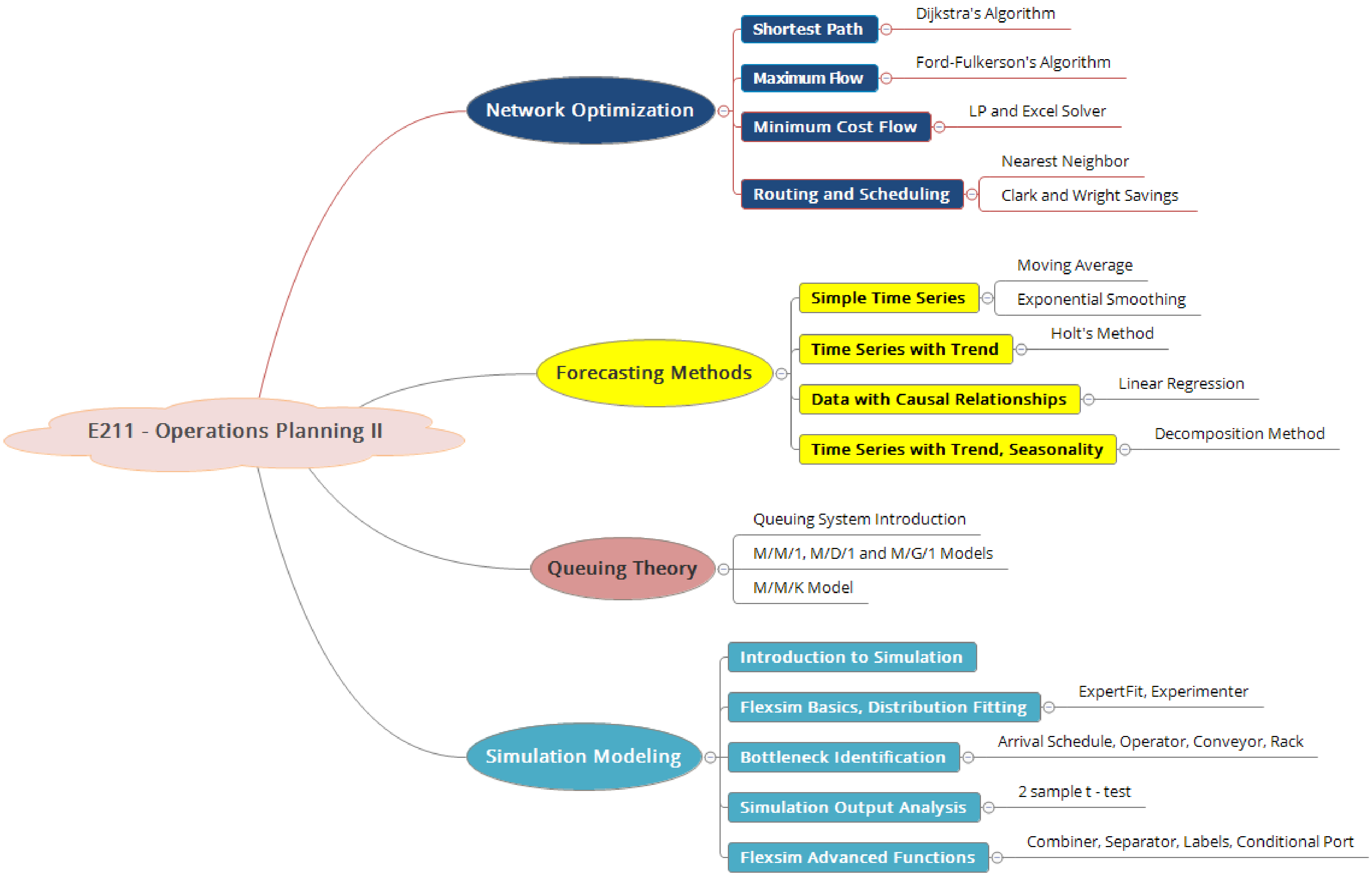
Problem 01

Shortest Path and Maximum Flow

E211 – Operations Planning II

SCHOOL OF ENGINEERING

Module Coverage: E211 Topic Tree



Shortest Path Problem

- The shortest path problem is to find the shortest distance / minimum cost / shortest time between an origin and various destination points.

- For example
  - ➢ Finding the best route to go from one town to another on a road map through a number of intermediate towns.
  - ➢ In this case, the nodes represent towns and the arcs represent segments of road and are weighted with the distance/time needed to travel that segment.

Dijkstra’s Algorithm for Shortest Path Problem

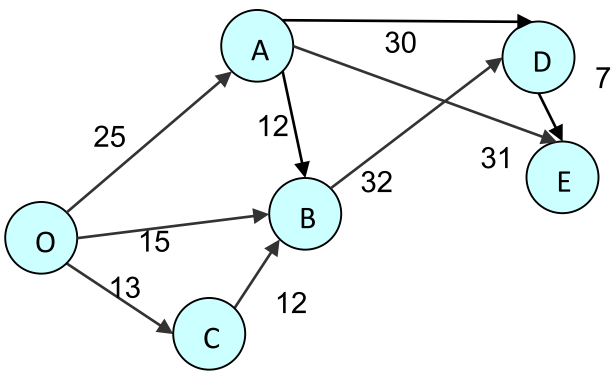
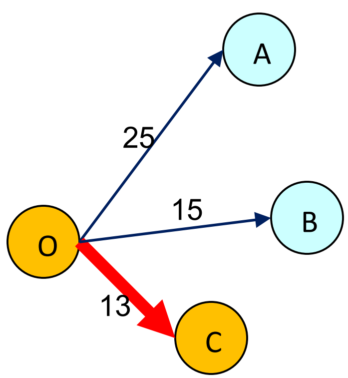
- Step 1 - Put the origin into the solution set

- Step 2 - Identify the arcs originating from the origin and put the terminating node of the shortest arc into the solution set

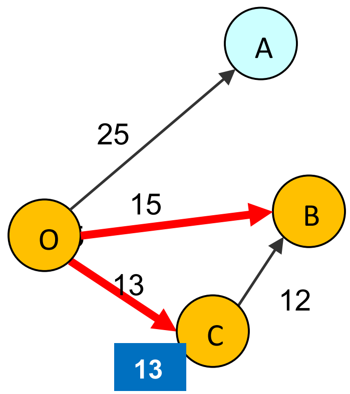
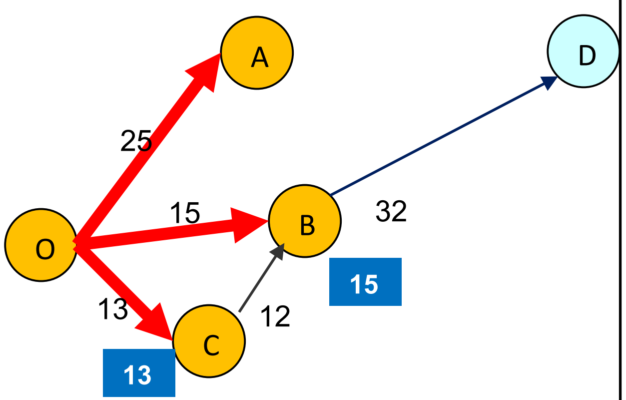
- Step 3 - Identify the arcs originating from the nodes in the solution set and select the node closest to the origin to join the solution set

- Step 4 - Repeat step 3 until all nodes have joined the solution set

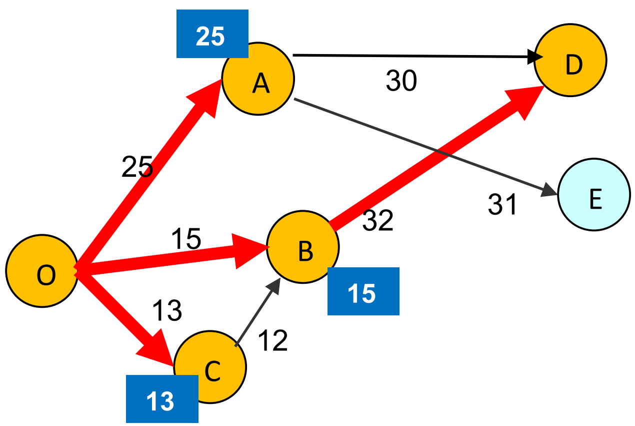
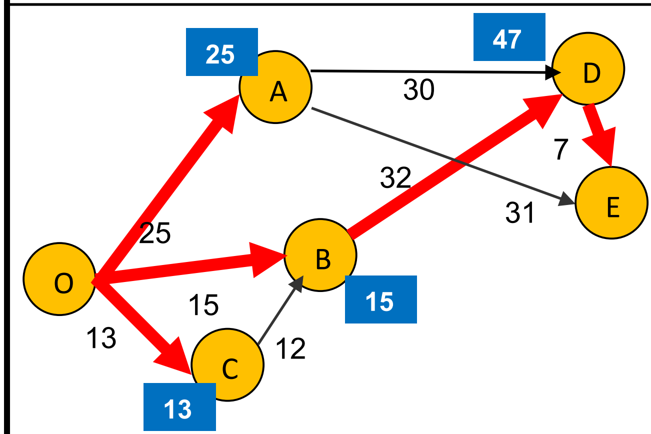
Dijkstra's Algorithm - Example

Network	Description	Solution Set	Arcs	Total Dist
	Problem: To find the shortest path from O to all other nodes. Numbers on arc can represent i.Distance between nodes ii.Time taken to travel between nodes iii.Cost to travel between nodes			
	Origin O is added into solution set. Nodes A, B and C are directly connected to O. Arc OC is the shortest arc from O, add node C into the solution set, shortest distance from O to C is 13.	O	O → A O → B O → C	25 15 13

