

# **MN50750 - Optimisation & Spreadsheet Modelling**

Coursework Assignment II - Decision Support System  
for The Great Western Railway

Name: Ritwik Singh

Word Count: 2067 words

# CONTENTS

• Introduction.....	3
• Problem Background.....	3
• Report Aim.....	4
• Ford-Fulkerson Algorithm.....	4
• Features of the Decision Support System.....	4
• User Interface of the Decision Support System.....	5
• Output of the Five Trials.....	8
• Assumptions of the Decision Support System.....	11
• Strengths of the Decision Support System.....	12
• Limitations of the Decision Support System.....	12
• Conclusion.....	12
• Appendix.....	14

# INTRODUCTION

The Great Western Railway (GWR) is a historic institution that was founded in 1833 and is a symbol of the British railway industry's innovative spirit and rich heritage. GWR has been constructing a complex network of railways over more than 200 years, covering near about 3800 miles through the beautiful scenery of southwest and south-western England. The legacy of GWR extends beyond a many kilometres of railway track and includes the determination and creativity needed to deal with the ever-changing transportation environment.

Throughout its long journey, GWR has faced various challenges, but each one has been overcome with creative thinking and calculated planning. On the other hand, a new era of demands and complications have been brought about by contemporary trends in transportation. Notably, the emergence of demand spikes, spurred by events such as football matches, festivals, and the influx of vacationers, has presented GWR with a unique set of challenges.

For the route managers of GWR, their ability to effectively handle unexpected spikes in demand has become a critical issue. The demand for a strong decision support system has grown as the influx and outflow of passengers increase at times. It is under this framework that GWR plans to develop an advanced technology that will enable route managers to effortlessly optimise train flow by giving them actionable insights.

This report explores the strategic attempt to establish a decision support system and the complexities of GWR's difficulties. GWR aims to improve its operating efficiency and provide its customers with an unmatched travel experience by minimising the effects of the demand spikes.

# PROBLEM BACKGROUND

The Great Western Railway (GWR) has a growing number of issues in managing the complexities of rail operations due to an increasing disparity between supply and demand. The key problem is how to handle abrupt increases in passenger demand, which is a difficult undertaking for a railway network that covers large areas of southern and southwestern England. In response, GWR starts working on a decision support system that would give route managers important information. This novel approach aims to maximise train frequencies from designated origin stations to their destinations and improve the capacity of rail segments.

CEO of HoptimaL, Mrs Lorena Railsworth, sees a collaborative cooperation with GWR, anticipating the production of a bespoke mobile application to boost operating efficiency. In addition to enhancing GWR's position in the market, this action presents HoptimaL with expansion prospects. Mrs. Railsworth sees potential in this concept and launches a pilot programme aimed at 34 important GWR stations. Around these stations, an automated decision support system will be constructed to help users navigate the railway system by letting them choose their starting and finishing points. The technology will use complex algorithms to determine the most effective routes, calculate the number of passengers for each segment, and indicate regions with high traffic.

The initiative's planned network consists of 88 unique railway segments that are elaborately arranged into 44 segments that are bidirectional, highlighting the intricacy and scope of the suggested decision support system. The initiative involves using cutting-edge algorithms, like the Ford-Fulkerson approach, to improve the traveller experience, make the most use of available resources, and bolster GWR's standing as a pioneer of contemporary railway excellence.

## REPORT AIM

The purpose of the report is to construct a user-friendly decision support system for the Great Western Railway, using an advanced algorithm - the Ford-Fulkerson approach. The objectives of this system are to improve overall operating efficiency for route management, handle demand fluctuations, and optimise train timetables.

## FORD-FULKERSON ALGORITHM

The Ford-Fulkerson algorithm (Appendix A) is used to determine a flow network's maximum flow. Until there are no more augmenting paths in the network, it works by iteratively augmenting the flow along paths from the source to the sink.

The Ford-Fulkerson algorithm can be compared to a GPS system for determining the optimal paths inside intricate networks, such as a city's road network. Imagine if every road had a maximum capacity for traffic.

