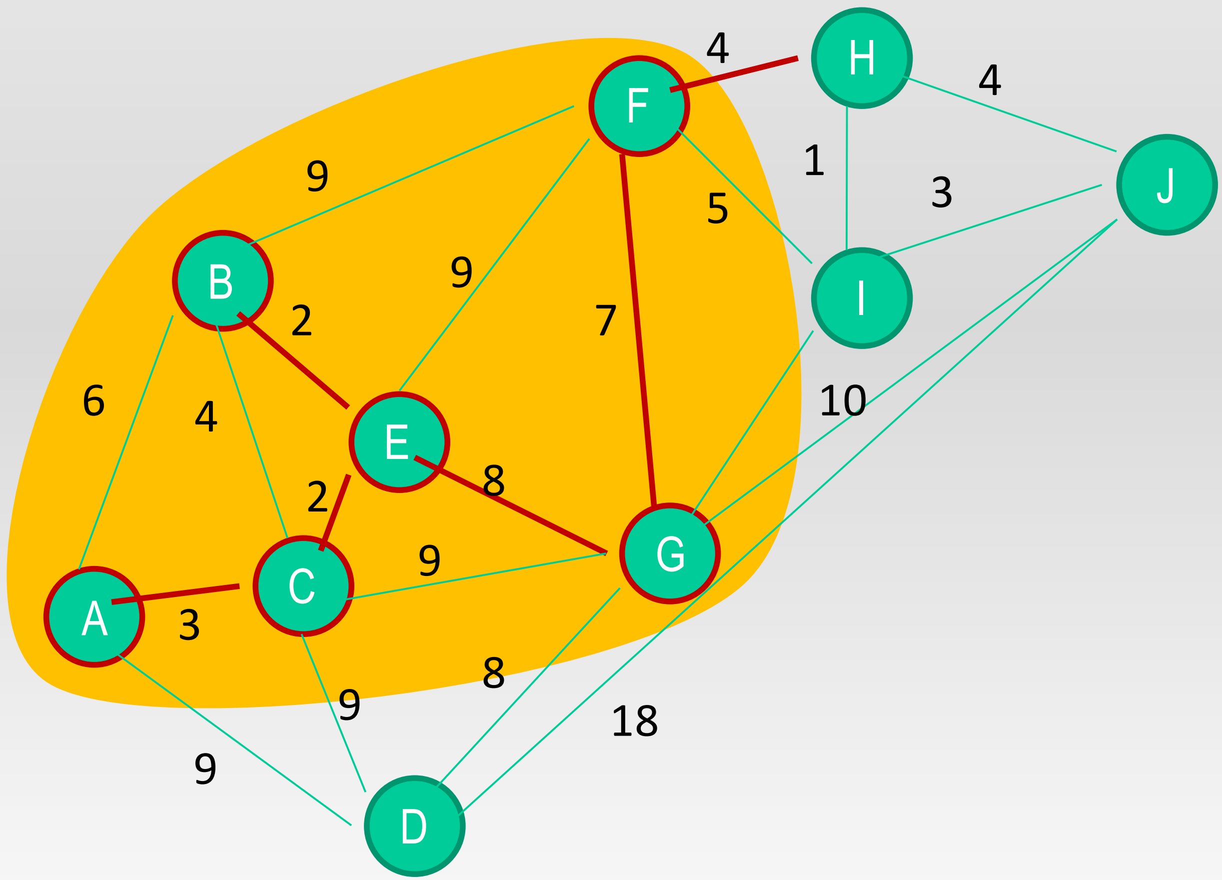
Solution (cont.)



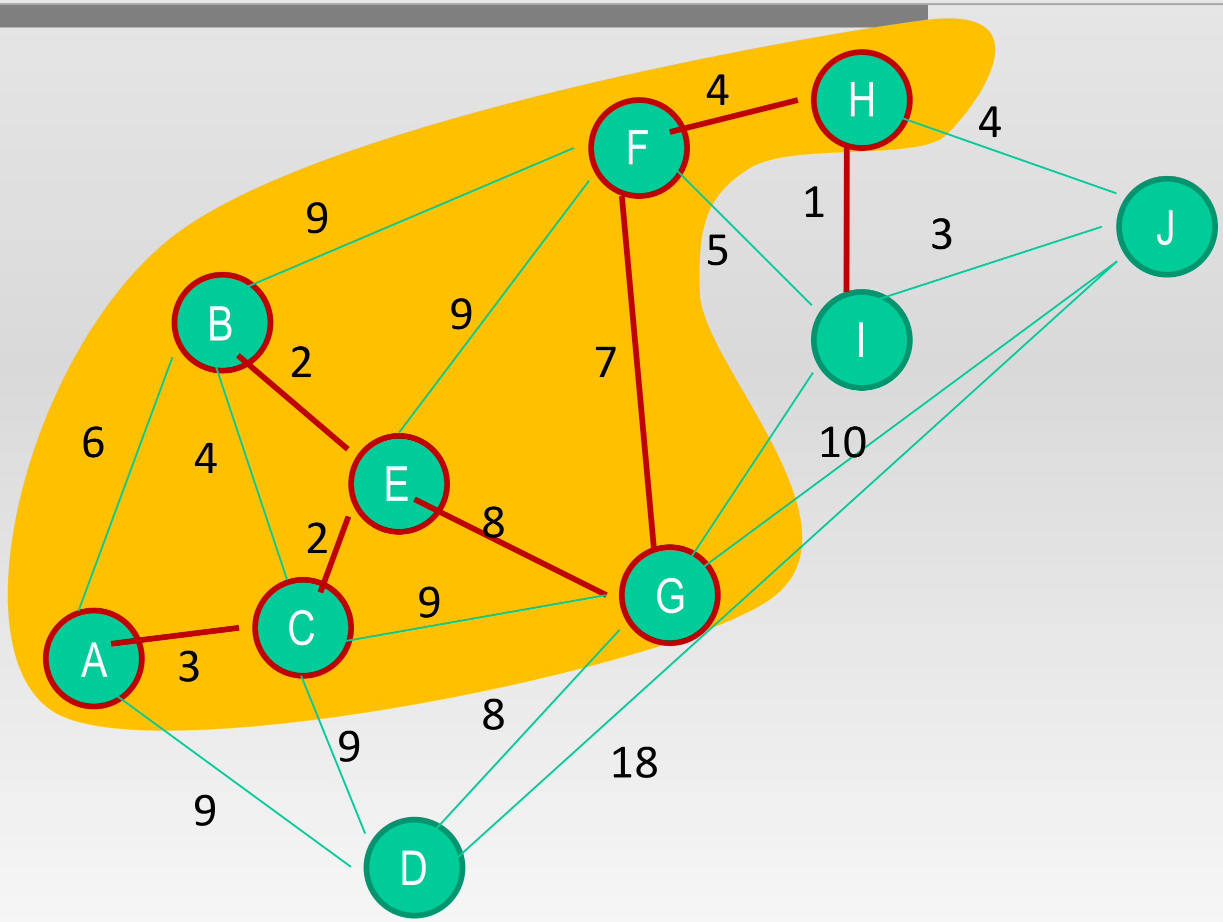
Solution (cont.)



