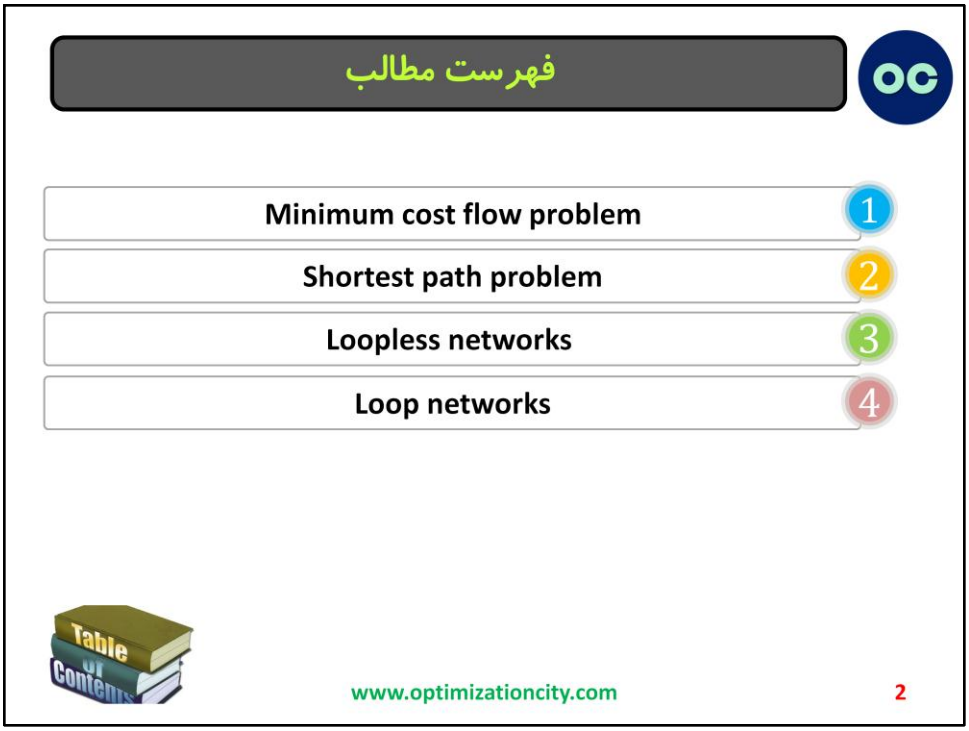
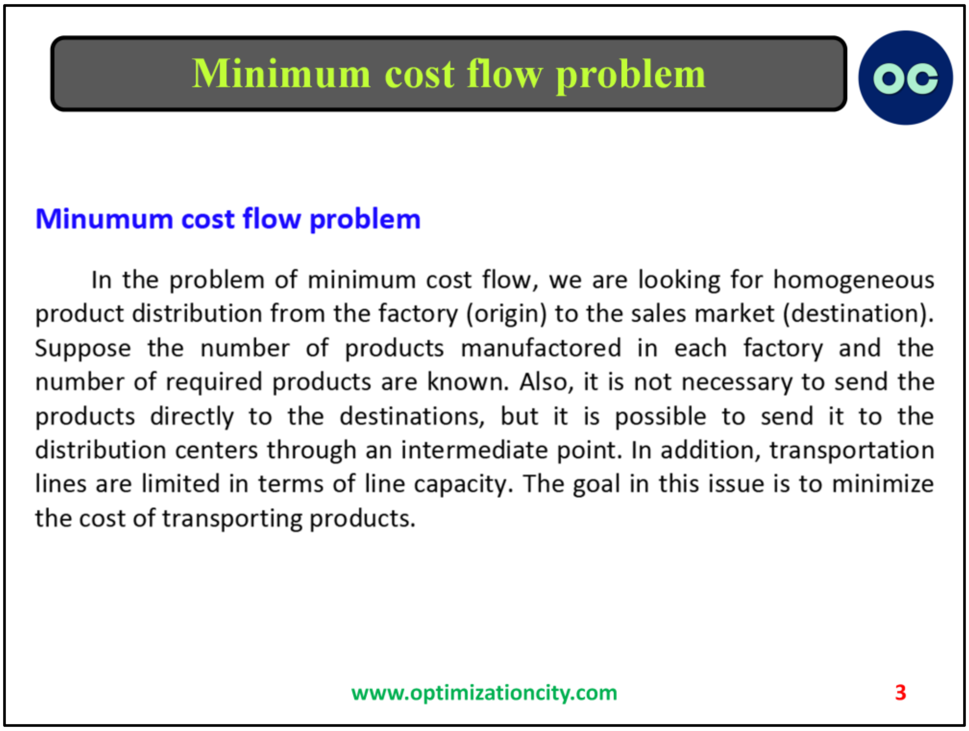
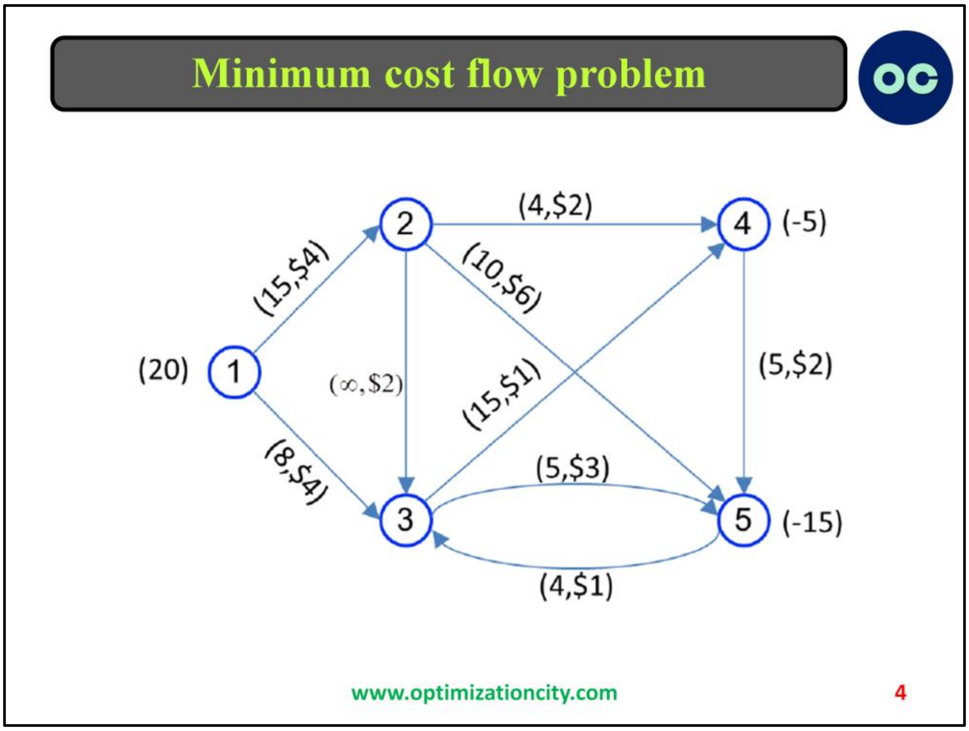
Welcome to the sixth course of optmiztion city.com! Today, we're going to explore something called the "shortest path problem." It's a really interesting concept that involves finding the quickest or most efficient route between different places on a map. Think of it like finding the fastest way to get from point A to point B. This problem comes up in lots of real-life situations, like planning travel routes, organizing deliveries, or even just figuring out the shortest way to walk to your friend's house. It's like solving a puzzle to find the best path and save time or effort. So, get ready to dive into the world of finding the shortest paths, and let's have some fun with it! Let's get started!



The table of contents includes the following sections: Minimum Cost Flow Problem, Shortest Path Problem, Loopless Networks, and Loop Networks. Each section explores its respective topic, providing an introduction, key concepts, and examples.



In the minimum cost flow problem, we want to distribute the same type of product from the factory to the sales market efficiently. We know the quantity of products produced at each factory and the required quantity at each destination. Instead of sending products directly to the destinations, we can send them to distribution centers through an intermediate point if needed. However, there are limits on how much can be transported on each transportation line. The objective is to minimize the cost of transporting the products while meeting the demand.



Let's look at a simple example of the minimum cost flow problem using the diagram below. The nodes are shown as numbered circles, and the arcs are represented by arrows. The arcs have a specific direction, meaning materials or goods can be transported from one node to another in that direction. For instance, we can send materials from node 1 to node 2, but we cannot send them from node 2 to node 1. We use the notation i-j to indicate the arc going from node i to node j.