1. **Getting Started:** Initially, there is a simple plan that states, "Hey, these roads can handle this much traffic." It's comparable to having a map that indicates each road's capacity.
2. **Looking for Better Paths:** At this moment, the user starts looking for wiser paths to follow. The user looks through the map for any less-travelled streets or shortcuts.
3. **Updating the Plan:** When the user discovers a better course of action, necessary revisions to the original plan are made. This involves shifting more traffic onto the faster pathways, much like when automobiles are rerouted to escape a traffic jam.
4. **Repeat the Process:** Continue this pattern of finding better routes and modifying the design until no further advancements are possible. It's similar to optimising the path for maximum effectiveness.

To put it briefly, Ford-Fulkerson functions as your own personal travel advisor, ensuring that you make the most of every road without creating traffic bottlenecks.

## FEATURES OF THE DECISION SUPPORT SYSTEM

- **User Interface Integration:** The system is seamlessly connected with a user interface (UI) to improve the user experience and give route managers a platform to easily communicate with the system.

- **Station Addition Functionality:** Contains a station addition function that gives users the ability to enter and save new station data, giving them flexibility to adjust to changes in the railway network.
- **Rail Segment Configuration:** Makes rail segment configuration easier by letting route managers choose beginning and finishing stations and daily capacity for each section.
- **Dynamic Link Capacity Adjustment:** Includes the capability to dynamically modify each rail segment's capacity, giving route managers the freedom to optimise for different demand scenarios.
- **Submit Button for Data Processing:** Includes a Submit button in the user interface, allowing the processing of submitted data and triggering the addition of new stations or adjustments to existing configurations.
- **Data Restoration Capability:** Provides a data restoration function for backing up and recovering data, guaranteeing data persistence against mistakes or disturbances.

These features collectively contribute to the functionality and usability of the decision support system, empowering route managers to efficiently manage train schedules and address operational challenges within the Great Western Railway network.

## USER INTERFACE OF THE DECISION SUPPORT SYSTEM

The following buttons are added to the excel sheet to enhance the user-friendly behaviour of the decision support system:

- **Enter Station Names:**  
Goal: Enables users to add or modify railway station names.  
Functionality: Opens a form where users can enter or modify station names.