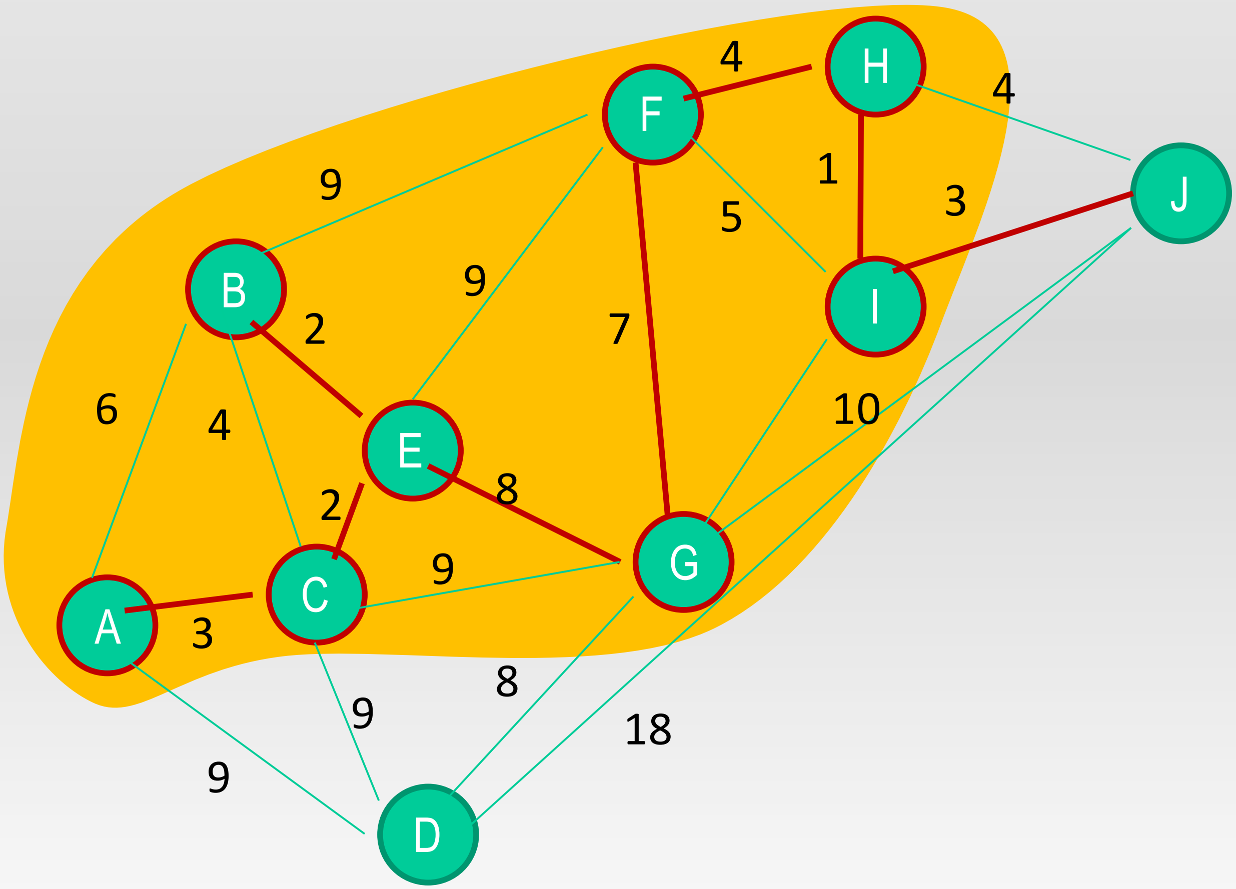
Solution (cont.)