Dijkstra's Algorithm - Example

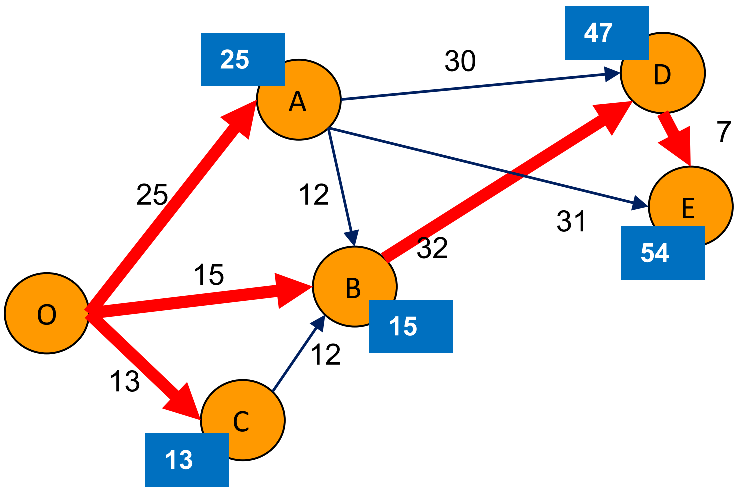
Network	Description	Solution Set	Arcs	Total Dist
	Nodes A and B are directly connected to O; node B is directly connected to C. Through arc CB, the total distance from O to B is 25 (= 13 + 12) Arc OB is the shortest arc from O, add node B into the solution set, shortest distance from O to B is 15.	O C (13)	O → A O → B C → B	25 15 25
	Node A is directly connected to O; node D is directly connected to B. Through arc BD, the total distance from O to D is 47 (= 15 + 32) Arc OA is the shortest arc from O, add node A into the solution set, shortest distance from O to A is 25.	O C (13) B (15)	O → A B → D	25 47

Dijkstra's Algorithm - Example

Network	Description	Solution Set	Arcs	Total Dist
	Nodes D and E are directly connected to A; node D is directly connected to B. The total distance from O to D through A is 55 (=25+30); from O to E through A is 56 (=25+31); from O to D through B is 47 (=15+32). Node D (having the shortest total distance of 47) is added to the solution set.	O C (13) B (15) A (25)	A → D A → E B → D	55 56 47
	Node E is directly connected to A; node E is directly connected to D. Node E (having the shortest total distance of 54 based on the route of O→ B→ D→ E) is lastly added to the solution set.	O C (13) B (15) A (25) D (47) E (54)	A → E D → E	56 54

Dijkstra's Algorithm - Example

Full solution is obtained when all nodes are added into the solution set. The optimal routes are shown by the bold red arcs .



From node O to	Route	Total Dist
Node A	O → A	25
Node B	O → B	15
Node C	O → C	13
Node D	O → B → D	47
Node E	O → B → D→ E	54

Variations of Shortest Path Problem

- Other variations of shortest path problem

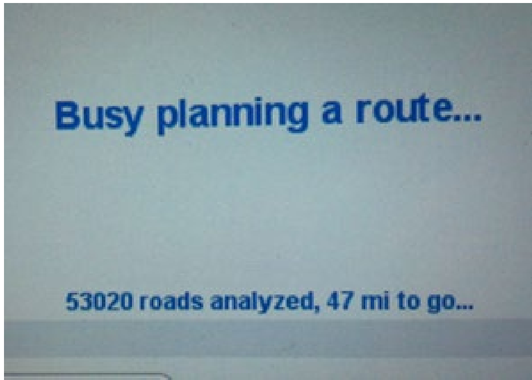
  - ➢ To find the shortest path (the path with the minimum distance) from the origin to one destination only.

  - ➢ To find the shortest paths from every node to every other node in the network.

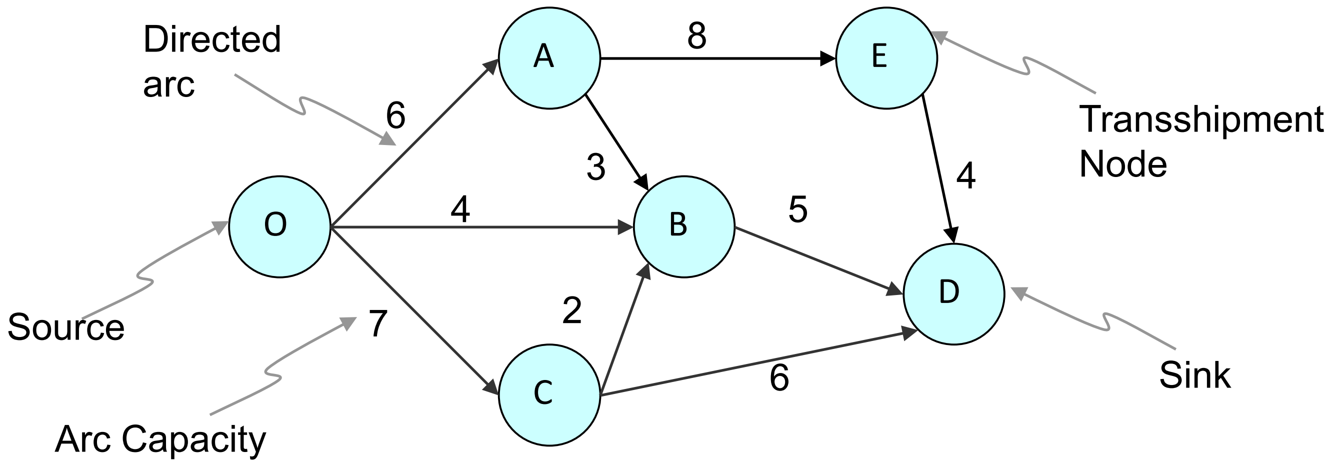
Maximum Flow Problem

- The maximum flow problem is to maximize the amount of flow of items from an origin to a destination when the arcs of the network have limited flow capacities.

- Examples of Maximum Flow Problems include flow of water, gas, or oil through a network of pipelines; the flow of traffic through a road network or the flow of products through a production line system.



Maximum Flow Problem



- All flow through a directed and connected network originates at one node, called the source, and terminates at one other node, called the sink.
- All the remaining nodes are transshipment nodes
- Flow through an arc is allowed only in the direction indicated by the arrowhead, where the maximum amount of flow is given by the capacity of that arc.

Ford-Fulkerson's Algorithm for Maximum Flow Problem

- Step 1 – Change each directed arc in the original network into an undirected arc. Arc capacity in the original direction remains the same; arc capacity in the opposite direction is set to be zero.

- Step 2 - Choose one path through the network from origin to destination.

- Step 3 - Determine the maximum flow that can pass through the path (maximum flow is the **minimum** of all the flow capacities of the arcs along the path)

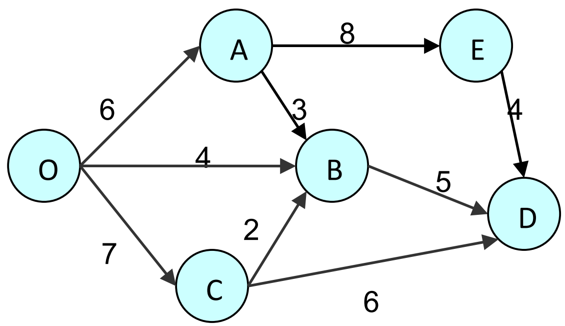
- Step 4 - Adjust the capacities of the arcs along the path by subtracting the maximum flow determined in step 3

- Step 5 - Adjust the capacities of the arcs in the opposite direction by adding the maximum flow

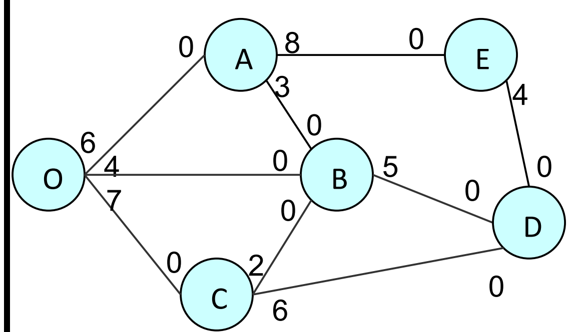
- Step 6 - Repeat steps 2 to 5 until there are no more paths with available flow

Ford-Fulkerson's Algorithm - Example

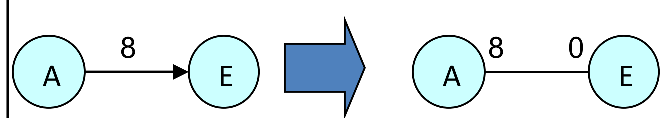
<table>
<tr><th>Network</th><th>Description</th></tr>
<tr><td>



</td><td>Problem: Find the maximum flow of items from the origin, O, to the destination, D, based on the limited capacities as shown by the number along the arcs.</td></tr>
<tr><td>

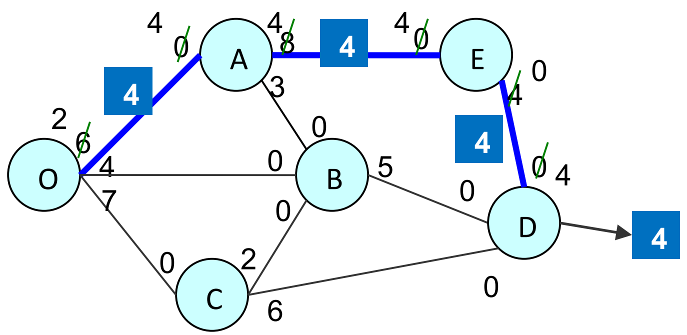
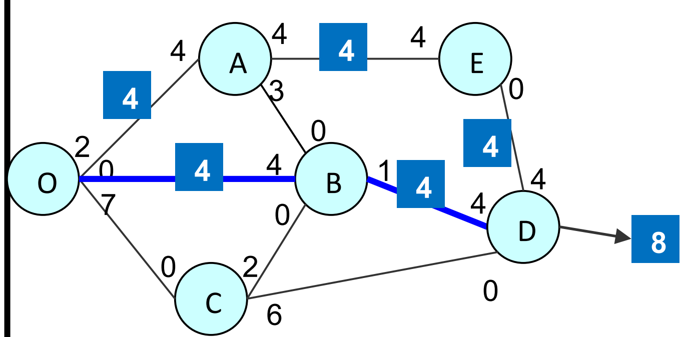


</td><td>Start by splitting the arc capacities for either direction of flow on the arc.