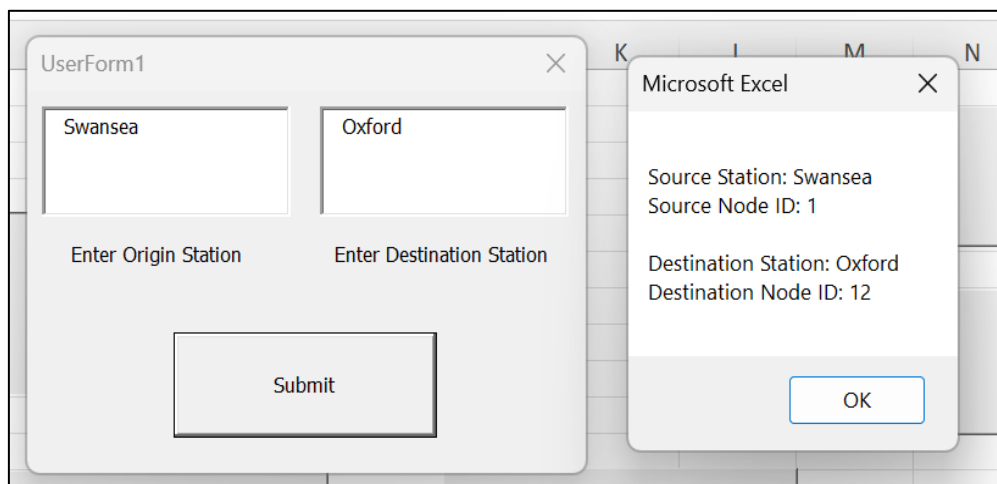


Fig 1: “Enter Station Names” pop up

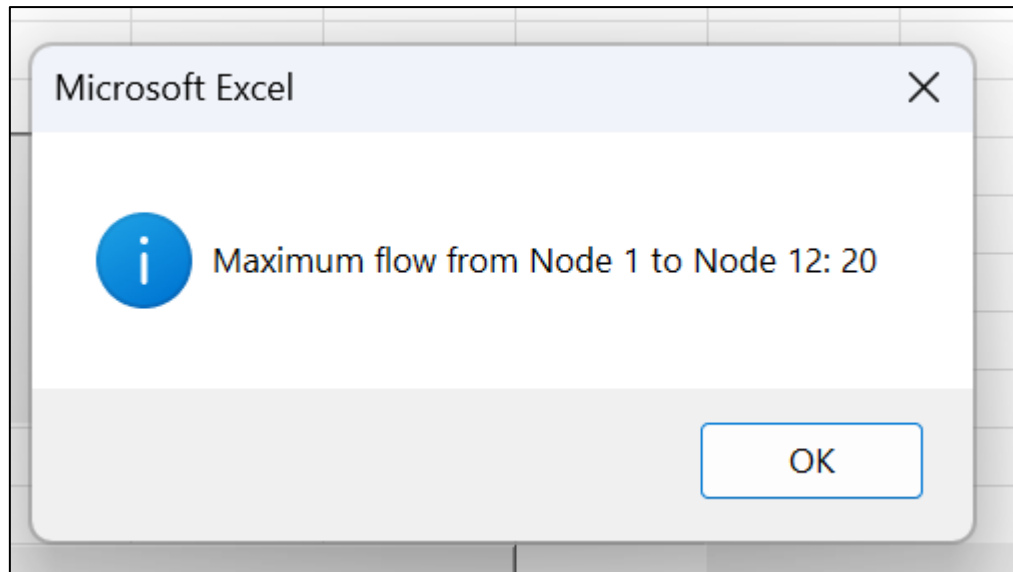


Fig 2: Pop up which shows The Maximum Flow between the two stations

- **Add New Stations:**

Goal: Facilitates the addition of new railway stations to the system.

Functionality: Clicking it will make a form to popup and ask user to add the name of the new station.

 A screenshot of a 'UserForm2' dialog box. The title bar says 'UserForm2'. The main area has the text 'Enter the Name of the New Station'. Below this text is a text input field containing the word 'Warwick'. At the bottom center of the dialog is a button labeled 'Submit'.

Fig 3: “Add New Stations” pop up

Node ID	Name
35	Warwick

Fig 4: New Station Added

- **Add New Segment:**

Goal: Makes it possible to add additional train sections that connect stations.

Functionality: When user clicks this button, a form opens, allowing them to specify information about a new rail segment.

UserForm2

Enter Origin: Swansea

Enter Destination: Warwick

Enter Capacity: 24

Submit

Fig 5: “Add New Segment” pop up

Link ID	Node 1 ID	Node 2 ID	Capacity
90	1	35	24

Fig 6: New Segment added

- **Reset Workbook:**

Goal: Removes all data and configurations by resetting the entire workbook.

Functionality: When clicked, data from all the sheets is cleared, making room for fresh inputs.

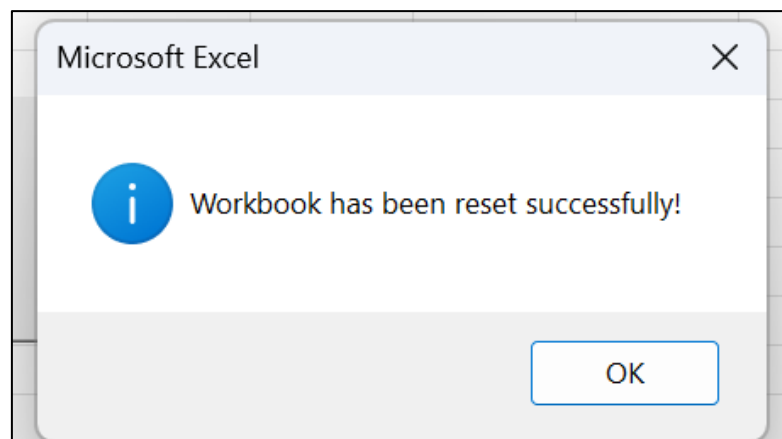


Fig 7: Prompt when the whole Workbook is reset

- **Restore Workbook Data:**

Goal: Recovers data that has been saved in the backup files, returning the workbook to its original state.