In the diagram above, each arrow represents a path with a specific capacity and cost for transportation. The capacity indicates the maximum amount that can be transported along that path, while the cost per unit represents the expense associated with each unit of transportation. For instance, in the arrow (2-4), the flow can range from 0 to 4 units, and each unit passing through this path costs $2. The symbol ∞ indicates that there is no limit on the capacity of that particular path.

Moreover, the numbers in parentheses next to the circles represent the available supply and the required demand at each node. In this diagram, node 1 is the starting point with a supply of 20 units. Nodes 4 and 5 are the destinations that require 5 and 15 units respectively, indicated by the negative sign.

Minimum cost flow problem

x_{ij}: is the number of units transported from node i to node j using arc i-j.

$$
\begin{aligned}
&Min \quad 4x_{12}+4x_{13}+2x_{23}+2x_{24}+6x_{25}+x_{34}+3x_{35}+2x_{45}+x_{53}\\
&s.t.\\
&(1)\quad x_{12}+x_{13}=20\\
&(2)\quad -x_{12}+x_{23}+x_{24}+x_{25}=0\\
&(3)\quad -x_{13}-x_{23}+x_{34}+x_{35}-x_{53}=0\\
&(4)\quad -x_{24}-x_{34}+x_{45}=-5\\
&(5)\quad -x_{25}-x_{35}-x_{45}+x_{53}=-15\\
&x_{12}\le 15; x_{13}\le 8; x_{23}\le \infty; x_{24}\le 4; x_{25}\le 10; x_{34}\le 15; x_{35}\le 5; x_{45}\le \infty; x_{53}\le 4.
\end{aligned}
$$

In the minimum cost flow problem, our objective is to discover the flow pattern that incurs the lowest cost. To formulate this problem as a linear programming model, let's consider the following notation:
xij: represents the quantity of units transported from node i to node j using the arc i-j.
Now, let's present the linear programming model for the minimum cost flow problem.

Equations 1 to 5 represent the flow balance equations in the network. Let's consider the balance equation at node 1 as an example.
The equation states that the combined flow leaving node 1 (x12 + x13) should be equal to the supply available at node 1, which is 20 units. This ensures that the outgoing flow matches the available supply.

Similarly, the balance equation at node 2 indicates that the incoming flow to node 2 (x12) is equal to the outgoing flow from node 2 (x23 + x24 + x25). This ensures that the flow entering and leaving node 2 is properly balanced.

The minimum cost flow model in the network has a unique structure that helps determine the solution algorithm. The flow variables, denoted as xij in the balance equations, have coefficients of 0, +1, or -1. Additionally, these variables appear precisely in two balance equations: once with a coefficient of +1 representing the node where the flow originates, and once with a coefficient of -1 representing the

node where the flow enters.
Based on the above characteristics, the general form of the Minimum Cost Flow problem can be expressed as follows.

Minimum cost flow problem

$$\begin{aligned}
&Min \quad \sum_i \sum_j c_{ij} x_{ij} \\
&s.t. \\
&\qquad \sum_j x_{ij} - \sum_k x_{ki} = b_i \quad i = 1, \ldots, n \\
&\qquad l_{ij} \le x_{ij} \le u_{ij}
\end{aligned}$$

www.optimizationcity.com 6

The model shown here represents the general form of the minimum cost flow problem. However, we can simplify this model into more straightforward forms, which are explained below.

Shortest path problem

Shortest path problem

The networks mentioned in this course are divided into two types:

- Loopless networks
- Loop networks

In the following, we will examine each of these two types of networks.

In the real world, network theory finds practical applications in determining the shortest path within a network. In this context, networks consist of areas and regions represented as nodes, and the communication paths connecting them are known as arcs. Any node within this network can be considered either the starting point or the destination. The objective is to identify a path between the origin and the destination that results in the shortest distance, known as the shortest path, within the network. The networks covered in this course can be categorized into two types: loopless networks and loop networks. In the upcoming sections, we will explore each of these network types in detail, examining their characteristics, properties, and applications.

Loopless network

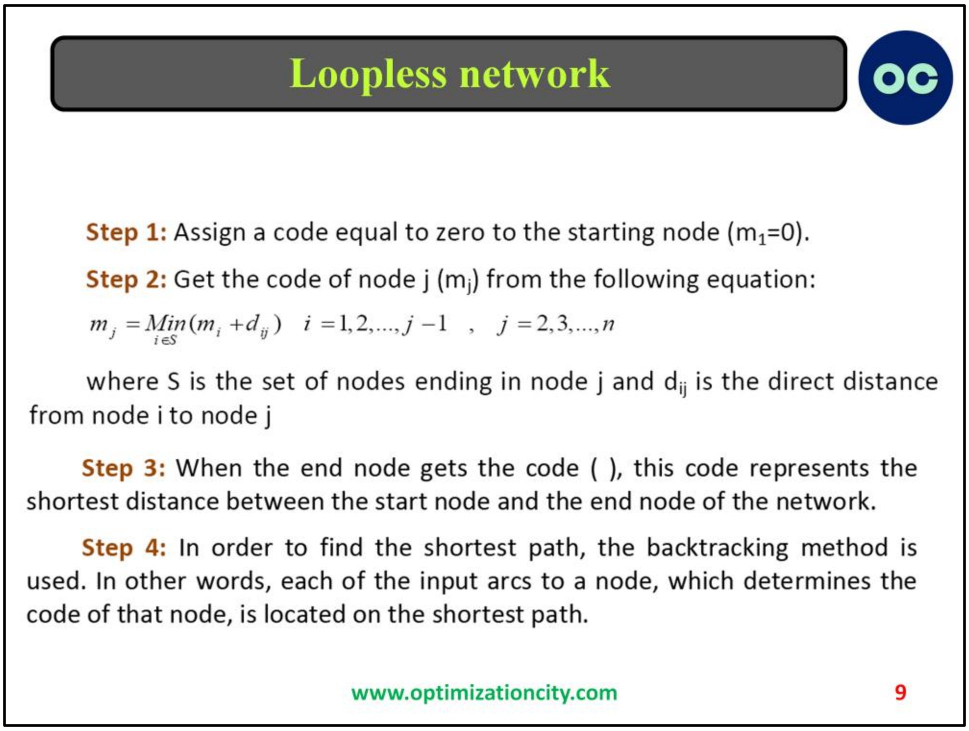
Loopless networks

Simple shortest path method

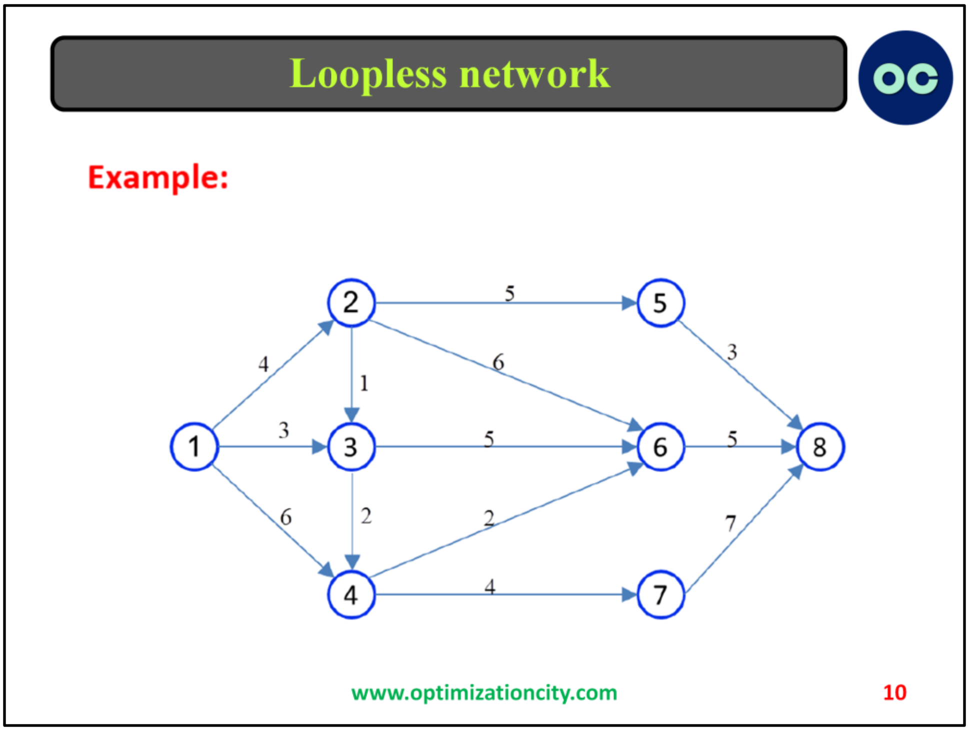
In this method, a computational movement is performed from the origin node (start) to the destination node (end). During this computational move, each node is assigned a code (m) that represents the shortest distance of that node from the starting node. It is assumed that the number of the starting node is equal to 1 and the number of the end node is to n, and the middle nodes are also numbered in ascending order. The algorithm steps of this method are as follows:

www.optimizationcity.com 8

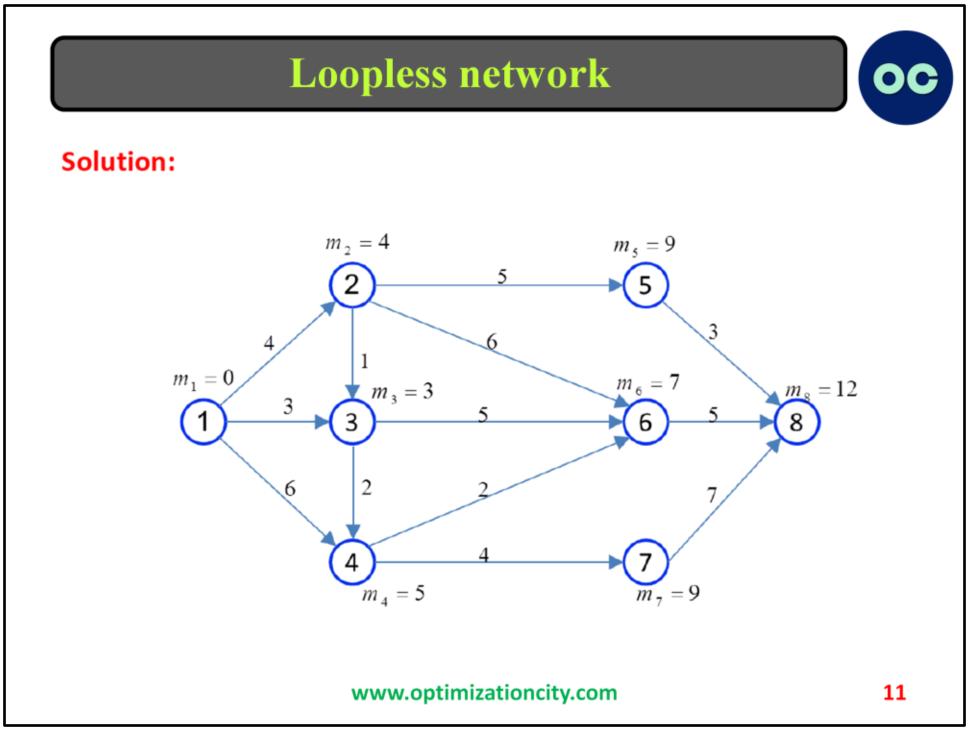
Loopless networks are similar to basic networks, but in this case, both nodes can act as the origin and destination nodes for movement.
In the simple shortest path method, computation begins from the origin node (start) and continues towards the destination node (end). During this computation, each node is assigned a code (m) that represents its shortest distance from the starting node. The starting node is assumed to have a number of 1, the end node is numbered as n, and the intermediate nodes are sequentially numbered. The algorithm follows these steps:



Step 1: Start by setting the code of the starting node to zero (m1=0).
Step 2: Calculate the code of node j (mj) using the following equation:
mj = min(mk + dij)
where S is the set of nodes that lead to node j, and dij represents the direct distance between node i and node j.
Step 3: Once the code of the end node is determined, it represents the shortest distance between the starting node and the end node in the network.
Step 4: To find the shortest path, we employ the backtracking method. This means that each input arc leading to a node, which determines the code of that node, is part of the shortest path.



Let's apply the method we just learned to find the shortest path between nodes 1 and 8 in the given network. The distance between each pair of nodes is indicated on the connecting lines.



Here's how we can find the shortest path between nodes 1 and 8 in the given network using the method we discussed. We start by setting the code of node 1 to zero (m1=0).

Next, we calculate the code for node 2 by adding the distance from node 1 to node 2 (m2=m1+d12=0+4=4).

Moving on to node 3, since there are two paths to reach it, we need to consider both of these paths:

[Path 1: m3 = m1 + d13 = 0 + 3 = 3]

[Path 2: m3 = m2 + d23 = 4 + 2 = 6]

We continue these calculations for the other nodes, following the same process. The summarized results can be seen in the figure below. From the figure, we can observe that the shortest distance from node 1 to node 8 is 12.

Loopless network

$$m_8 - d_{58} = 12 - 3 = 9 = m_5$$
$$m_8 - d_{68} = 12 - 5 = 7 = m_6$$
$$m_8 - d_{78} = 12 - 7 = 5 \neq m_7$$

Arcs (5,8) and (6,8) are on the shortest path. Therefore, there are two shortest paths so far. First, we continue working from node 5.

$$m_5 - d_{25} = 9 - 5 = 4 = m_2$$

The arc (2,5) lies on the shortest path.

$$m_2 - d_{12} = 4 - 4 = 0 = m_1$$

The arc (1,2) is on the shortest path. The first route was determined. Now it's time for the second path. we continue the process from the sixth node.

In order to determine the shortest path between nodes 1 and 8, we use a method called backtracking. This involves checking each node (j) that meets the condition mi = mj - dij, which helps us identify the nodes along the shortest path. We start the process from the end node and explore the three branches connected to it.

Loopless network

$$m_6 - d_{26} = 7 - 6 = 1 \neq m_2$$
$$m_6 - d_{36} = 7 - 5 = 2 \neq m_3$$
$$m_6 - d_{46} = 7 - 2 = 5 = m_4$$

The arc (4,6) is on the shortest path.

$$m_4 - d_{34} = 5 - 2 = 3 = m_3$$
$$m_4 - d_{14} = 5 - 6 = -1 \neq m_1$$

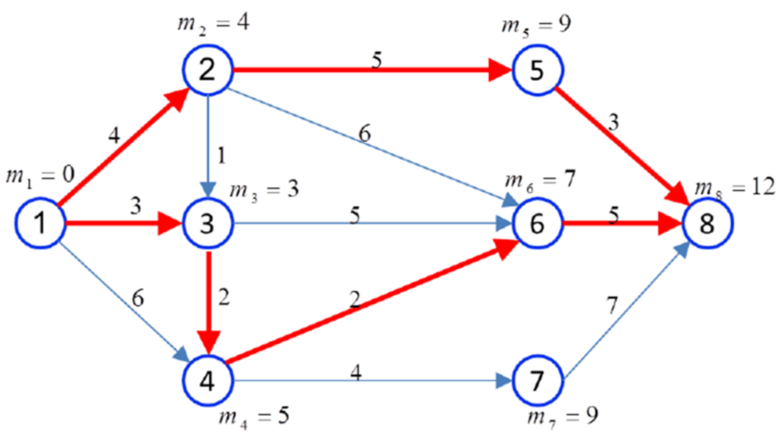
Arc (3,4) lies on the shortest path.

$$m_3 - d_{23} = 3 - 1 = 2 \neq m_2$$
$$m_3 - d_{13} = 3 - 3 = 0 = m_1$$

The arc (1,3) lies on the shortest path. It can be seen that there are two paths in this network, which has the shortest path.