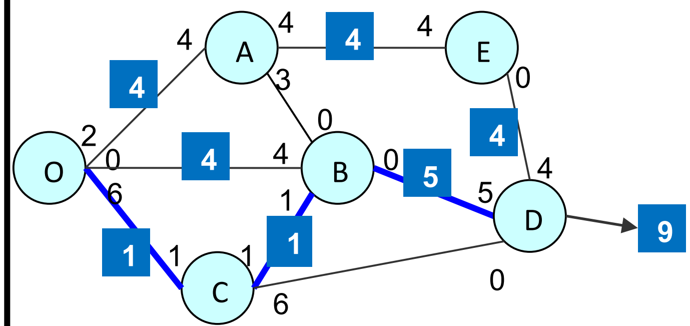
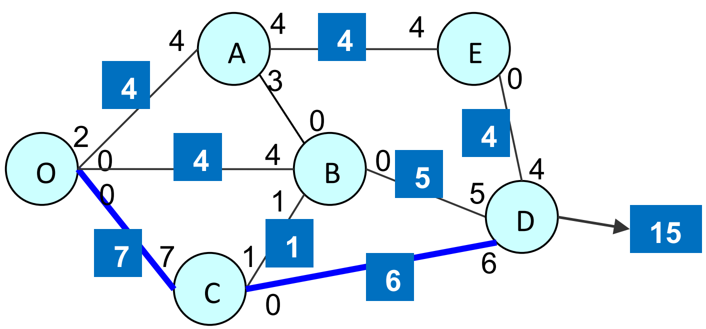


Max. flow from A to E is 8, and from E to A is 0</td></tr>
</table>

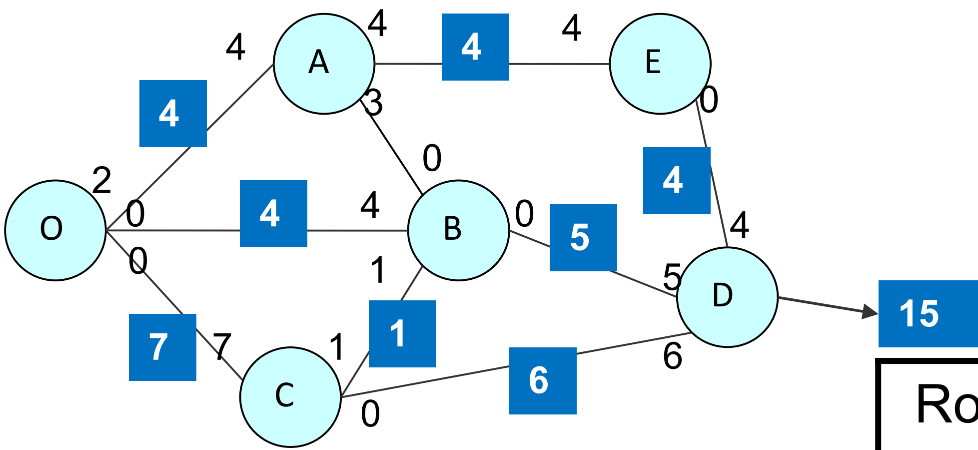
Ford-Fulkerson's Algorithm - Example

Network	Description
	- Path O → A → E → D is arbitrarily chosen. - Maximum flow that can pass through this path is 4, which is the min {6, 8, 4} - Subtract 4 from each of the arc capacities on path towards D - Add 4 to the arc capacities in the opposite direction
	- Choose another path, O → B → D - Maximum flow along this path is 4 - Adjust arc capacities accordingly - Total units that can be shipped through network is 8

Ford-Fulkerson's Algorithm - Example

<table>
<tr><th>Network</th><th>Description</th></tr>
<tr><td>
</td><td>- Choose another path, O → C → B → D
- Maximum flow along this path is 1
- Adjust arc capacities accordingly
- Total units that can be shipped through network is 9</td></tr>
<tr><td>
</td><td>- Choose another path, O → C → D
- Maximum flow along this path is 6
- Adjust arc capacities accordingly
- Total units that can be shipped through network is 15</td></tr>
</table>

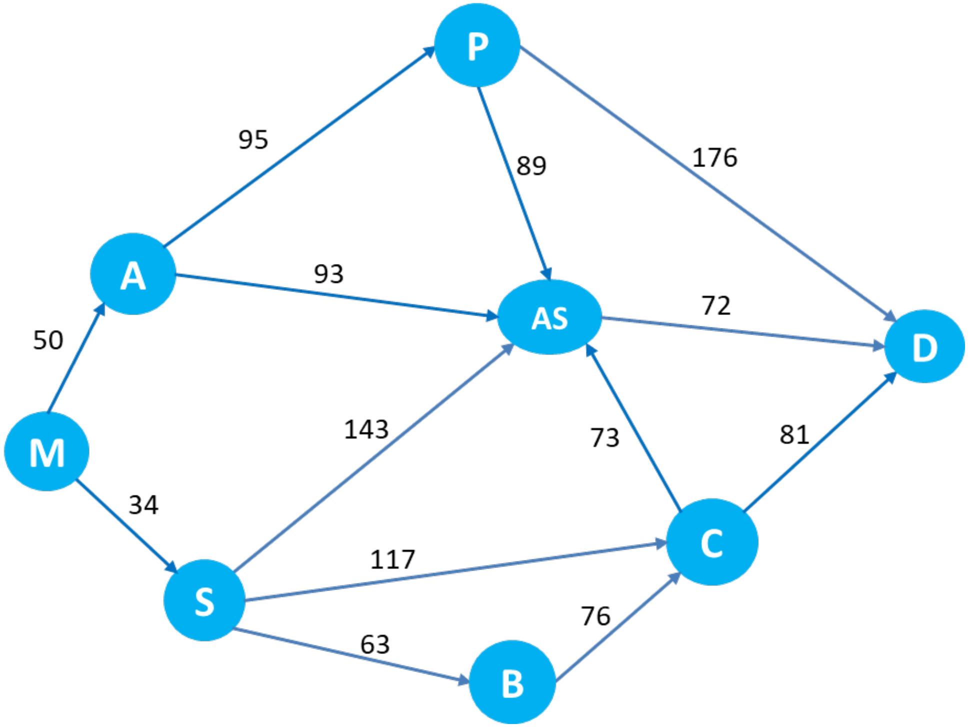
Ford-Fulkerson’s Algorithm - Example



Route	Flow Quantity
O → A → E → D	4
O → B → D	4
O → C → B → D	1
O → C → D	6
Total	15

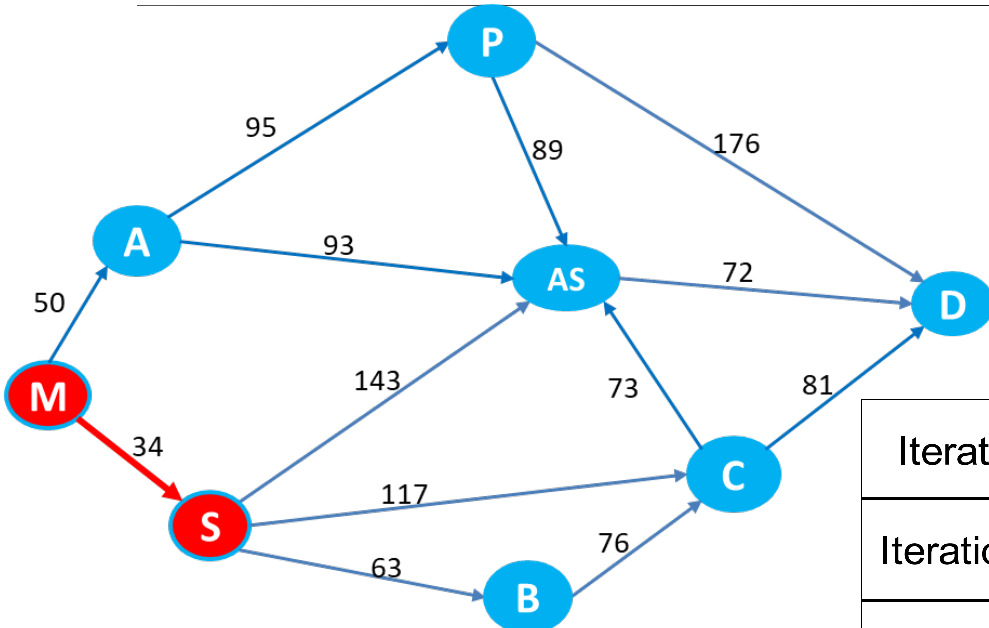
P01 Suggested Solution

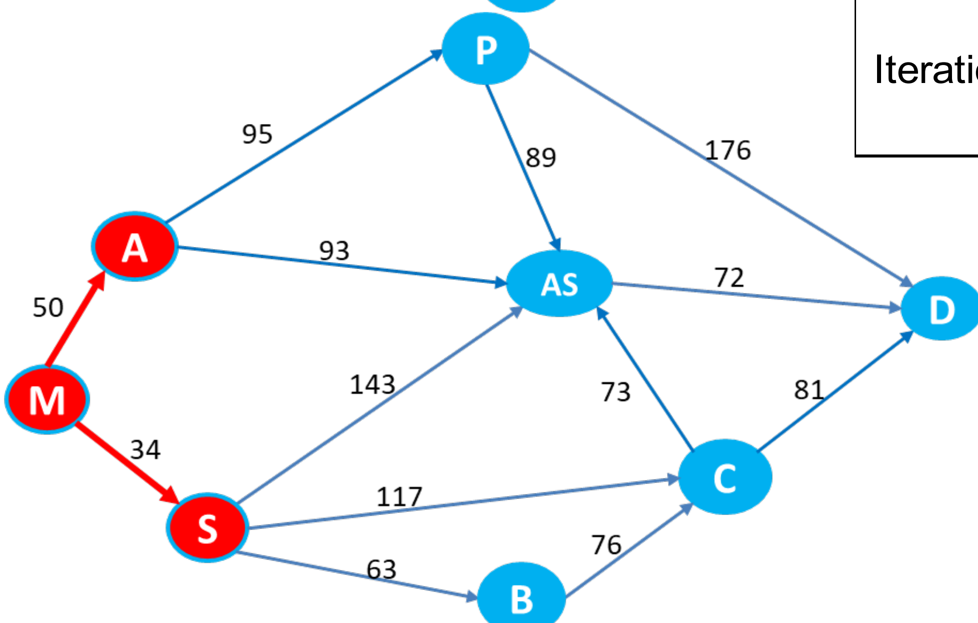
Problem Discussion



- The product delivery routes can be represented as a network model.
- The eight cities are represented by nodes.
- The delivery routes are represented by arcs.
- Minimum cost delivery route from Melbourne to each destination can be found by solving it as a shortest path problem using the Dijkstra's Algorithm.