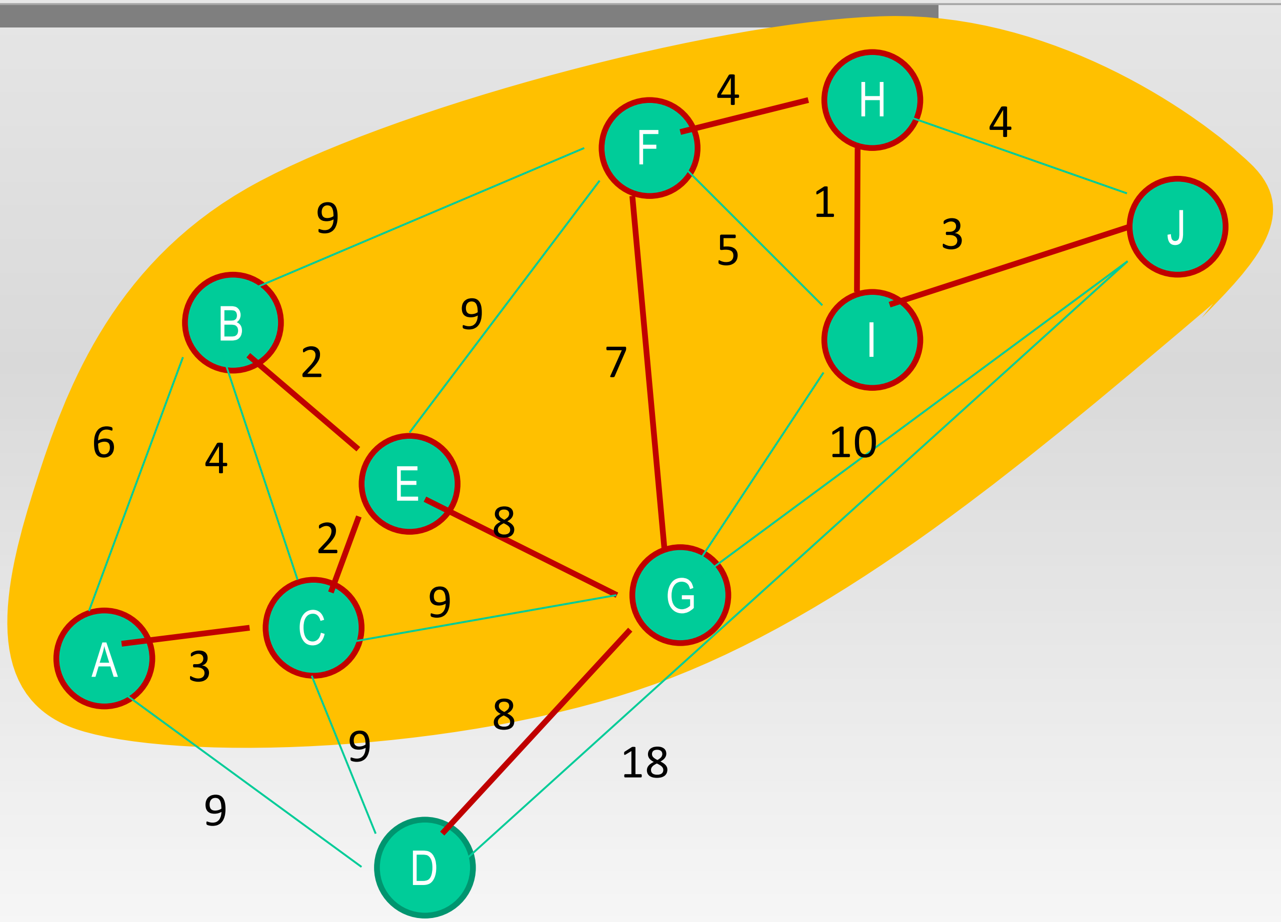
Solution (cont.)