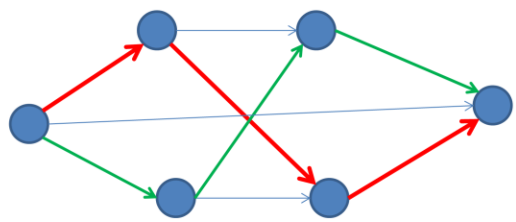
$$1 \to 2 \to 5 \to 8$$
$$1 \to 3 \to 4 \to 6 \to 8$$

Loopless network



Loopless network

The k-th shortest path

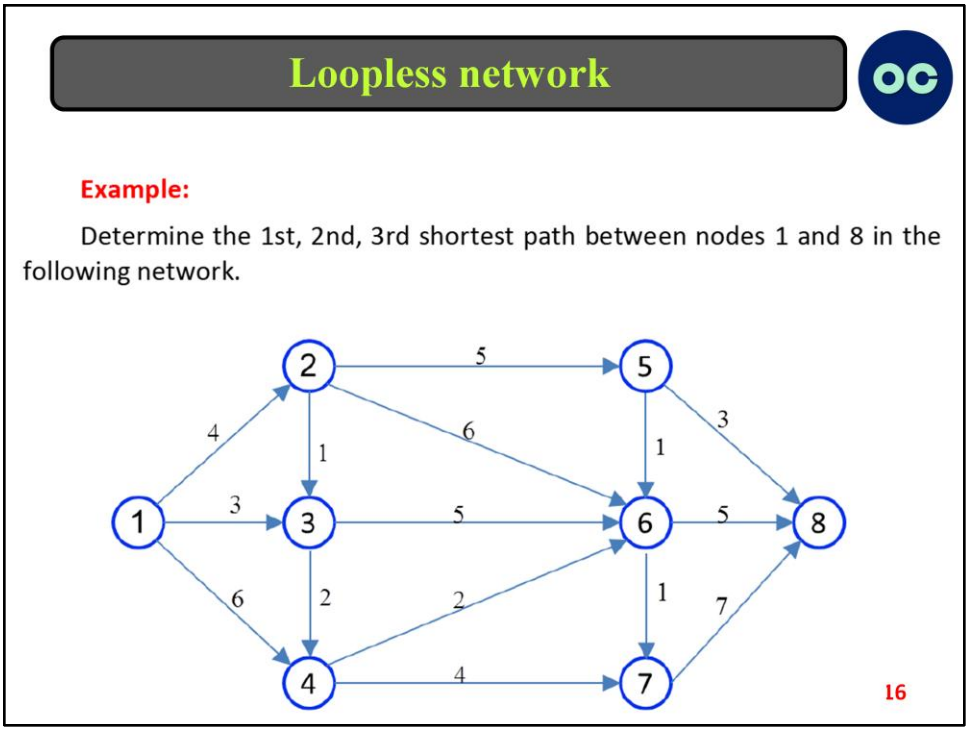


$$m_j^{(k)} = \underset{i \in S}{Mink}\,\{m_i^{(r)} + d_{ij}\} \quad = 1,2,\ldots,j-1; j = 2,3,\ldots,n; 1 \le r \le k$$

where Mink means choosing the k-th minimum value inside {}.

www.optimizationcity.com 15

Let's Move one the k-th shortest path
consider the scenario where a bus driver needs to take passengers from the origin city (X) to the destination city (Z) via cities U, V, and W, choosing the shortest route. However, during a specific week, there is an obstacle preventing the bus from going through city U. In this situation, the bus driver intends to use the second shortest route as an alternative. This example highlights the need to determine the second, third, or k-th shortest path in certain cases. To find the k-th shortest path, a modification is made to the second step of the simple shortest path method.



Let's find the 1st, 2nd, and 3rd shortest paths between nodes 1 and 8 in the given network.

Loopless network

Solution:

Using the simple shortest path method, the shortest path (the first shortest path) to each node is obtained:

$$m_1^{(1)} = 0 \quad , \quad m_2^{(1)} = 4 \quad , \quad m_3^{(1)} = 3 \quad , \quad m_3^{(1)} = 5$$
$$m_5^{(1)} = 9 \quad , \quad m_6^{(1)} = 7 \quad , \quad m_7^{(1)} = 8 \quad , \quad m_8^{(1)} = 12$$

Regarding the second short path, the source node is still assigned a code equal to zero.

$$m_1^{(2)} = 0$$

Only one branch enters node 2, so there is no second shortest distance between nodes 1 and 2:

$$m_2^{(2)} = -$$

Solution:
Applying the simple shortest path method, we first obtain the shortest path (which is the 1st shortest path) to each node:

Now, let's determine the second and third shortest paths between nodes 1 and 8 using the backtracking method. There are two possible solutions for the second shortest path and one solution for the third shortest path:
Second Shortest Path:
Since only one branch enters node 2, there is no second shortest distance between nodes 1 and 2.

Loopless network

But for node 3, we can write:

$$m_3^{(2)} = Min2\{m_1^{(1)} + d_{13}, m_1^{(2)} + d_{13}, m_2^{(1)} + d_{23}, m_2^{(2)} + d_{23}\}$$
$$= Min2\{0+3, 0+3, 4+1, -\} = 5$$

For other nodes, we proceed as follows:

$$m_4^{(2)} = Min2\{m_1^{(1)} + d_{14}, m_1^{(2)} + d_{14}, m_3^{(1)} + d_{34}, m_3^{(2)} + d_{34}\}$$
$$= Min2\{0+6, 0+6, 3+2, 5+2\} = 6$$

$$m_5^{(2)} = -$$

$$m_6^{(2)} = Min2\{m_2^{(1)} + d_{26}, m_2^{(2)} + d_{26}, m_3^{(1)} + d_{36}, m_3^{(2)} + d_{36}, m_4^{(1)} + d_{46}, m_4^{(2)} + d_{46}, m_5^{(1)} + d_{56}, m_5^{(2)} + d_{56}\}$$
$$= Min2\{4+6, -, 3+5, 5+5, 5+2, 6+2, 9+1, -\} = 8$$

$$m_7^{(2)} = Min2\{m_4^{(1)} + d_{47}, m_4^{(2)} + d_{47}, m_6^{(1)} + d_{67}, m_6^{(2)} + d_{67}\}$$
$$= Min2\{5+4, 6+4, 7+1, 8+1\} = 9$$

www.optimizationcity.com 18

For node 3, we have:

Loopless network

$$m_8^{(2)} = Min2\{m_5^{(1)} + d_{58}, m_5^{(2)} + d_{58}, m_6^{(1)} + d_{68}, m_6^{(2)} + d_{68}, m_7^{(1)} + d_{78}, m_7^{(2)} + d_{78}\}$$
$$= Min2\{9+3, -, 7+5, 8+5, 8+7, 9+7\} = 13$$

The third shortest distance from the origin node to itself is also equal to zero:

$$m_1^{(3)} = 0$$

Because only one branch enters node 2, there is also no third shortest distance between nodes 1 and 2.

$$m_2^{(3)} = -$$

For node 3 we have:

$$m_3^{(3)} = Min3\{m_1^{(1)} + d_{13}, m_1^{(2)} + d_{13}, m_1^{(3)} + d_{13}, m_2^{(1)} + d_{23}, m_2^{(2)} + d_{23}, m_2^{(3)} + d_{23}\}$$
$$= Min3\{0+3, 0+3, 0+3, 4+1, -, -\} = 5$$

www.optimizationcity.com 19

Third Shortest Path:
Similarly, there is no third shortest distance between nodes 1 and 2, as only one branch enters node 2.
For node 3, we have:

Loopless network

In the same way, we can write:

$$m_4^{(3)} = Min3\{m_1^{(1)} + d_{14}, m_1^{(2)} + d_{14}, m_1^{(3)} + d_{14}, m_3^{(1)} + d_{34}, m_3^{(2)} + d_{34}, m_3^{(3)} + d_{34}\}$$
$$= Min3\{0+6, 0+6, 0+6, 3+2, 5+2, 5+2\} = 7$$

$$m_5^{(3)} = -$$

$$m_6^{(3)} = Min3\{m_2^{(1)} + d_{26}, m_2^{(2)} + d_{26}, m_2^{(3)} + d_{26}, m_3^{(1)} + d_{36}, m_3^{(2)} + d_{36}, m_3^{(3)} + d_{36}, m_4^{(1)} + d_{46}, m_4^{(2)}$$
$$+ d_{46}, m_4^{(3)} + d_{46}, m_5^{(1)} + d_{56}, m_5^{(2)} + d_{56}, m_5^{(3)} + d_{56}\}$$
$$= Min3\{4+6, -, -, 3+5, 5+5, 5+5, 5+2, 6+2, 7+2, 9+1, -\} = 9$$

$$m_7^{(3)} = Min3\{m_4^{(1)} + d_{47}, m_4^{(2)} + d_{47}, m_4^{(3)} + d_{47}, m_6^{(1)} + d_{67}, m_6^{(2)} + d_{67}, m_6^{(3)} + d_{67}\}$$
$$= Min3\{5+4, 6+4, 7+4, 7+1, 8+1, 9+1\} = 10$$

$$m_8^{(3)} = Min3\{m_5^{(1)} + d_{58}, m_5^{(2)} + d_{58}, m_5^{(3)} + d_{58}, m_6^{(1)} + d_{68}, m_6^{(2)} + d_{68}, m_6^{(3)} + d_{68},$$
$$m_7^{(1)} + d_{78}, m_7^{(2)} + d_{78}, m_7^{(3)} + d_{78}\}$$
$$= Min3\{9+3, -, -, 7+5, 8+5, 9+5, 8+7, 9+7, 10+7\} = 14$$