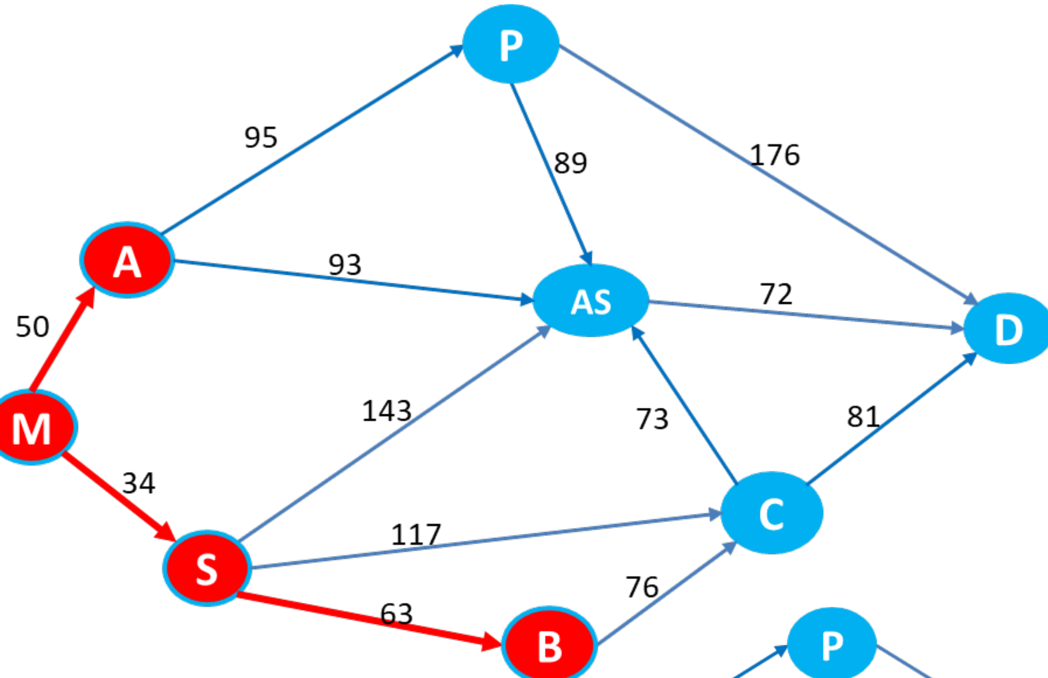
Today’s Problem: Shortest Path Problem



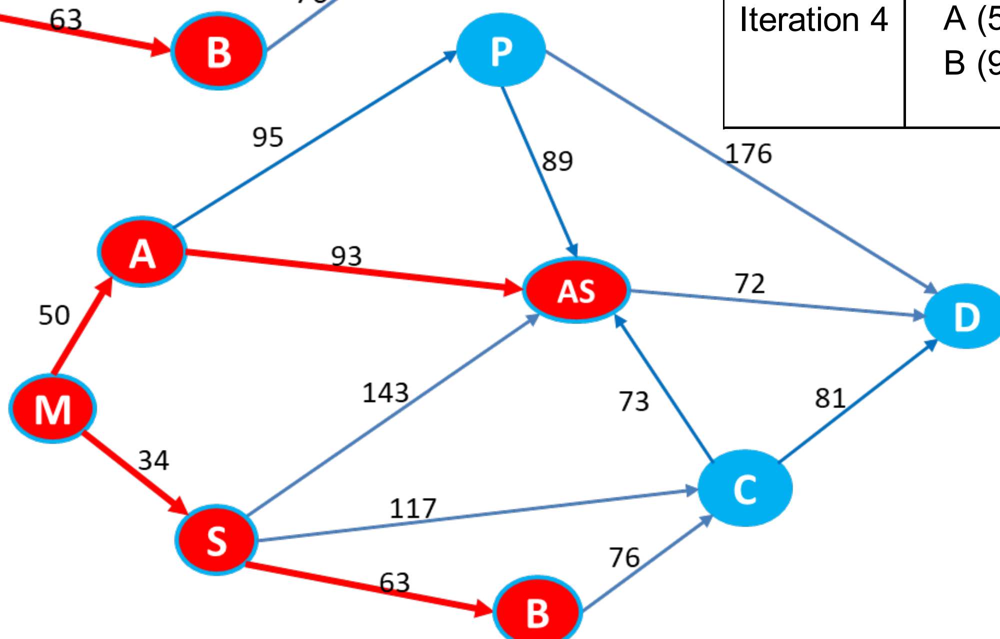


Iteration	Solution Set	Arcs	Total Cost
Iteration 1	M	M → A M → S	50 34
Iteration 2	M S(34)	M → A S → AS S → C S → B	50 34 + 143 = 177 34 + 117 = 151 34 + 63 = 97