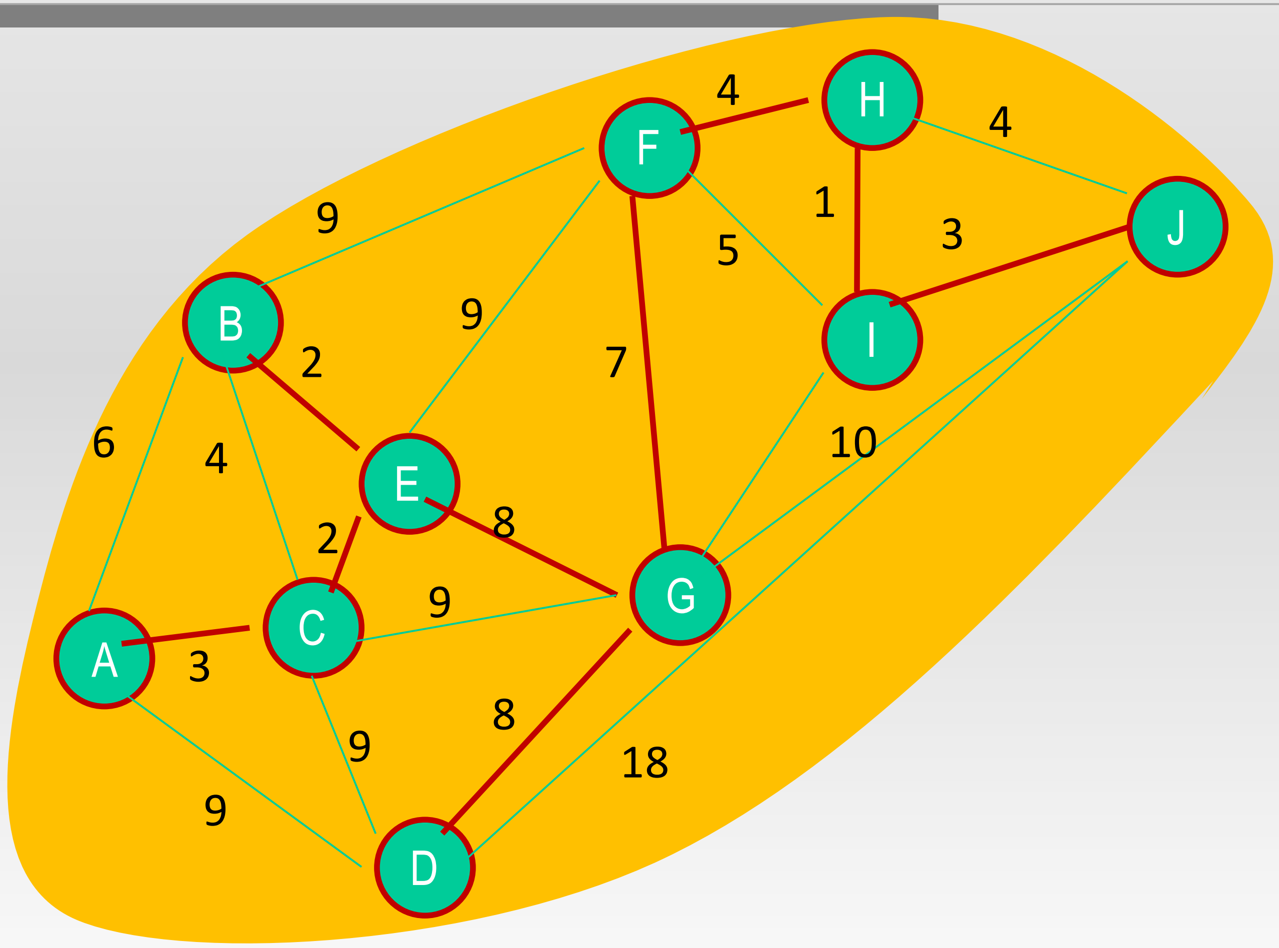
Solution (cont.)



Solution (cont.)



Solution (cont.)



Shortest- Route problem

- The shortest route problem determines the shortest route between a source and destination.
- There are two algorithms to solve shortest-route problems:
- 1- **Dijkstra's algorithm** that design to determine the shortest routes between the source node every other node in the network
- 2- **Floyd's algorithms** is general because it allow the determination of the shortest route between any two node in network

Dijkstra'a algorithm

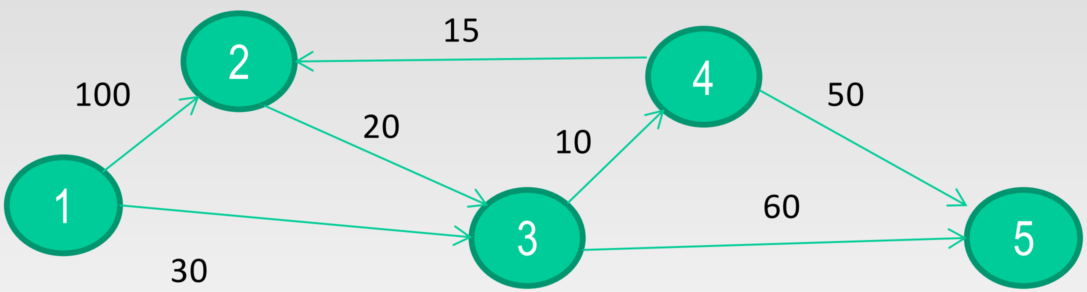
Step0: label the source node(node1) with the permanent label [0,--]. Set i=1

Stepi= (a) compute the temporary labels[ui+dij,i] for each node j that can be reached through node i. provided j is not permanently label. If node j is already label with [uj,k] through another node k and if ui+dij< uj, replace [uj,k] with [ui+ dij, i]

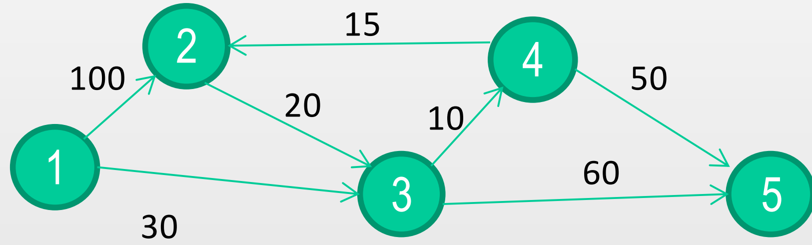
(b) if all node have premanent label stop. Otherwise select the label [Ur,s] having the shortest distance (=ui) among all temporary label. Set i=r and repeat step i

Example

- The figure give the route and their length in miles between city 1 and four other cities. Determine the shortest route between city 1 and each of the remaining four cities.



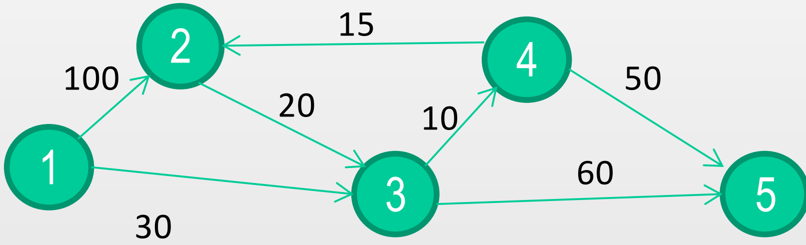
Example(cont.)