Functionality: Starts a procedure to restore data from a backup, enabling users to go back in time to a prior version of their workbook.

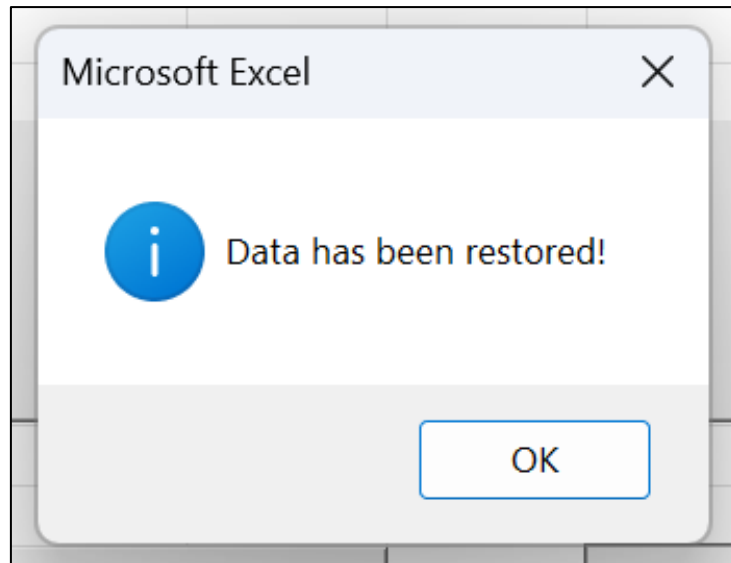


Fig 8: Prompt when the whole Workbook is restored to its original form

## OUTPUTS OF THE FIVE TRIALS

### 1. First Trial: Swansea to Cardiff Central

Link ID	Node 1 ID	Station 1	Node 2	Station 2	Capacity	Flow
9	1	Swansea	2	Cardiff Central	20	20

### 2. Second Trial: Swansea to Gatwick Airport

UserForm1

Swansea

Gatwick Airport

Enter Origin Station

Enter Destination Station

Submit

Microsoft Excel

Source Station: Swansea

Source Node ID: 1

Destination Station: Gatwick Airport

Destination Node ID: 34

OK

Link ID	Node 1 ID	Station 1	Node 2	Station 2	Capacity	Flow
	4	11 Cheltenham Spa	12	Oxford	22	20
	8	12 Oxford	14	Didcot Parkway	22	20
	9	1 Swansea	2	Cardiff Central	20	20
	10	2 Cardiff Central	3	Patchway	23	20
	11	3 Patchway	4	Bristol Parkway	16	16
	12	3 Patchway	5	Bristol Temple Meads	25	4
	13	5 Bristol Temple Meads	4	Bristol Parkway	24	4
	20	14 Didcot Parkway	15	Reading	45	20
	30	15 Reading	33	Wokingham	32	20
	32	33 Wokingham	34	Gatwick Airport	30	20
	51	4 Bristol Parkway	11	Cheltenham Spa	28	20

### 3. Third Trial: Bath Spa to London Paddington

UserForm1

Bath Spa

London Paddington

Enter Origin Station

Enter Destination Station

Submit

Microsoft Excel

Source Station: Bath Spa

Source Node ID: 6

Destination Station: London Paddington

Destination Node ID: 16

OK

Link ID	Node 1 ID	Station 1	Node 2 ID	Station 2	Capacity	Flow
4		11 Cheltenham Spa	12	Oxford	22	22
8		12 Oxford	14	Didcot Parkway	22	22
13		5 Bristol Temple Meads	4	Bristol Parkway	24	21
16		6 Bath Spa	7	Chippenham	18	18
18		7 Chippenham	13	Swindon	26	23
19		13 Swindon	14	Didcot Parkway	42	23
20		14 Didcot Parkway	15	Reading	45	45
21		15 Reading	16	London Paddington	56	51
23		6 Bath Spa	23	Westbury	12	12
34		23 Westbury	24	Newbury	16	6
35		24 Newbury	15	Reading	24	6
51		4 Bristol Parkway	11	Cheltenham Spa	28	22
58		7 Chippenham	4	Bristol Parkway	16	1
59		6 Bath Spa	5	Bristol Temple Meads	21	21
68		23 Westbury	7	Chippenham	19	6