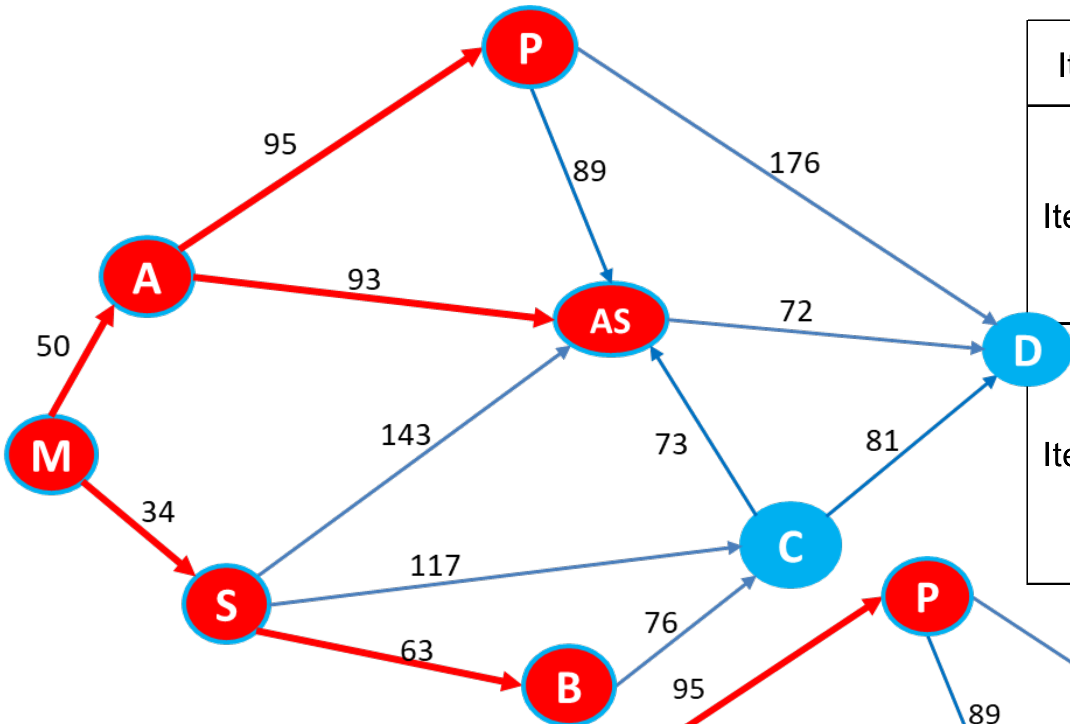
Today's Problem: Shortest Path Problem



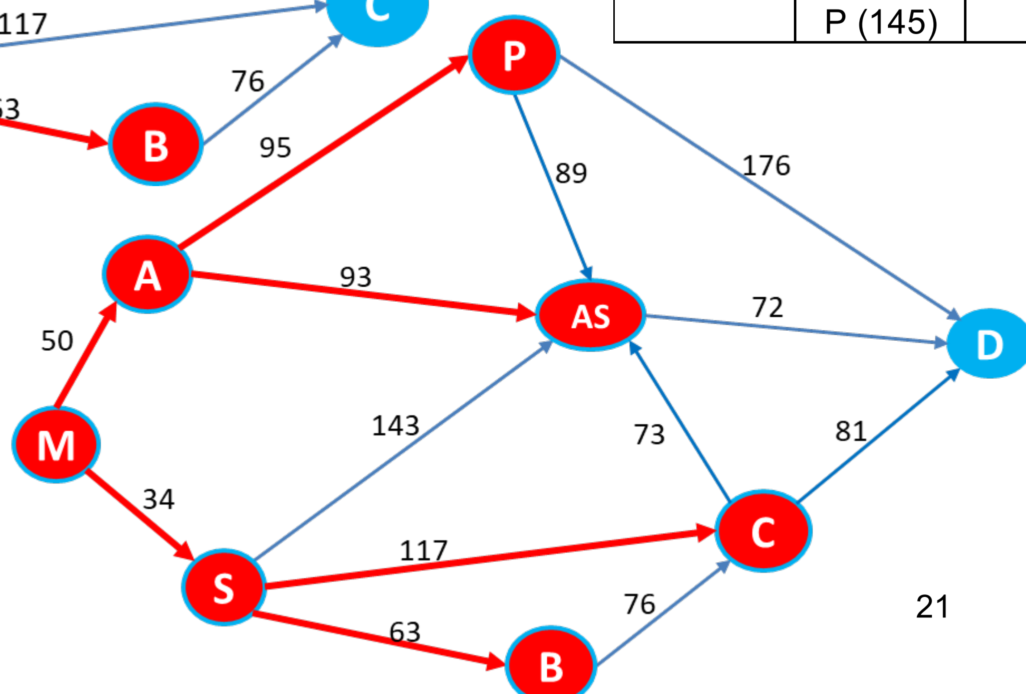
Iteration	Solution Set	Arcs	Total Cost
	M S(34) A (50)	S → AS S → C S → B A → P A → AS	34 + 143 = 177 34 + 117 = 151 34 + 63 = 97 50 + 95 = 145 50 + 93 = 143
Iteration 4	M S(34) A (50) B (97)	S → AS S → C A → P A → AS B → C	34 + 143 = 177 34 + 117 = 151 50 + 95 = 145 50 + 93 = 143 97 + 76 = 173



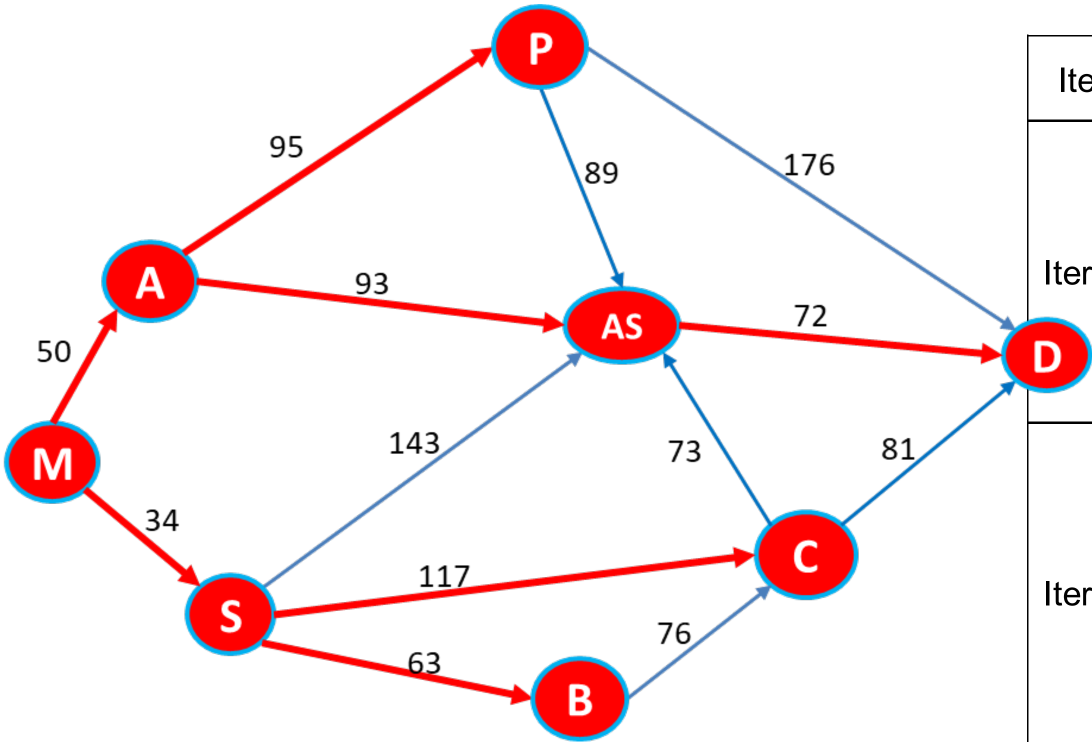
Today's Problem: Shortest Path Problem



Iteration	Solution Set	Arcs	Total Cost
Iteration 5	M S(34) A (50) B (97) AS(143)	S → C A → P B → C AS → D	34 + 117 = 151 50 + 95 = 145 97 + 76 = 173 143 + 72 = 215
Iteration 6	M S(34) A (50) B (97) AS(143) P (145)	S → C B → C AS → D P → D	34 + 117 = 151 97 + 76 = 173 143 + 72 = 215 145 + 176 = 321

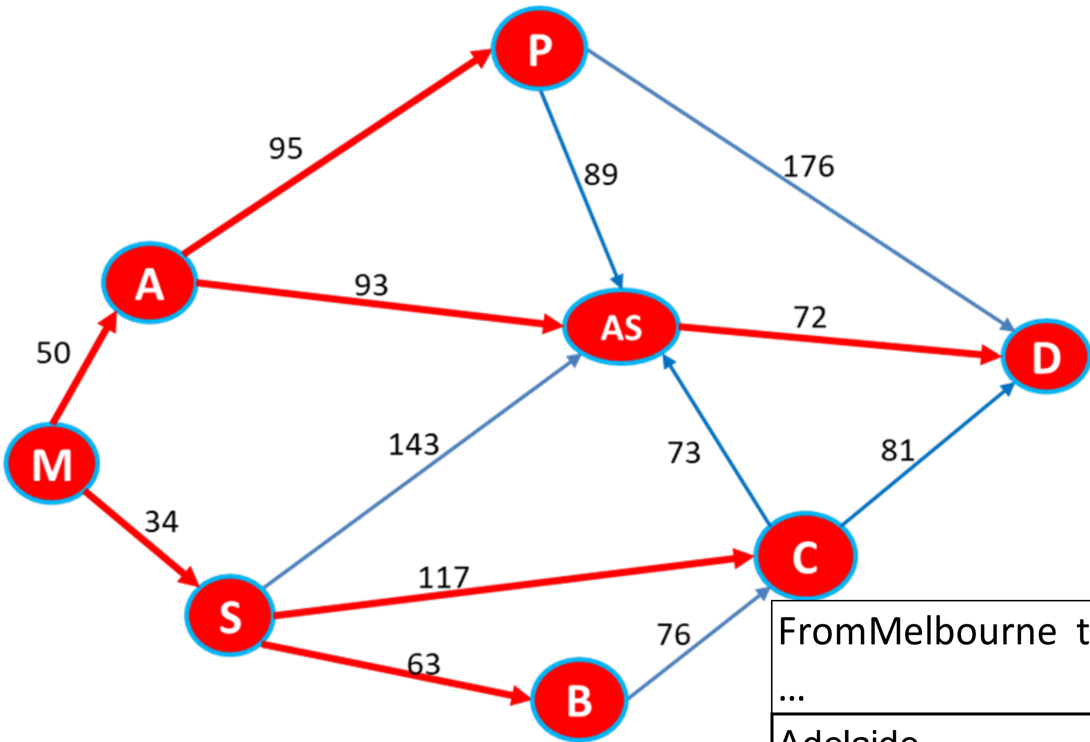


Today’s Problem: Shortest Path Problem



Iteration	Solution Set	Arcs	Total Cost
Iteration 7	M S(34) A (50) B (97) AS(143) P (145) C (151)	AS → D P → D C → D	143 + 72 = 215 145 + 176 = 321 151 + 81 = 232
Iteration 8	M S (34) A (50) B (97) AS (143) P (145) C (151) D (215)		

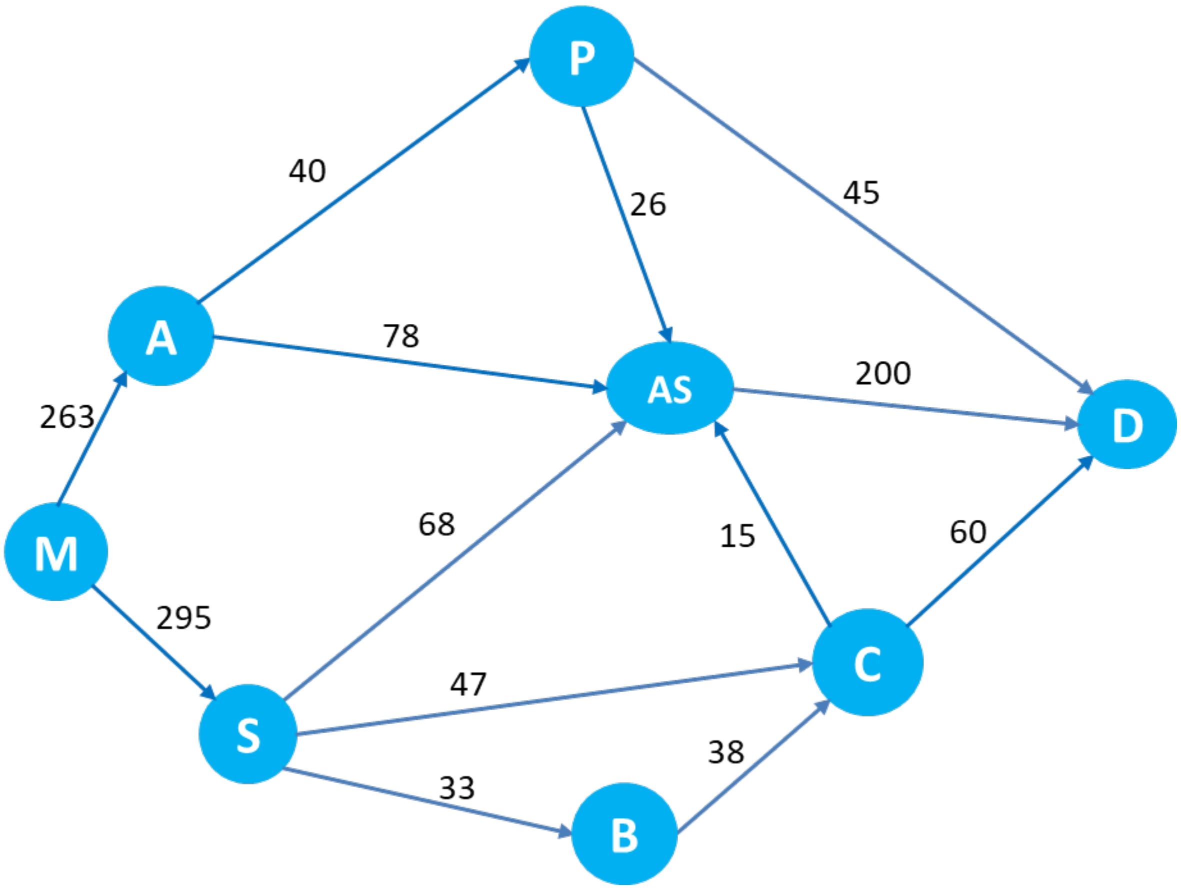
Today's Problem: Shortest Path Problem



The cheapest route from Melbourne to each of the other cities is shown by the bold red arcs.