We can continue this process for the remaining nodes:

Loopless network

The second short path:

$1 \rightarrow 3 \rightarrow 6 \rightarrow 8 \quad \& \quad 1 \rightarrow 4 \rightarrow 6 \rightarrow 8$

The third short route:

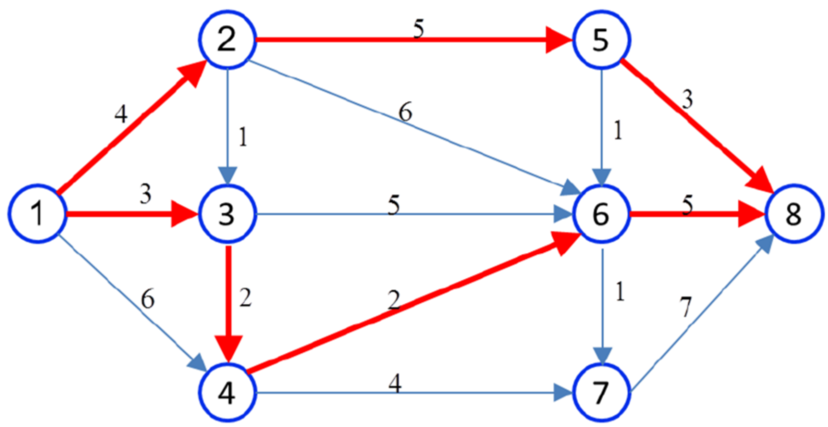
$1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 6 \rightarrow 8$

Three types of routes are shown in the following figures:

The second and third shortest paths between nodes 1 and 8 are determined by examining the nodes and branches using backtracking. In this case, there are two possible answers for the second shortest path and one possible answer for the third shortest path.

Loopless network

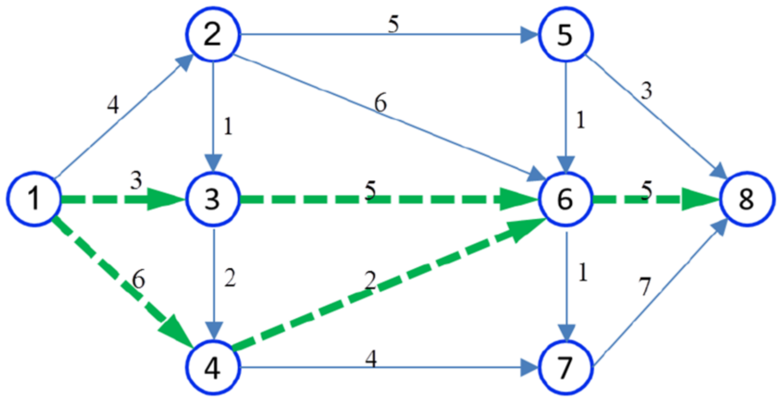
The shortest routes



www.optimizationcity.com

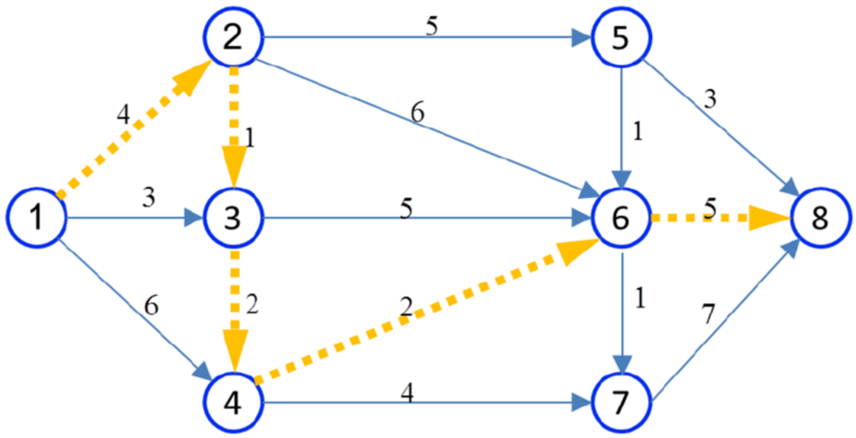
Loopless network

The second short paths



Loopless network

The third short paths



Loop networks

Loop networks

Dijkestra's method

The Dijkestra method is named after its inventor, and two codes are assigned to each node:

m_i: the shortest distance of node i from the origin node to the current step of the algorithm, which is called the *temporary code*. The shortest distance may also be obtained in the next steps.

M_i: the shortest distance of node i from the origin node, which is called the *permanent code*, and a shorter distance will not be obtained in the next steps.

www.optimizationcity.com 25

In loop networks, there are arcs that create loops, allowing movement between nodes in both forward and backward directions. To find the shortest path in such networks, two methods are commonly used:
Dijkstra's method:
This method is named after its inventor. In Dijkstra's method, two codes are assigned to each node:
mi: This represents the shortest distance of node i from the origin node up to the current step of the algorithm. It is known as the temporary code since the shortest distance may be updated in subsequent steps.
Mi: This represents the shortest distance of node i from the origin node and is referred to as the permanent code. Once assigned, this code will not be updated with a shorter distance in the following steps.

Loop networks

Dijkestra's algorithm is as follows:

Step 1: Assign a permanent code equal to zero to the start node (M1=0(

Step 2: Assign a temporary code to the nodes adjacent to the nodes that have received a permanent code using the following relationship:

$$m_j = M_i + d_{ij}$$

where d_{ij} is the direct distance from node i with permanent code to node j which is the temporary code:

temporary code *permanent code*.

www.optimizationcity.com 26

Dijkstra's algorithm can be summarized in the following steps:
Step 1: Start by assigning a permanent code of zero to the starting node (M1=0).
Step 2: Assign a temporary code to the nodes adjacent to the nodes that already have a permanent code using the following formula:
Temporary Code of Node j = Minimum of (Temporary Code of Node i + dij)
Here, dij represents the direct distance from a node i with a permanent code to the adjacent node j with a temporary code. If a node already has a temporary code, it is compared with the new temporary code, and the smaller value is chosen. Among all the nodes with temporary codes, the node with the smallest code is converted to a permanent code. The process continues by assigning temporary codes to the nodes adjacent to the node with the newly assigned permanent code, and so on.

Loop networks

Step 3: Stop when the end (destination) node has received the permanent code. M_n represents the shortest distance between the end node and the start node.

Step 4: In order to find the shortest path, use the backtracking method with the following relation:

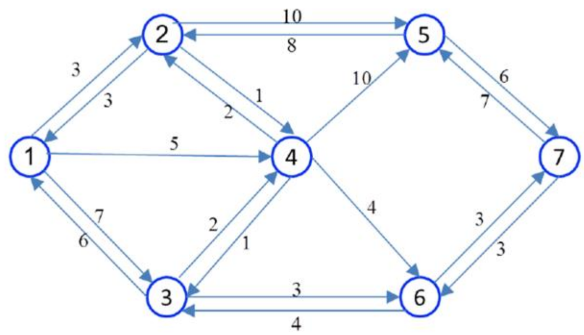
$$M_i = M_j - d_{ij}$$

Step 3: Stop the algorithm when the end (destination) node receives a permanent code. The value of Mn represents the shortest distance between the end node and the start node.
Step 4: To find the shortest path, use the backtracking method with the following relationship:
Shortest Path from Node i to Node j = Shortest Path from Node i to Node k + Direct Distance from Node k to Node j
This relation helps determine the shortest path by considering the nodes along the way.
To better understand the steps of the algorithm, let's take a look at the following example.

Loop networks

Example:

Determine the shortest distance and path between nodes 1 and 7 in the following network using Dijkestra 's method.



Let's use Dijkstra's method to find the shortest distance and path between nodes 1 and 7 in the given network.

Loop networks

Solution:

$$M_1 = 0$$

$$m_2 = M_1 + d_{12} = 0 + 3 = 3$$
$$m_3 = M_1 + d_{13} = 0 + 7 = 7$$
$$m_4 = M_1 + d_{14} = 0 + 5 = 5$$

$$Min(3,7,5) = 3 \rightarrow M_2 = 3$$

Here's the solution to finding the shortest distance and path between nodes 1 and 7 using Dijkstra's method:
1: We start by assigning a permanent code of zero to node 1, which is the origin of movement.
2: Since node 1 has received a permanent code, we assign temporary codes to its adjacent nodes.
3: Among the temporary codes available so far, the minimum value is 3. Therefore, the temporary code of node 2 is converted into a permanent code.