- Iteration 0: assign permanent label [0,--] to node 1
- Iteration 1: node 2 and 3 can be reached from (the last permanent labeled) node 1 thus the list labeled node (temporary and permanent) becomes

Node	label	status
1	[0,--]	permanent
2	[0+100, 1]	temporary
3	[0+30,1]	temporary

- For both two temporary label[100,1] and [30,1] **node 3** is smallest distance so, status of node 3 is changed to permanent

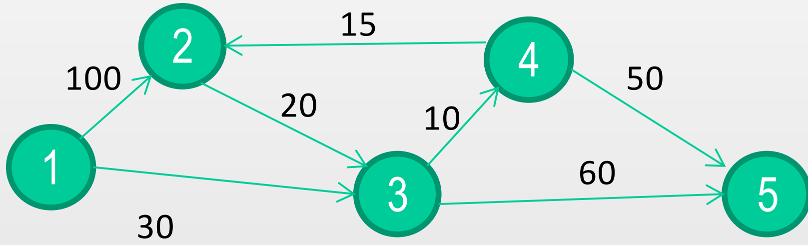


Example(cont.)

- Iteration2: node 4, and 5 can be reached from node 3 and the list labeled node becomes:

Node	label	status
1	[0,--]	permanent
2	[100, 1]	temporary
3	[30,1]	Permanent
4	[30+10,3]=[40,3]	temporary
5	[30+60,3]=[90,3]	temporary

- **node 4** is smallest distance so from the temporaries list. so, status of node 4 is changed to permanent



Example(cont.)

- Iteration 3: node 2 and 5 can be reached from node4. the list of labeled is updated as

Node	label	status
1	[0,--]	permanent
2	[40+12,4]=[55,4]	temporary
3	[30,1]	Permanent
4	[40,3]	Permanent
5	[90,3] or [40+50,4]	temporary

- **Node 2** is permanent

Example(cont.)

- Iteration 4: only node 3 can be reached from node 2, the node 3 is permanent , so the new list remain the same

Node	label	status
1	[0,--]	permanent
2	[55,4]	permanent
3	[30,1]	Permanent
4	[40,3]	Permanent
5	[90,3] or [40+50,4]	temporary

- Because node 5 is not lead to other node, it is status will convert to permanent

Example(cont.)

- The process ends

Node	label	status
1	[0,--]	permanent
2	[55,4]	permanent
3	[30,1]	Permanent
4	[40,3]	Permanent
5	[90,3] or [40+50,4]	Permanent

- The shortest route between node1 and node2 is:
- (2) [55,4]→(4) [40,3]→(3) [30,1]→(1)
- So the disired route is 1→3→4→2 with total length 55 miles

Floyd's algorithm

- Step0: define starting distance matrix D0 and node sequence matrix S0. the diagonal elements are marked with(-). Set k=1
- General step k: define row k and column as pivot row and pivot column. Apply the ***triple operation*** to each element dij in Dk-1. if the condition:

Dik+dkj<dij

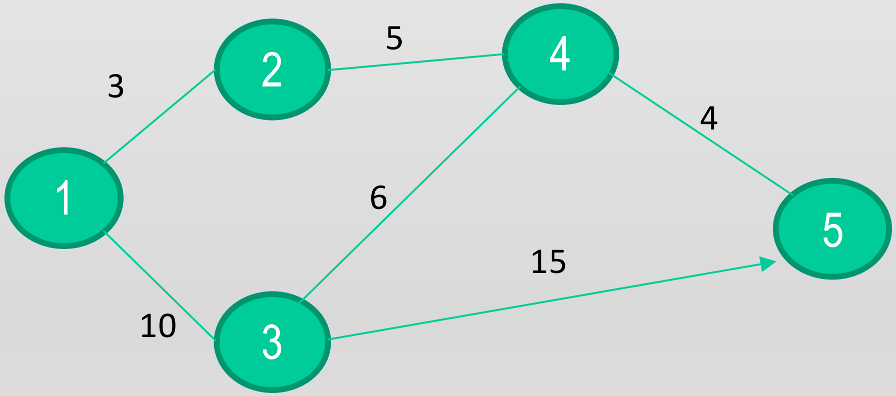
Is satisfied, make the following changes:

- (a) creat Dk by replacing dij in Dk-1 with dik+dkj
- (b) create Sk by repacing sij in sk-1 with k. set k=k+1 and repeat step k.

Floyd's algorithm (cont.)

- Step k : if the sum elements on the pivot row and povot coumn is smaller thanassociated intersection elements, the it is optimal to replace the intersection distance by the sum of pivot distance.
- After n step, it can determine the shortest route by using the following rules:
- 1- from D dij gives the shortest distance
- 2- from S determine the intermediate node.

Example



D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

s0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- **K=1**
- We highlight the **first** column and **first** row of the Distance matrix and compare all other items with the sum of the items highlighted in the same row and column.
- If the sum is less than the item then it should be replaced with the sum.