FromMelbourne to …	Route	Total Cost
Adelaide	M → A	50
Sydney	M → S	34
Brisbane	M → S → B	97
Perth	M → A → P	145
Alice Spring	M → A → AS	143
Cairns	M → S → C	151
Darwin	M → A → AS →D	215

Today's Problem: Maximum Flow Problem



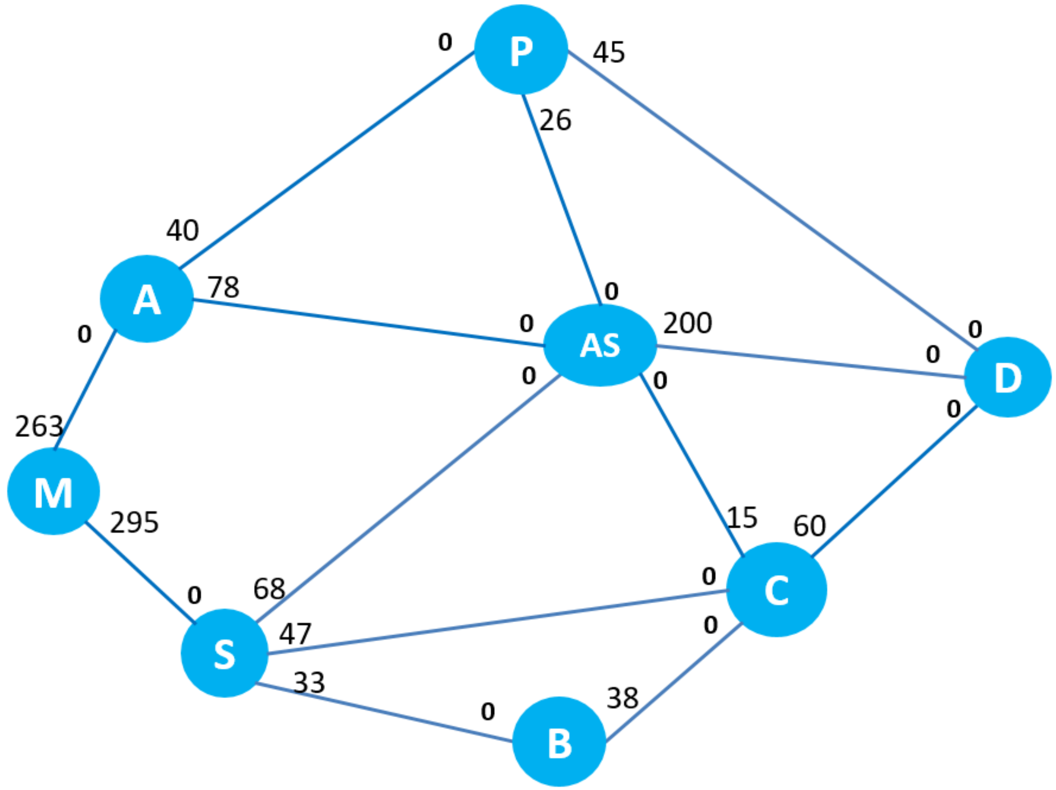
To determine the maximum flow of products (in number of containers) from Melbourne to Darwin and the delivery of products on each route, we can model it as a Maximum Flow Problem and solve it using the Ford-Fulkerson's Algorithm.

Assumptions:

- Cost of delivery on each route is not considered here.
- Products are only delivered from Melbourne to Darwin.