Loop networks

$$m_4 = M_2 + d_{24} = 3 + 1 = 4$$
$$m_5 = M_2 + d_{25} = 3 + 10 = 13$$

$\Downarrow$

$$m_4 = Min(5,4) = 4$$

$\Downarrow$

$$Min(7,4,13) = 4 \rightarrow M_4 = 4$$

www.optimizationcity.com 30

4: We assign temporary codes to the nodes adjacent to node 2, which now has a permanent code.
5: Node 4 already had a temporary code of 5, but at this stage, a new temporary code of 4 is obtained. Since the new temporary code is smaller, it replaces the previous code.
Step 6: The minimum temporary codes available so far are 4 for node 4.

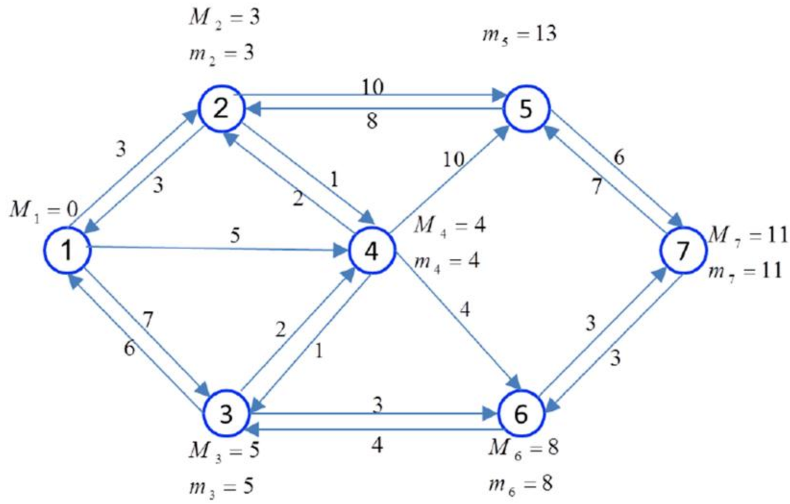
Loop networks

The process continues like this:

$$m_3 = M_4 + d_{43} = 4 + 1 = 5 \rightarrow m_3 = 5$$
$$m_5 = M_4 + d_{45} = 4 + 10 = 14 \rightarrow m_5 = 13$$
$$m_6 = M_4 + d_{46} = 4 + 4 = 8$$
$$Min(m_3, m_5, m_6) = Min(5,13,8) = 5 \rightarrow M_3 = 5$$
$$m_6 = M_3 + d_{36} = 5 + 3 = 8 \rightarrow m_6 = 8$$
$$Min(m_5, m_6) = Min(13,8) = 8 \rightarrow M_6 = 8$$
$$m_7 = M_6 + d_{67} = 8 + 3 = 11$$
$$Min(m_5, m_7) = Min(13,11) = 11 \rightarrow M_7 = 11$$

The process continues in this manner, assigning temporary codes and updating them accordingly.
The diagram below summarizes the results of the different steps of the algorithm.

Loop networks



Loop networks

In order to determine the shortest path between nodes 1 and 7, backtracking is used. We continue as described in networks without loops.

$$m_7 - d_{57} = 11 - 6 = 5 \neq M_5$$
$$m_7 - d_{67} = 11 - 3 = 8 = M_6$$

$$m_6 - d_{46} = 8 - 4 = 4 = M_4$$
$$m_6 - d_{36} = 8 - 3 = 5 = M_3$$

$$m_3 - d_{43} = 5 - 1 = 4 = M_4$$
$$m_3 - d_{13} = 5 - 7 = -2 \neq M_1$$

www.optimizationcity.com 33

To find the shortest path between nodes 1 and 7, we can use backtracking. We follow a similar approach as in networks without loops:
We start by examining the arcs and nodes along the shortest path.

The arc (6,7) is part of the shortest path.

The arcs (4,6) and (3,6) are also part of the shortest paths.

The arc (3,4) is on the shortest path.

Loop networks

$$m_2 - d_{12} = 3 - 3 = 0 = M_1$$
$$m_2 - d_{25} = 3 - 8 = -5 \neq M_5$$

$$\Downarrow$$

$$1 \to 2 \to 4 \to 6 \to 7$$
$$1 \to 2 \to 4 \to 3 \to 6 \to 7$$

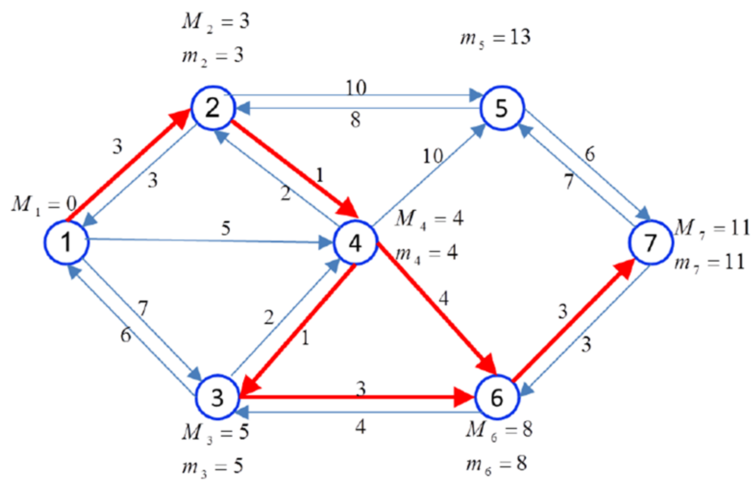
The figure below shows the shortest network paths as follows:

www.optimizationcity.com 34

The arc (2,4) is included in the shortest path.

Finally, the arc (1,2) is part of the shortest path.
Therefore, in this network, there are two paths that have the shortest distance between nodes 1 and 7.
The figure below illustrates the shortest paths in the network:

Loop networks



H1

Loop networks

Comprehensive shortest path method

The comprehensive shortest path method is a way to find the shortest route in a network. Dijkestra's algorithm is a simple method for this, but it has a limitation. It only determines the shortest distance between the starting and ending nodes in each step of the algorithm. However, with the comprehensive shortest path method, we can calculate the shortest path between all nodes in the network. If the network has n nodes, we start numbering them from 1 to n, with the starting node as 1 and the ending node as n. The rest of the nodes are numbered in ascending order from the starting node to the ending node.

Loop networks

$$D = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1n} \\ d_{21} & d_{2} & \dots & d_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ d_{n1} & d_{n2} & \cdots & d_{nn} \end{bmatrix} \qquad P = \begin{bmatrix} p_{11} & p_{12} & \dots & p_{1n} \\ p_{21} & p_{2} & \dots & p_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{bmatrix}$$

∞

www.optimizationcity.com 37

In this method, we use two matrices: one for distances (D) and another for paths (P). The value dij represents the distance between node i and node j, while pij represents the path from node i to node j. If there is no direct path between two nodes, we consider the distance between them as infinity (∞) and mark the path with a "-" sign. It's important to note that the distance from one node to another (dij) may not be the same as the distance for the return journey (dji).
For example, d12 represents the distance from node 1 to node 2. Similarly, p12 represents the path from node 1 to node 2.

Type equation here.

Loop networks

In each stage j (j=1,2,...,n), node j is considered as an intermediary node and the route between any two nodes other than node j is obtained using this intermediary. If this distance is less than the previous distance, it will be replaced, otherwise the previous distance will be kept. In other words, the j-1 step matrices, namely D_{j-1} and P_{j-1}, will be converted into j step matrices, namely D_j and P_j, by the following relations.