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When (-) is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	∞	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 10+3=13 is less than ∞

So change

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 10+3=13 is less than ∞

So change

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	∞	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+10=13 less than ∞,

So change

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When (-) is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞ is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When (-) is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞ is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D0

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S0

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	3	4	5
3	1	2	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- We have now completed one iteration. We rename the new matrices:

D1

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- Set k=2
- We highlight the **second** column and **second** row of the Distance matrix and compare all other items with the sum of the items highlighted in the same row and column.
- If the sum is less than the item then it should be replaced with the sum.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When (-) is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+13=16 Not Less than 10

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	∞	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+5=8 less than ∞

So change

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+5=8 less than ∞

So change

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞ is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+13=16 Not less than 10

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When (-) is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 5+13=18 Not less than 6

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	∞	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+5=8 less than ∞

So change

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 3+5=8 less than ∞

So change

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- 13+5=18 Not less than 6

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When (-) is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞ is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	4	5
2	1	-	1	4	5
3	1	1	-	4	5
4	1	2	3	-	5
5	1	2	3	4	-

- When ∞is involved we leave the item.

Example (cont.)

D1

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S1

	1	2	3	4	5
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- We have now completed two iteration. We rename the new matrices:

D2

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S2

	1	2	3	4	5
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- Set k=3
- We highlight the **third** column and **third** row of the Distance matrix and compare all other items with the sum of the items highlighted in the same row and column.
- If the sum is less than the item then it should be replaced with the sum.

Example (cont.)

D2

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S2

	1	2	3	4	5
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

D2

	1	2	3	4	5
1	-	3	10	8	∞
2	3	-	13	5	∞
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S2

	1	2	3	4	5
1	-	2	3	2	5
2	1	-	1	4	5
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

D2

	1	2	3	4	5
1	-	3	10	8	25
2	3	-	13	5	28
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S2

	1	2	3	4	5
1	-	2	3	2	3
2	1	-	1	4	3
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- We have now completed third iteration. We rename the new matrices:

D3

	1	2	3	4	5
1	-	3	10	8	25
2	3	-	13	5	28
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S3

	1	2	3	4	5
1	-	2	3	2	3
2	1	-	1	4	3
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

- Set k=4
- We highlight the **fourth** column and **fourth**row of the Distance matrix and compare all other items with the sum of the items highlighted in the same row and column.
- If the sum is less than the item then it should be replaced with the sum.

Example (cont.)

D3

	1	2	3	4	5
1	-	3	10	8	25
2	3	-	13	5	28
3	10	13	-	6	15
4	8	5	6	-	4
5	∞	∞	∞	4	-

S3

	1	2	3	4	5
1	-	2	3	2	3
2	1	-	1	4	3
3	1	1	-	4	5
4	2	2	3	-	5
5	1	2	3	4	-

Example (cont.)

D3

	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

S3

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

Example (cont.)

- We have now completed fourth iteration. We rename the new matrices:

D4

	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

S4

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

Example (cont.)

- Set k=5
- We highlight the **fifth** column and **fifth** row of the Distance matrix and compare all other items with the sum of the items highlighted in the same row and column.
- If the sum is less than the item then it should be replaced with the sum.

Example (cont.)

D4

	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

S4

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

- No further improvement are possible in this iteration, D5,S5 are the same D4 and S4

Example (cont.)

D4

	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

S4

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

- Shortest distance is d15 =12
- Associated route: recall segment(I,j) if Sij=J is direct link otherwise they link through intermediate node.

Example (cont.)

	1	2	3	4	5
1	-	3	10	8	12
2	3	-	11	5	9
3	10	11	-	6	10
4	8	5	6	-	4
5	12	9	10	4	-

	1	2	3	4	5
1	-	2	3	2	4
2	1	-	4	4	4
3	1	4	-	4	4
4	2	2	3	-	5
5	4	4	4	4	-

- S15= 4 ≠ 5 so. The initial link is 1→4→5
- Now, s14=2. is not direct link and 1→4 must replaced with 1→2→4, so the road from 1 to 5 will be change to 1→2→4→5.
- Now s12=2, s24=4,s45=5. the route 1→2→4→5 need no further dissecting and the process end

Maximal flow algorithm

- In a maximal flow problem, we seek to find the maximum volume of ***flow*** from a source node to terminal sink node in a capacitated network.

- Maximum flow algorithm is straightforward.

How it works

- In maximum flow algorithm, we determine if there is any path from source to sink that can carry flow.

- If there is , the flow is augmented as much as possible along this path; and residual capacities of the arc used on the path are reduced accordingly.

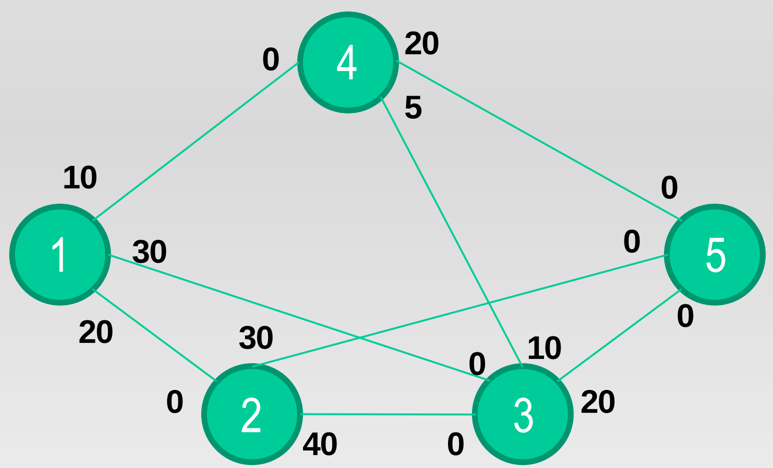
Steps of maximum flow algorithm

- *Step1*: find path from the source to the sink that has positive ***residual capacities***. If no path have positive, **STOP**; the maximum flow have been found
- *Step2*: Find the minimum ***residual capacity*** of the arc on the path (call it K) and augment the ***flow*** on each involved arc by K
- *Step3:* Adjust the ***residual capacities*** of arcs on the path by **decreasing** the ***residual capacities*** in direction of ***flow*** by **K**; and **increasing** the ***residual capacities*** in the direction opposite the ***flow*** by **K;**

GO TO STEP 1

Example

- Determine the maximum flow in the network.