Today's Problem: Maximum Flow Problem

Solution Step 1

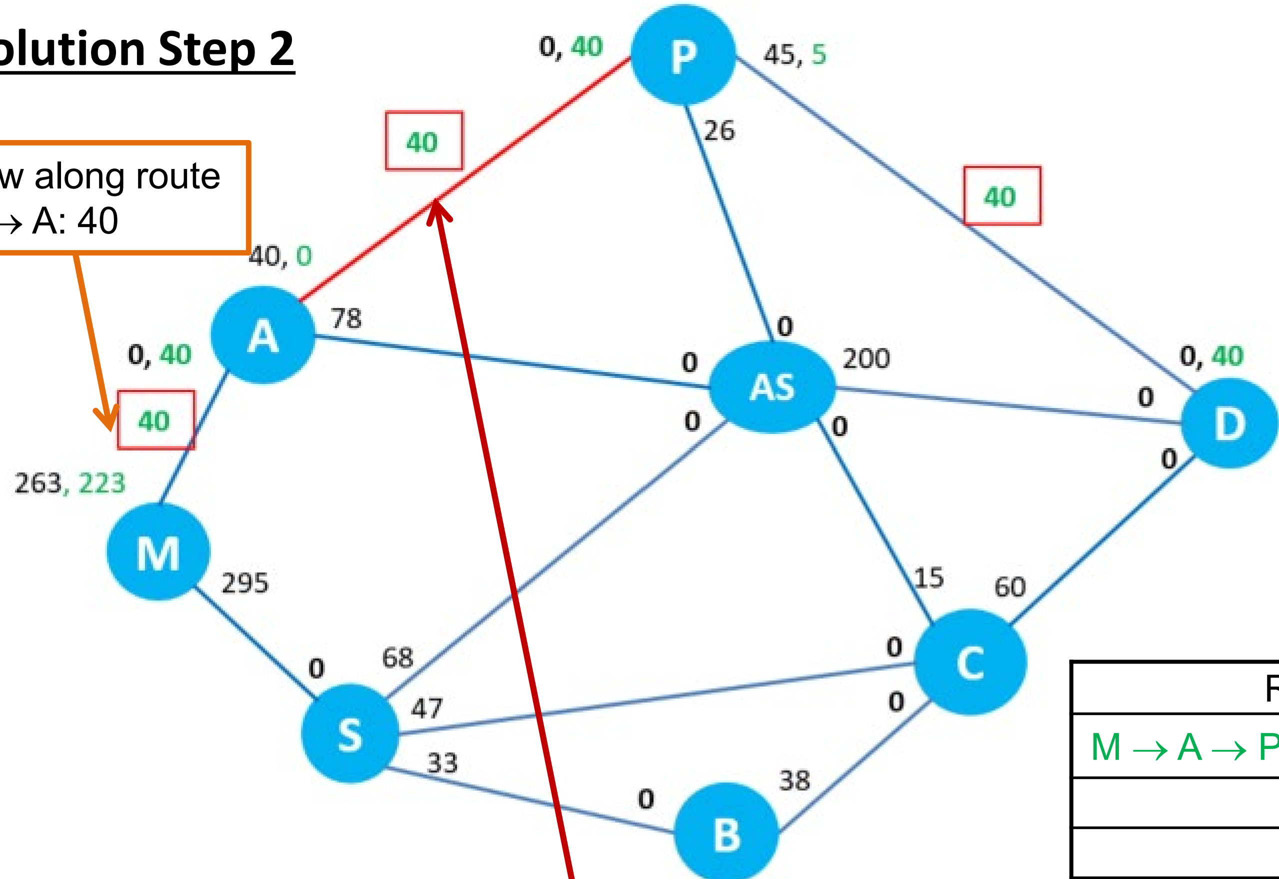


Route	Flow Quantity
Total	

Today's Problem: Maximum Flow Problem

Solution Step 2

Flow along route
M → A: 40



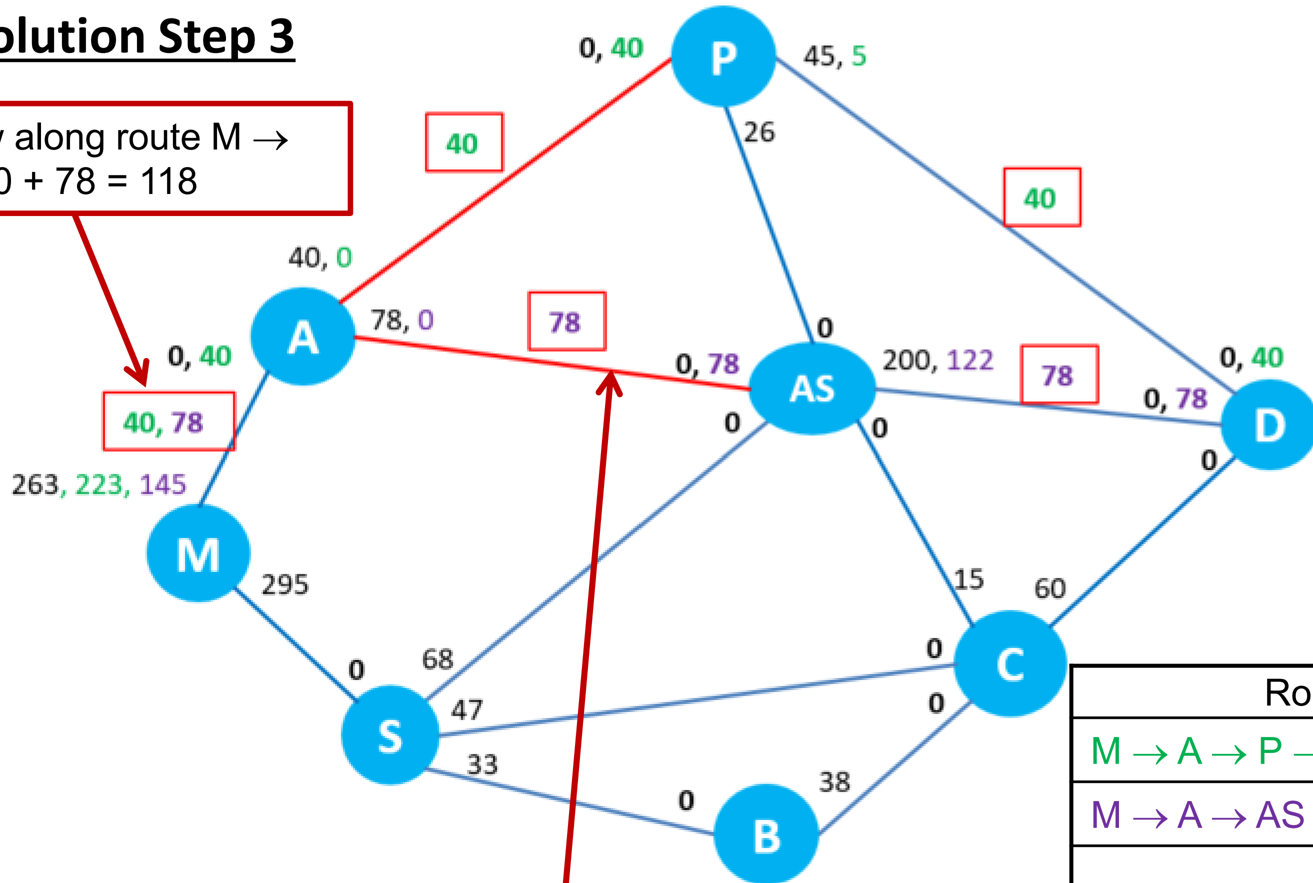
Arc in 'red' represents that flow capacity along route A → P is reached