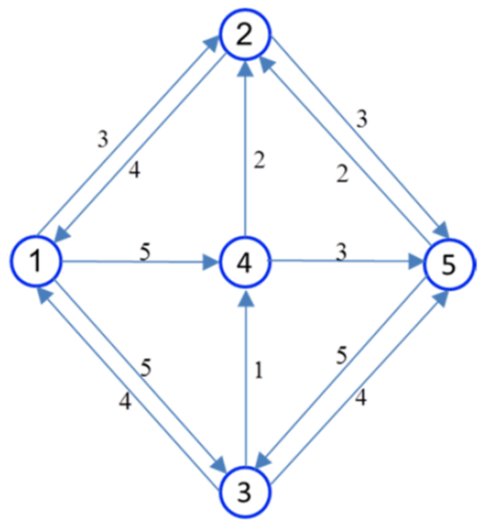
$$d_{ik} = \left.\begin{cases} d_{ik} & \text{if } d_{ik} \le d_{ij} + d_{jk} \\ d_{ij} + d_{jk} & \text{if } d_{ik} > d_{ij} + d_{jk} \end{cases}\right\} = Min(d_{ik}, d_{ij} + d_{jk}), i \ne j \ne k$$

$$p_{ik} = \begin{cases} p_{ik} & \text{if } d_{ik} \le d_{ij} + d_{jk} \\ p_{ij} & \text{if } d_{ik} > d_{ij} + d_{jk} \end{cases}$$

At each stage (j=1,2,...,n) of the process, we select node j as an intermediary node. Then, we calculate the route between any two nodes excluding node j, using this intermediary. If the newly calculated distance is shorter than the previous distance, we update it with the new value. Otherwise, we keep the previous distance. In simpler terms, we transform the matrices Dj-1 and Pj-1 from the (j-1)th step into Dj and Pj for the jth step using the following relationships.
Let me illustrate this with an example to provide further clarification.

Loop networks

Example:



39

Example:

Using the comprehensive shortest path method, find the shortest routes between all the nodes in the network shown in the diagram below.

Loop networks

Solution:

First, we form the matrices D_0 and P_0.

$$D_0 = \begin{bmatrix} 0 & 3 & 5 & 5 & \infty \\ 4 & 0 & \infty & \infty & 3 \\ 4 & \infty & 0 & 1 & 4 \\ \infty & 2 & \infty & 0 & 3 \\ \infty & 2 & 5 & \infty & 0 \end{bmatrix} \qquad P_0 = \begin{bmatrix} - & 2 & 3 & 4 & - \\ 1 & - & - & - & 5 \\ 1 & - & - & 4 & 5 \\ - & 2 & - & - & 5 \\ - & 2 & 3 & - & - \end{bmatrix}$$

Loop networks

Stage 1 (intermediary node 1): the distance between nodes 2 and 3 in the matrix is equal to ∞, while through the intermediary node, i.e. node 1, this distance will be equal to 9. Because the distance through the intermediary is less than the distance in the matrix, it is replaced by:

$d_{23} = Min(d_{23}, d_{21} + d_{13}) = Min(\infty, 4+5) = 9 \rightarrow$ Distance and route do change

Up to this stage, to get from node 2 to node 3, you have to move through node 1, and the distance is equal to 9. The same is done for other nodes:

$d_{24} = Min(d_{24}, d_{21} + d_{14}) = Min(\infty, 4+5) = 9 \rightarrow$ Distance and route do change

$d_{25} = Min(d_{25}, d_{21} + d_{15}) = Min(3, 4+\infty) = 3 \rightarrow$ Distance and route do not change

$d_{32} = Min(d_{32}, d_{31} + d_{12}) = Min(\infty, 4+3) = 7 \rightarrow$ Distance and route do change

Stage 1 (with intermediary node 1): In the matrix, the distance between nodes 2 and 3 is marked as infinity (∞). However, when we consider the path through intermediary node 1, the distance between nodes 2 and 3 becomes 9. Since this intermediary path is shorter than the distance in the matrix, we replace it with the new value.

Loop networks

$d_{34} = Min(d_{34}, d_{31} + d_{14}) = Min(1, 4+5) = 1 \rightarrow$ Distance and route do not change

$d_{35} = Min(d_{35}, d_{31} + d_{15}) = Min(4, 4+\infty) = 4 \rightarrow$ Distance and route do not change

$d_{42} = Min(d_{42}, d_{41} + d_{12}) = Min(2, \infty+3) = 2 \rightarrow$ Distance and route do not change

$d_{43} = Min(d_{43}, d_{41} + d_{13}) = Min(\infty, \infty+5) = \infty \rightarrow$ Distance and route do not change

$d_{45} = Min(d_{45}, d_{41} + d_{15}) = Min(3, \infty+\infty) = 3 \rightarrow$ Distance and route do not change

$d_{52} = Min(d_{52}, d_{51} + d_{12}) = Min(2, \infty+3) = 2 \rightarrow$ Distance and route do not change

$d_{53} = Min(d_{53}, d_{51} + d_{13}) = Min(5, \infty+5) = 5 \rightarrow$ Distance and route do not change

$d_{54} = Min(d_{54}, d_{51} + d_{14}) = Min(\infty, \infty+5) = \infty \rightarrow$ Distance and route do not change

Loop networks

It can be seen that at this stage the distance and route change from node 2 to 4 and also from node 3 to 2. Therefore, the matrices of this step are:

$$D_1 = \begin{bmatrix} 0 & 3 & 5 & 5 & \infty \\ 4 & 0 & 9 & 9 & 3 \\ 4 & 7 & 0 & 1 & 4 \\ \infty & 2 & \infty & 0 & 3 \\ \infty & 2 & 5 & \infty & 0 \end{bmatrix} \qquad P_1 = \begin{bmatrix} - & 2 & 3 & 4 & - \\ 1 & - & 1 & 1 & 5 \\ 1 & 1 & - & 4 & 5 \\ - & 2 & - & - & 5 \\ - & 2 & 3 & - & - \end{bmatrix}$$

Stage 2 (intermediary node 2)

$d_{13} = Min(d_{13}, d_{12} + d_{23}) = Min(5, 3+9) = 5 \rightarrow$ Distance and route do not change

$d_{14} = Min(d_{14}, d_{12} + d_{24}) = Min(5, 3+9) = 5 \rightarrow$ Distance and route do not change

$d_{15} = Min(d_{15}, d_{12} + d_{25}) = Min(\infty, 3+3) = 6 \rightarrow$ Distance and route do change

$d_{31} = Min(d_{31}, d_{32} + d_{21}) = Min(4, 7+4) = 4 \rightarrow$ Distance and route do not change

$d_{34} = Min(d_{34}, d_{32} + d_{24}) = Min(1, 7+9) = 1 \rightarrow$ Distance and route do not change

$d_{35} = Min(d_{35}, d_{32} + d_{25}) = Min(4, 7+3) = 4 \rightarrow$ Distance and route do not change

At this stage, we observe changes in the distance and route between node 2 and node 4, as well as between node 3 and node 2. As a result, the matrices for this step are updated accordingly.

Loop networks

$d_{41} = Min(d_{41}, d_{42} + d_{21}) = Min(\infty, 2+4) = 6 \rightarrow$ Distance and route do change

$d_{43} = Min(d_{43}, d_{42} + d_{23}) = Min(\infty, 2+9) = 11 \rightarrow$ Distance and route do change

$d_{45} = Min(d_{45}, d_{42} + d_{25}) = Min(3, 2+3) = 3 \rightarrow$ Distance and route do not change

$d_{51} = Min(d_{51}, d_{52} + d_{21}) = Min(\infty, 2+4) = 6 \rightarrow$ Distance and route do change

$d_{53} = Min(d_{53}, d_{52} + d_{23}) = Min(5, 2+9) = 5 \rightarrow$ Distance and route do not change

$d_{54} = Min(d_{54}, d_{52} + d_{24}) = Min(\infty, 2+9) = 11 \rightarrow$ Distance and route do change

Loop networks

$$D_2 = \begin{bmatrix} 0 & 3 & 5 & 5 & 6 \\ 4 & 0 & 9 & 9 & 3 \\ 4 & 7 & 0 & 1 & 4 \\ 6 & 2 & 11 & 0 & 3 \\ 6 & 2 & 5 & 11 & 0 \end{bmatrix} \qquad P_2 = \begin{bmatrix} - & 2 & 3 & 4 & 2 \\ 1 & - & 1 & 1 & 5 \\ 1 & 1 & - & 4 & 5 \\ 2 & 2 & 2 & - & 5 \\ 2 & 2 & 3 & 2 & - \end{bmatrix}$$

In this step, we observe changes in the distance and route between node 1 and node 5, node 4 and node 1, node 4 and node 3, node 5 and node 1, and node 5 and node 4. As a result, the matrices for step 2 can be updated as follows:

Loop networks

Stage 3 (intermediary node 3)

$d_{12} = Min(d_{12}, d_{13} + d_{32}) = Min(3, 5+7) = 3 \rightarrow$ Distance and route do not change

$d_{14} = Min(d_{14}, d_{13} + d_{34}) = Min(5, 5+1) = 5 \rightarrow$ Distance and route do not change