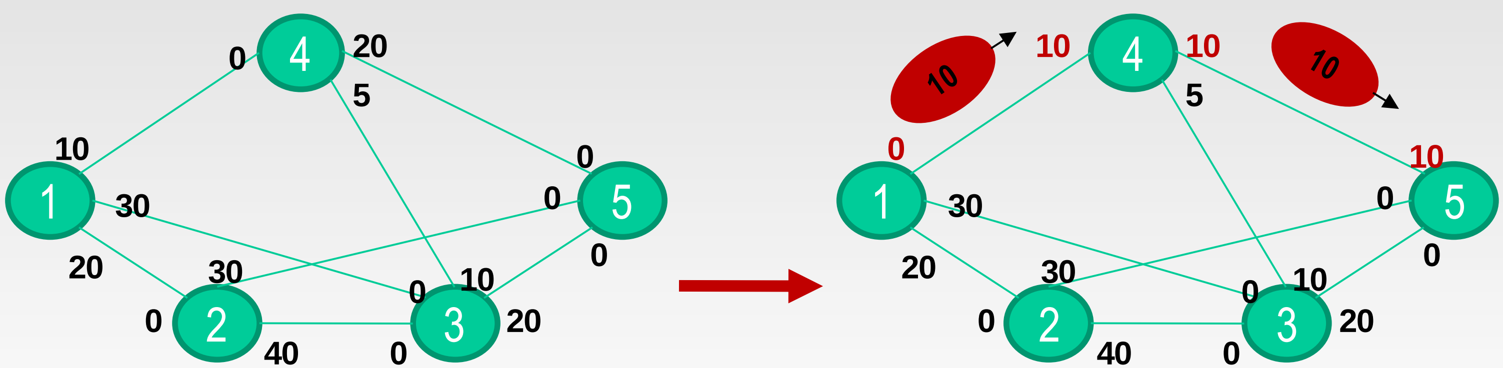
Example (cont.)

- Iteration1:

Select Path: 1→4→5

Residual capacities	
1-4	10
2-5	20

Augment **flow** by **10**
Reduce forward capacities by **10**
Increase backward capacities by **10**

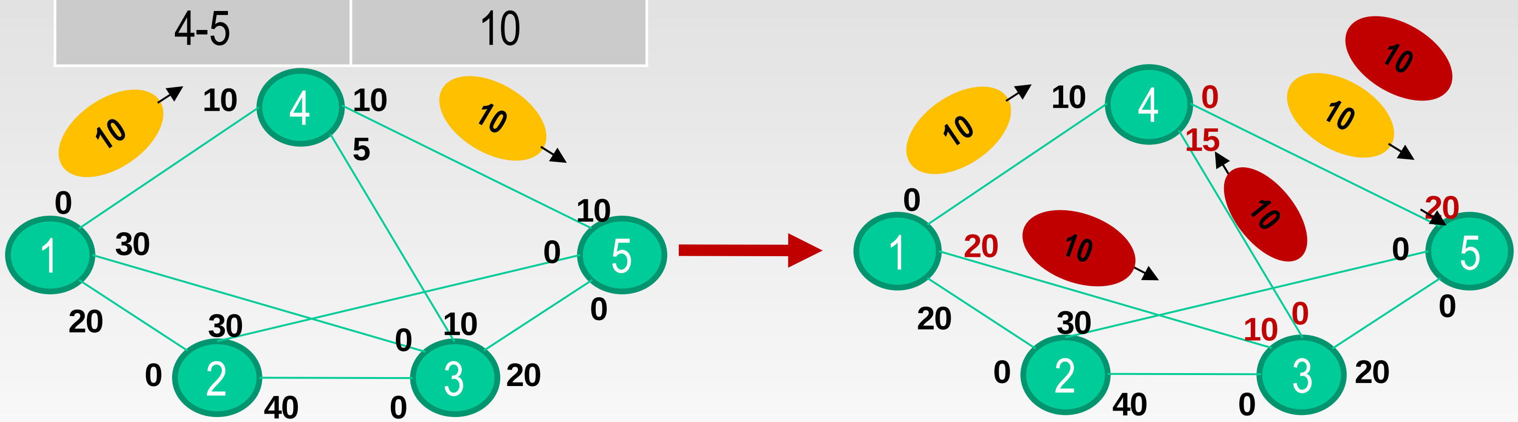


Example (cont.)

