

Chapter 490

Shortest Route

Introduction

Given a directed network defined by nodes and arcs, this procedure finds the shortest route between two specified nodes. One example of the need for such an algorithm is to be used in a GPS device to find the shortest route between two locations.

There are specialized algorithms that can be used to solve for the shortest path among several possible paths. Taha (2011) shows that the problem can be solved using a linear programming algorithm. This is the approach we have taken in **NCSS**.

Data Structure

This procedure requires a special data format in which the rows represent the arcs (branches) between two locations (nodes). Each branch is defined by two nodes and the positive distance between them. The flow is always from the first node to the second node.

Consider the following transportation example. Each row represents a road (arc) between two cities (nodes). The cities are identified by letters and the distance in miles between the cities is the given. For this example, we have to assume that all roads are one-way from the first listed city to the second. These data are stored in the dataset *Short Path*.

Short Path Dataset

City1	City2	Distance
A	B	6.8
A	D	6.2
B	C	9.1
B	D	8.5
B	E	7.2
C	E	10.3
D	E	8.8
D	F	5.6
E	F	2.5
E	G	9.4
F	G	12.3

Linear Programming Formulation of the Shortest Path Problem

As stated earlier, we use a linear programming algorithm to solve for the shortest path. Rather than present all the equations, we show how the above example is translated into a linear programming tableau. The LP tableau for the above data is

		X_{AB}	X_{AD}	X_{BC}	X_{BD}	X_{BE}	X_{CE}	X_{DE}	X_{DF}	X_{EF}	X_{EG}	X_{FG}	RHS
Minimize	$z =$	6.8	6.2	9.1	8.5	7.2	10.3	8.8	5.6	2.5	9.4	12.3	
City													
A		1	1										= 1
B		-1		1	1	1							= 0
C				-1			1						= 0
D			-1		-1			1	1				= 0
E						-1	-1	-1		1	1		= 0
F									-1	-1		1	= 0
G											-1	-1	= -1

Example 1 – Shortest Route through Several Cities

This section presents an example of how to find the shortest route between cities A and G in the Short Path dataset. As you use this example, you need to keep in mind that all roads are one-way roads.

Short Path Dataset

City1	City2	Distance
A	B	6.8
A	D	6.2
B	C	9.1
B	D	8.5
B	E	7.2
C	E	10.3
D	E	8.8
D	F	5.6
E	F	2.5
E	G	9.4
F	G	12.3

Setup

To run this example, complete the following steps:

1 Open the Short Path example dataset

- From the File menu of the NCSS Data window, select **Open Example Data**.
- Select **Short Path** and click **OK**.

2 Specify the Shortest Router procedure options

- Find and open the **Shortest Route** procedure using the menus or the Procedure Navigator.
- The settings for this example are listed below and are stored in the **Example 1** settings file. To load these settings to the procedure window, click **Open Example Settings File** in the Help Center or File menu.

Specifications Tab

Node 1 (Starting) Column **City1**
 Node 2 (Ending) Column **City2**
 Length Column **Distance**
 Node Values Input Type **Enter custom starting and stopping values**
 Path Starting Node Value **A**
 Path Ending Node Value **G**

3 Run the procedure

- Click the **Run** button to perform the calculations and generate the output.

Shortest Path

Shortest Path

Arc (Row)	Distance (Length)	City1 (Start)	City2 (End)
1	6.8	A (Start)	B
5	7.2	B	E
10	9.4	E	G (End)
Total	23.4		

This report lists the solution by giving the arcs that form the shortest path. The starting node is identified with the word *Start*. The ending city is identified with the word *End*.

The shortest path is A, B, E, and G.

Possible Network Arcs

Possible Network Arcs

Arc (Row)	Distance (Length)	City1 (Start)	City2 (End)
1	6.8*	A (Start)	B
2	6.2	A (Start)	D
3	9.1	B	C
4	8.5	B	D
5	7.2*	B	E
6	10.3	C	E
7	8.8	D	E
8	5.6	D	F
9	2.5	E	F
10	9.4*	E	G (End)
11	12.3	F	G (End)

* The starred arcs are on the shortest path.

This report lists all the arcs from which the shortest path derived. The starting node is identified with the word *Start*. The ending node is identified with the word *End*.