$d_{15} = Min(d_{15}, d_{13} + d_{35}) = Min(6, 5+4) = 6 \rightarrow$ Distance and route do not change

$d_{21} = Min(d_{21}, d_{23} + d_{31}) = Min(4, 9+4) = 4 \rightarrow$ Distance and route do not change

$d_{24} = Min(d_{24}, d_{23} + d_{34}) = Min(9, 9+1) = 9 \rightarrow$ Distance and route do not change

$d_{25} = Min(d_{25}, d_{23} + d_{35}) = Min(3, 9+4) = 3 \rightarrow$ Distance and route do not change

$d_{41} = Min(d_{41}, d_{43} + d_{31}) = Min(6, 11+4) = 6 \rightarrow$ Distance and route do not change

Loop networks

$d_{42} = Min(d_{42}, d_{43} + d_{32}) = Min(2, 11+7) = 2 \rightarrow$ Distance and route do not change

$d_{45} = Min(d_{45}, d_{43} + d_{35}) = Min(3, 11+4) = 3 \rightarrow$ Distance and route do not change

$d_{51} = Min(d_{51}, d_{53} + d_{31}) = Min(6, 5+4) = 6 \rightarrow$ Distance and route do not change

$d_{52} = Min(d_{52}, d_{53} + d_{32}) = Min(2, 5+7) = 2 \rightarrow$ Distance and route do not change

$d_{54} = Min(d_{54}, d_{53} + d_{34}) = Min(11, 5+1) = 6 \rightarrow$ Distance and route do change

$$D_3 = \begin{bmatrix} 0 & 3 & 5 & 5 & 6 \\ 4 & 0 & 9 & 9 & 3 \\ 4 & 7 & 0 & 1 & 4 \\ 6 & 2 & 11 & 0 & 3 \\ 6 & 2 & 5 & 6 & 0 \end{bmatrix} \qquad P_3 = \begin{bmatrix} - & 2 & 3 & 4 & 2 \\ 1 & - & 1 & 1 & 5 \\ 1 & 1 & - & 4 & 5 \\ 2 & 2 & 2 & - & 5 \\ 2 & 2 & 3 & 3 & - \end{bmatrix}$$

In step 3, the only change occurs in the distance and path between node 5 and node 4. Consequently, we can derive the matrices for step 3 as follows:

Loop networks

Stage 4 (intermediary node 4)

$d_{12} = Min(d_{12}, d_{14} + d_{42}) = Min(3, 5+2) = 3 \rightarrow$ Distance and route do not change

$d_{13} = Min(d_{13}, d_{14} + d_{43}) = Min(5, 5+11) = 5 \rightarrow$ Distance and route do not change

$d_{15} = Min(d_{15}, d_{14} + d_{45}) = Min(6, 5+3) = 6 \rightarrow$ Distance and route do not change

$d_{21} = Min(d_{21}, d_{24} + d_{41}) = Min(4, 9+6) = 4 \rightarrow$ Distance and route do not change

$d_{23} = Min(d_{23}, d_{24} + d_{43}) = Min(9, 9+11) = 9 \rightarrow$ Distance and route do not change

Loop networks

$d_{25} = Min(d_{25}, d_{24} + d_{45}) = Min(3, 9+3) = 3 \rightarrow$ Distance and route do not change

$d_{31} = Min(d_{31}, d_{34} + d_{41}) = Min(4, 1+6) = 4 \rightarrow$ Distance and route do not change

$d_{32} = Min(d_{32}, d_{34} + d_{42}) = Min(7, 1+2) = 3 \rightarrow$ Distance and route do change

$d_{35} = Min(d_{35}, d_{34} + d_{45}) = Min(4, 1+3) = 4 \rightarrow$ Distance and route do not change

$d_{51} = Min(d_{51}, d_{54} + d_{41}) = Min(6, 6+6) = 6 \rightarrow$ Distance and route do not change

$d_{52} = Min(d_{52}, d_{54} + d_{42}) = Min(2, 6+2) = 2 \rightarrow$ Distance and route do not change

$d_{53} = Min(d_{53}, d_{54} + d_{43}) = Min(5, 6+11) = 5 \rightarrow$ Distance and route do not change

Loop networks

$$D_4 = \begin{bmatrix} 0 & 3 & 5 & 5 & 6 \\ 4 & 0 & 9 & 9 & 3 \\ 4 & 3 & 0 & 1 & 4 \\ 6 & 2 & 11 & 0 & 3 \\ 6 & 2 & 5 & 6 & 0 \end{bmatrix} \qquad P_4 = \begin{bmatrix} - & 2 & 3 & 4 & 2 \\ 1 & - & 1 & 1 & 5 \\ 1 & 4 & - & 4 & 5 \\ 2 & 2 & 2 & - & 5 \\ 2 & 2 & 3 & 3 & - \end{bmatrix}$$

www.optimizationcity.com

50

In step 4, the only modification occurs in the distance and path between node 3 and node 2. Thus, we can obtain the matrices for step 4 as follows:

Loop networks

Stage 5 (intermediary node 5)

$d_{12} = Min(d_{12}, d_{15} + d_{52}) = Min(3, 6+2) = 3 \rightarrow$ Distance and route do not change

$d_{13} = Min(d_{13}, d_{15} + d_{53}) = Min(5, 5+6) = 5 \rightarrow$ Distance and route do not change

$d_{14} = Min(d_{14}, d_{15} + d_{54}) = Min(5, 6+6) = 5 \rightarrow$ Distance and route do not change

$d_{21} = Min(d_{21}, d_{25} + d_{51}) = Min(4, 3+6) = 4 \rightarrow$ Distance and route do not change

$d_{23} = Min(d_{23}, d_{25} + d_{53}) = Min(9, 3+5) = 8 \rightarrow$ Distance and route do change

$d_{24} = Min(d_{24}, d_{25} + d_{54}) = Min(9, 6+3) = 9 \rightarrow$ Distance and route do not change

$d_{31} = Min(d_{31}, d_{35} + d_{51}) = Min(4, 4+6) = 4 \rightarrow$ Distance and route do not change

Loop networks

$d_{32} = Min(d_{32}, d_{35} + d_{52}) = Min(3, 4+2) = 3 \rightarrow$ Distance and route do not change

$d_{34} = Min(d_{34}, d_{35} + d_{54}) = Min(1, 4+6) = 1 \rightarrow$ Distance and route do not change

$d_{41} = Min(d_{41}, d_{45} + d_{51}) = Min(6, 3+6) = 6 \rightarrow$ Distance and route do not change

$d_{42} = Min(d_{42}, d_{45} + d_{52}) = Min(2, 3+2) = 2 \rightarrow$ Distance and route do not change

$d_{43} = Min(d_{43}, d_{45} + d_{53}) = Min(11, 3+5) = 8 \rightarrow$ Distance and route do change

At this stage, we observe changes in the distance and route between node 2 and node 3, as well as between node 4 and node 3. Consequently, the matrices for this step, which represent the final matrix, are as follows:

Loop networks

$$D_5 = \begin{bmatrix} 0 & 3 & 5 & 5 & 6 \\ 4 & 0 & 8 & 9 & 3 \\ 4 & 3 & 0 & 1 & 4 \\ 6 & 2 & 8 & 0 & 3 \\ 6 & 2 & 5 & 6 & 0 \end{bmatrix} \qquad P_5 = \begin{bmatrix} - & 2 & 3 & 4 & 2 \\ 1 & - & 5 & 1 & 5 \\ 1 & 4 & - & 4 & 5 \\ 2 & 2 & 5 & - & 5 \\ 2 & 2 & 3 & 3 & - \end{bmatrix}$$

The algorithm concludes, and we can obtain the shortest distance between two nodes from the D5 matrix. Likewise, we can trace the corresponding path using the P5 matrix. For instance, in matrix D5, the shortest distance between node 2 and node 4 is 9 units. By examining matrix P5, we find that the shortest path from node 2 to node 4 includes node 1, and the shortest path from node 1 to node 4 passes through node 4 itself

You may ask for more exercises to practice what I've learned.

I've got just the thing for you. Check out my website where you'll find a bunch of extra exercises to dive into. You'll discover a variety of exercises covering different subjects like math, language, and critical thinking. Take a moment to swing by my website and explore the exercises waiting for you. They're not only educational but also enjoyable. I'm confident they'll help you reinforce what you've learned and gain even more confidence. Give it a shot.

Thanks

How to find us

www.optimizationcity.com

optimizationcity@gmail.com