- <u>Iteration 2:</u>
- No additional possible flow along arc(1,4); thus find new path; Select path 1→3→4→5

Residual capacities	
1-3	30
3-4	10
4-5	10

Augment **flow** by **10**
Reduce forward capacities by **10**
Increase backward capacities by **10**

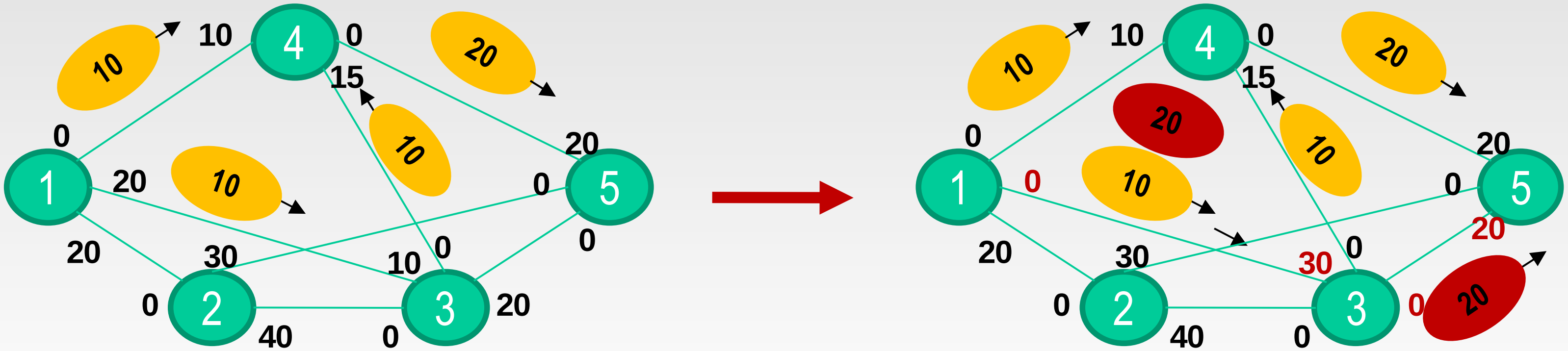


Example (cont.)

- <u>Iteration 3:</u>
- No additional possible flow along arc(3,4) and (4,5); thus find new path; Select path 1→3→5

Residual capacities	
1-3	20
3-5	20

Augment **flow** by **20**
Reduce forward capacities by **20**
Increase backward capacities by **20**

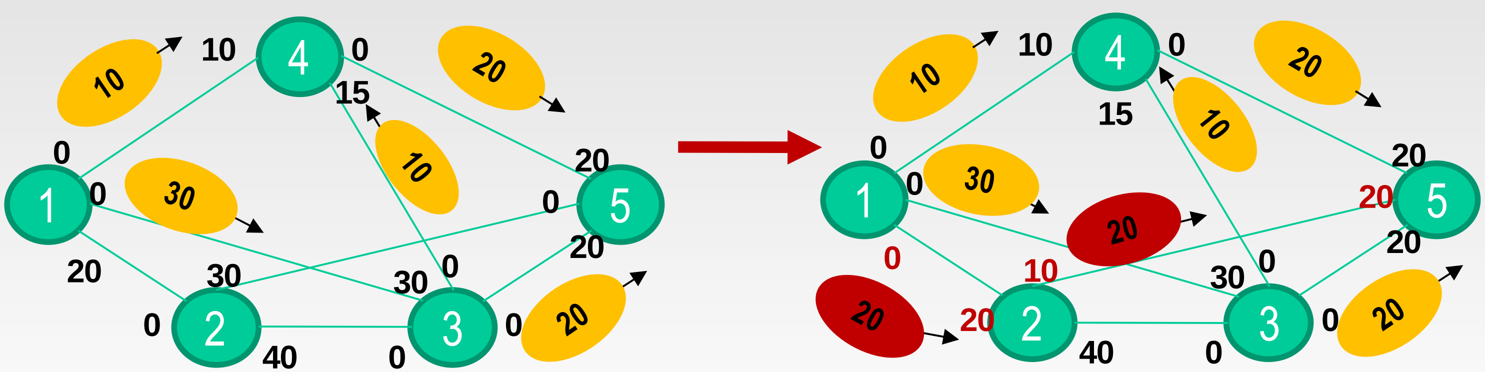


Example (cont.)

- <u>Iteration 4:</u>
- No additional possible flow along arc(1,3) and (3,5); thus find new path; Select path 1→2→5

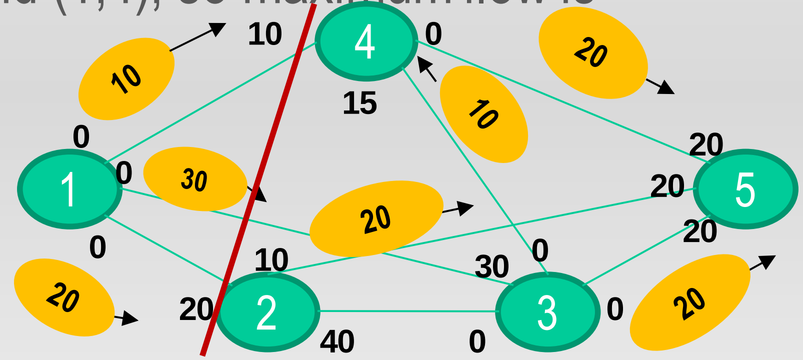
Residual capacities	
1-2	20
2-5	30

Augment **flow** by **20**
Reduce forward capacities by **20**
Increase backward capacities by **20**



Example (cont.)

- Iteration 4:
- No more flow is possible flow because there is no residual capacity left on the cut consisting (1,2),(1,3), and (1,4); so maximum flow is 20+30+10=60.



From	To	Flow
1	2	20
1	3	30
1	4	10
2	5	20
3	4	10
3	5	20
4	5	20