Route	Flow Quantity
M → A → P → D	40
Total	

Today’s Problem: Maximum Flow Problem

Solution Step 3

Flow along route M → A: 40 + 78 = 118

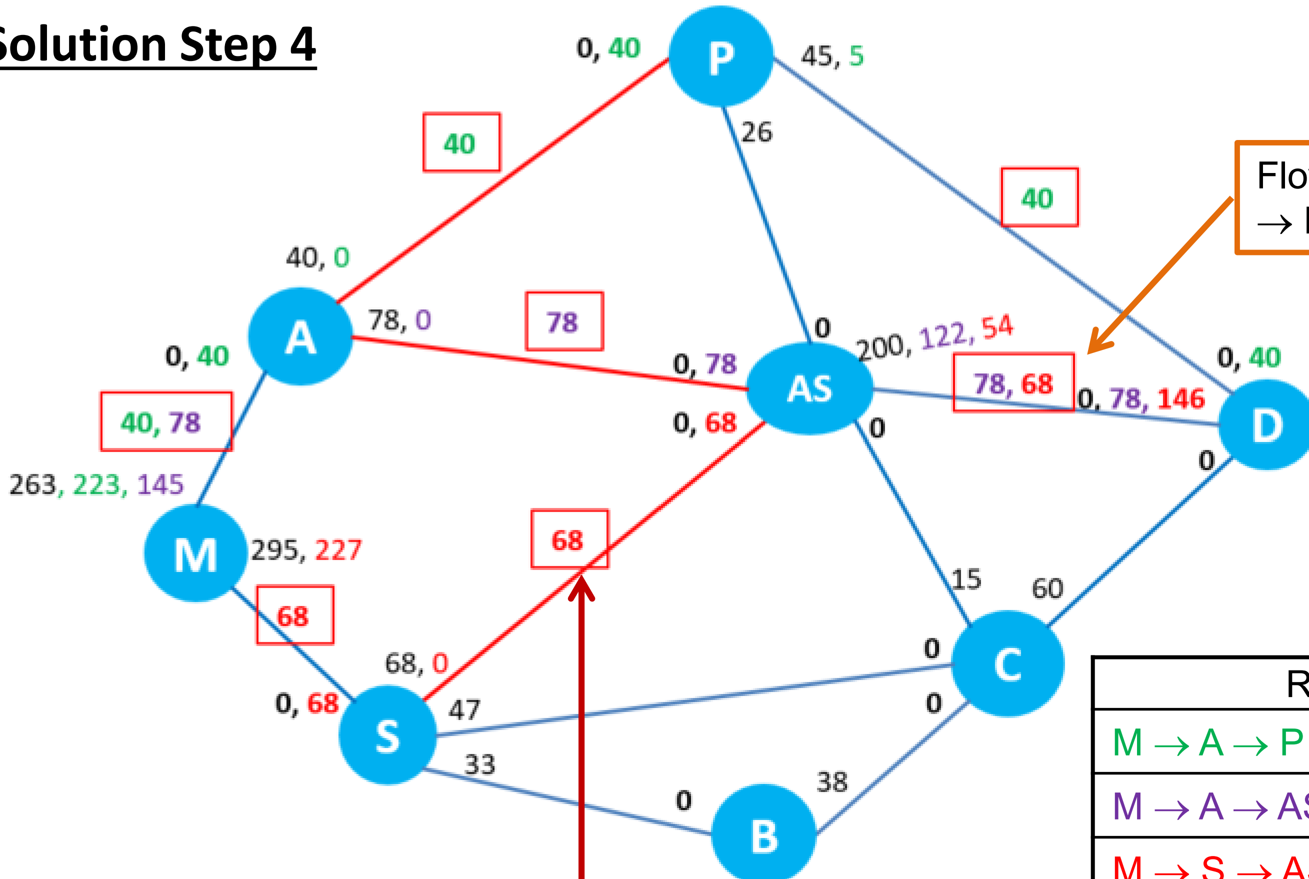


Arc in ‘red’ represents that flow capacity along route A→ AS is reached

Route	Flow Quantity
M → A → P → D	40
M → A → AS → D	78
Total	

Today’s Problem: Maximum Flow Problem

Solution Step 4



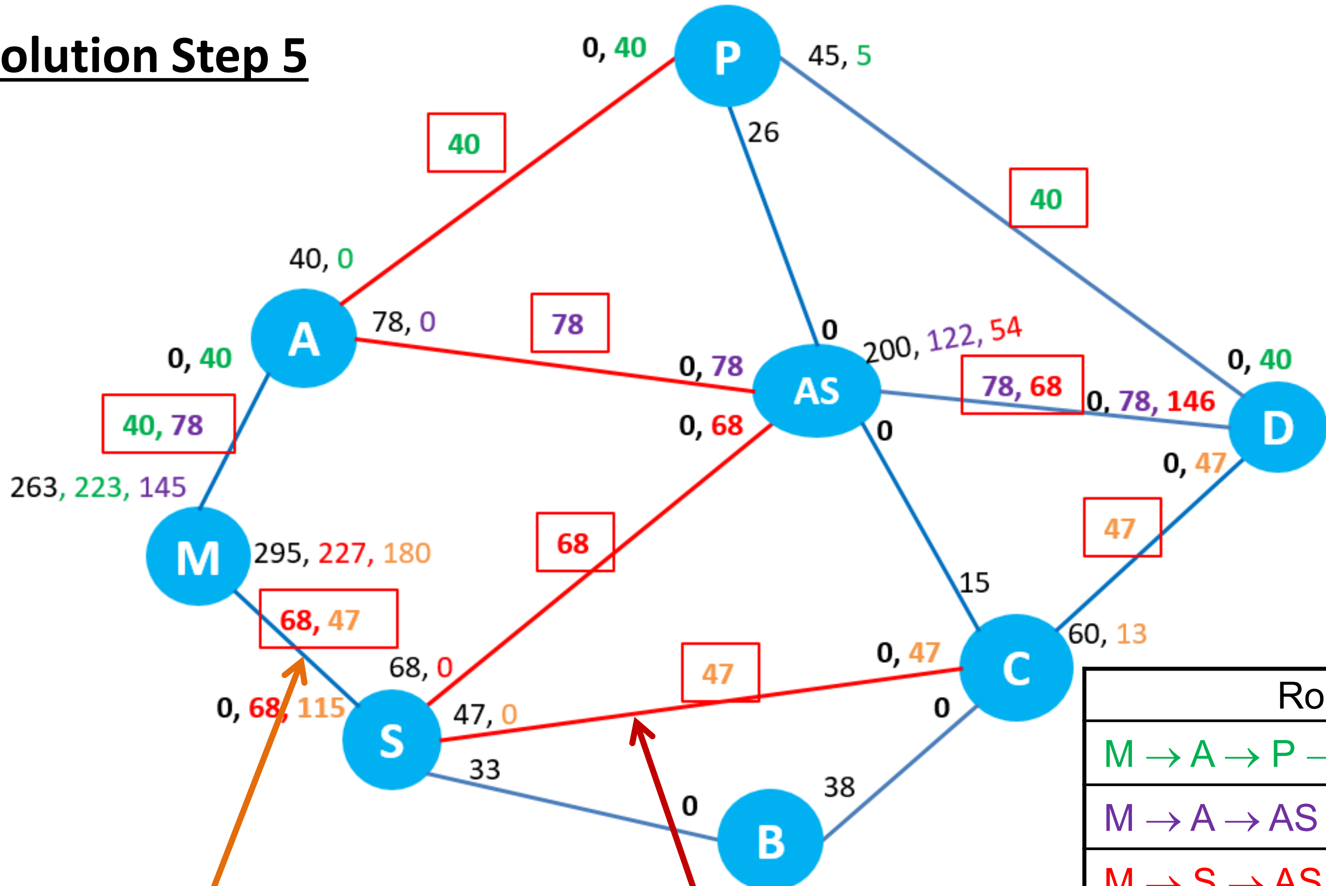
Flow along route AS → D: 78 + 68 = 146

Arc in ‘red’ represents that flow capacity along route S → AS is reached

Route	Flow Quantity
M → A → P → D	40
M → A → AS → D	78
M → S → AS → D	68
Total	

Today's Problem: Maximum Flow Problem

Solution Step 5



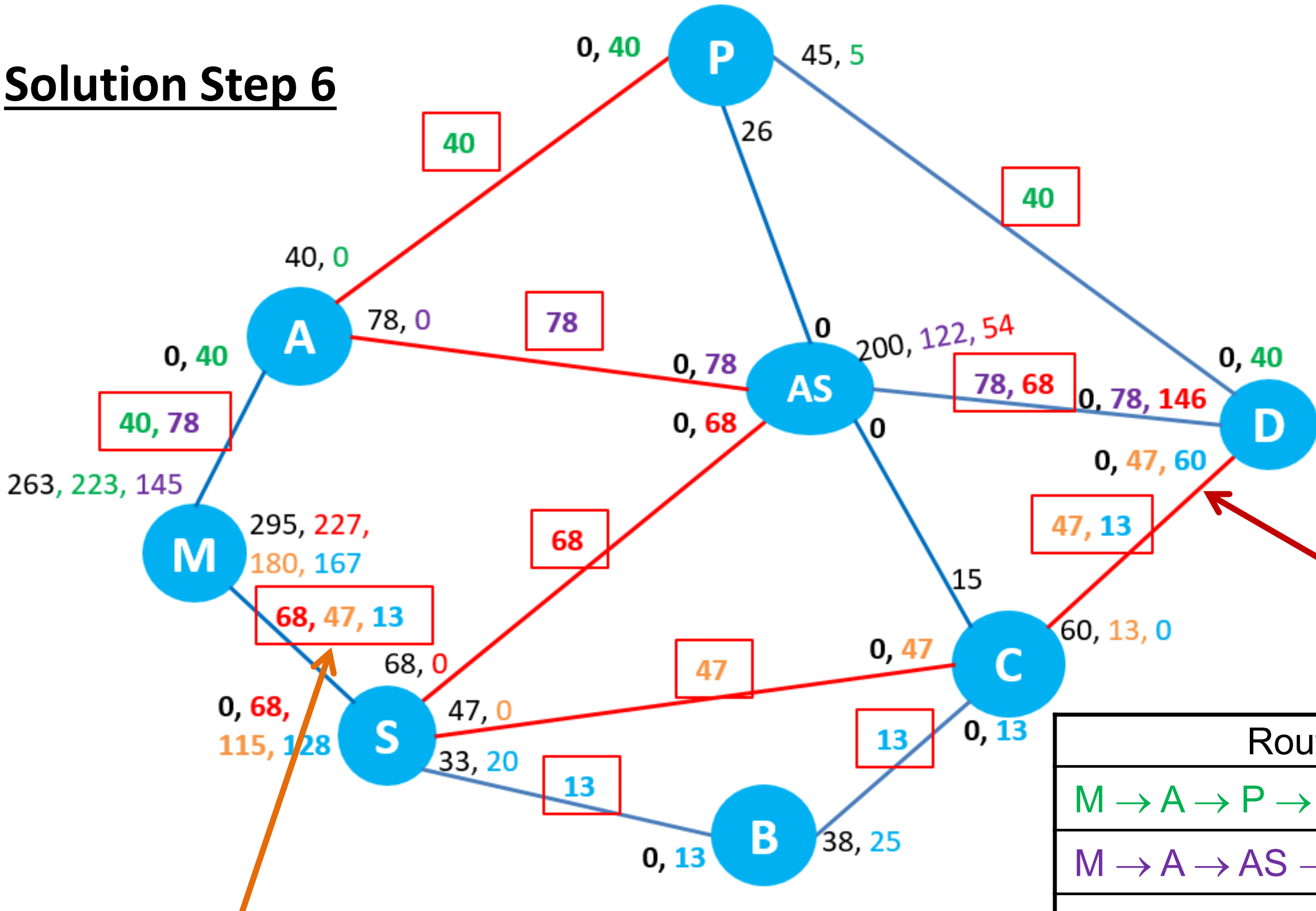
Flow along route M → S:
68 + 47 = 115

Arc in 'red' represents that flow capacity along route S→ C is reached

Route	Flow Quantity
M → A → P → D	40
M → A → AS → D	78
M → S → AS → D	68
M → S → C → D	47
Total	

Today's Problem: Maximum Flow Problem

<u>Solution Step 6</u>



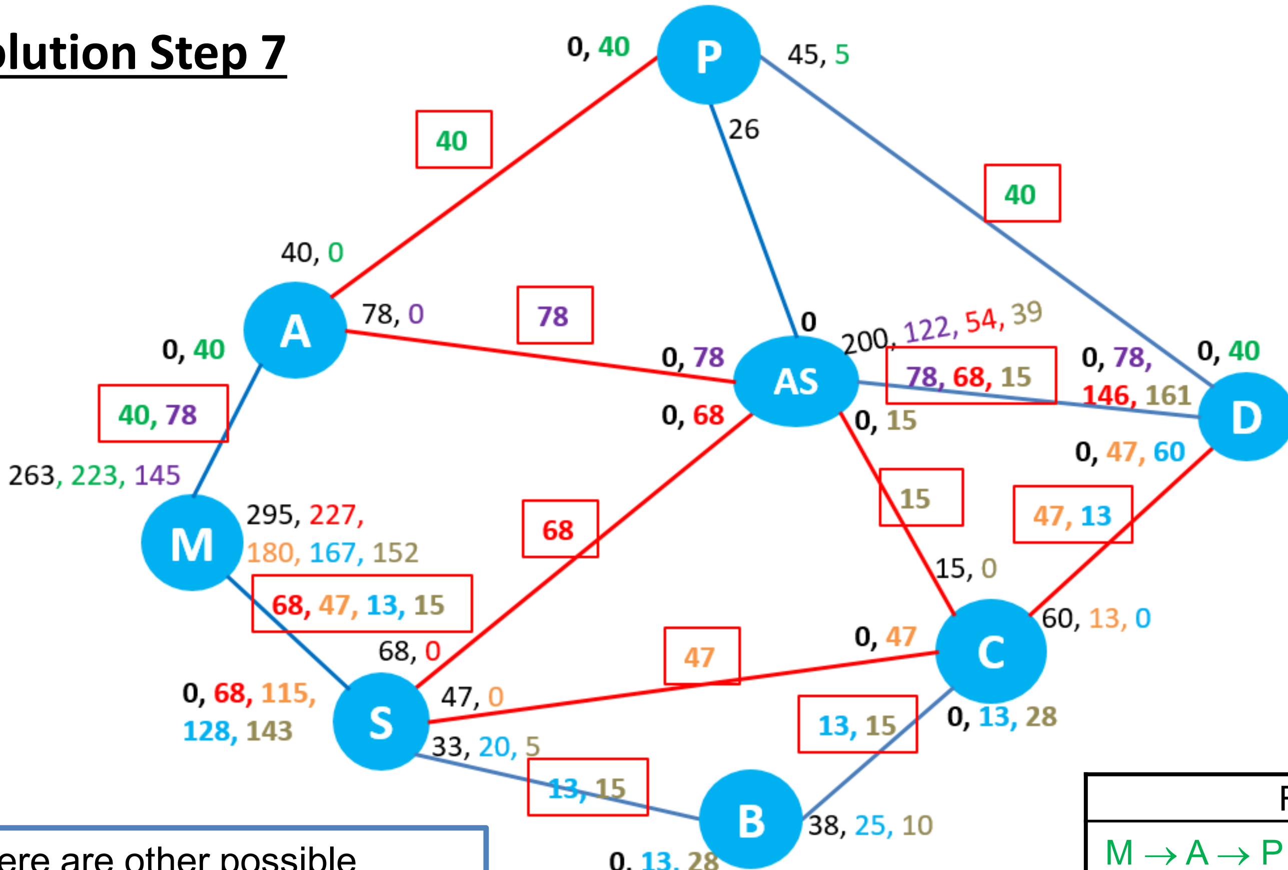
Arc in 'red' represents that flow capacity along route C→ D is reached

Flow along route M → S:
68 + 47 + 13 = 128

Route	Flow Quantity
M → A → P → D	40
M → A → AS → D	78
M → S → AS → D	68
M → S → C → D	47
M → S → B → C → D	13
Total	

Today's Problem: Maximum Flow Problem

Solution Step 7



There are other possible routes to pass through that can result in the maximum amount of 261 containers of products to be delivered from Melbourne to Darwin.

Route	Flow Quantity
M → A → P → D	40
M → A → AS → D	78
M → S → AS → D	68
M → S → C → D	47
M → S → B → C → D	13
M → S → B → C → AS → D	15
Total	261

Conclusion

- The cheapest product delivery routes from Melbourne to each of the other 7 cities can be found by solving it as a Shortest Path Problem.

- The maximum flow of products from Melbourne to Darwin and the planning of delivery on each route can be determined by solving it as a Maximum Flow Problem.

Learning Objectives

- Apply Dijkstra’s Algorithm to solve Shortest Path (can be shortest distance / minimum cost / shortest time between an origin and various destination points) problem.

- Apply Ford-Fulkerson’s Algorithm to solve Maximum Flow (maximize the amount of flow of items from an origin to a destination when the arcs of the network have limited flow capacities) problem.

Overview of E211 Operations Planning II Module