#### 4. Fourth Trial: Brighton to Bath Spa

Link ID	Node 1 ID	Station 1	Node 2 ID	Station 2	Capacity	Flow
60		7 Chippenham	6	Bath Spa	19	9
67		23 Westbury	6	Bath Spa	12	12
68		23 Westbury	7	Chippenham	19	9
80		25 Salisbury	23	Westbury	30	21
81		26 Southampton Central	25	Salisbury	23	21
83		28 Fareham	26	Southampton Central	21	21
85		30 Portsmouth Harbour	28	Fareham	29	2
86		29 Havant	30	Portsmouth Harbour	11	2
87		31 Brighton	29	Havant	21	21
88		29 Havant	28	Fareham	19	19

#### 5. Fifth Trial: Swansea to Oxford

The screenshot shows two overlapping windows. The 'UserForm1' window has two text input fields: the first contains 'Swansea' and is labeled 'Enter Origin Station' below it; the second contains 'Oxford' and is labeled 'Enter Destination Station' below it. A 'Submit' button is centered at the bottom. The 'Microsoft Excel' window, which is semi-transparent, displays the following text: 'Source Station: Swansea', 'Source Node ID: 1', 'Destination Station: Oxford', and 'Destination Node ID: 12'. An 'OK' button is at the bottom right of the Excel window.

Link ID	Node 1 ID	Station 1	Node 2 ID	Station 2	Capacity	Flow
	4	11 Cheltenham Spa	12	Oxford	22	20
9		1 Swansea	2	Cardiff Central	20	20
10		2 Cardiff Central	3	Patchway	23	20
11		3 Patchway	4	Bristol Parkway	16	16
12		3 Patchway	5	Bristol Temple Meads	25	4
13		5 Bristol Temple Meads	4	Bristol Parkway	24	4
51		4 Bristol Parkway	11	Cheltenham Spa	28	20

## ASSUMPTIONS OF THE DECISION SUPPORT SYSTEM

Some of the assumptions that the decision support system can rely on:

- **Accurate Input Data:** It assumes that the segment capacity, demand estimates, and station names are correct and current. Decisions made based on inaccurate information may not be ideal.
- **No External Network Disruptions:** The system operates under the assumption that there won't be any severe external disruptions that could affect railway operations in a way that goes beyond normal demand fluctuations, such as large-scale events or natural disasters.
- **Consistent Network Configuration:** Throughout the optimisation process, the configuration of the railway network stays constant. Network structure changes that occur during simulations are not considered.
- **Sufficient System Uptime:** A high degree of availability and uptime is assumed by the decision support system. It assumes that the system will be up and running during key periods, and that any downtime or technical problems that would prevent the system from operating continuously are minimised.
- **Steady Technological Environment:** The decision support system functions based on the assumption of a steady technological environment. It assumes that during the system's deployment, the technological elements, software dependencies, and integration interfaces will not change. Rapid advancements in technology could call for system modifications.

## STRENGTHS OF THE DECISION SUPPORT SYSTEM

- **Feedback Mechanism:** Provides feedback to route managers through alert messages, such as confirming the addition of new stations or notifying successful data restoration.
- **User-Friendly Error Handling:** Incorporates user-friendly error handling to inform route managers about any issues or discrepancies in the data submission process.
- **Real-time Capacity Adjustment for Individual Rail Segments:** This feature allows route management to adapt their capacity in a flexible way to suit different demand circumstances.
- **Algorithmic Processing:** Makes use of techniques, including the Ford-Fulkerson algorithm, to calculate maximum flow efficiently, allowing route managers to decide on train timetables with confidence.
- **Efficient Resource Utilisation:** The system improves operational efficiency by focusing on simplifying train operations and making the most use of the resources available. By fine-tuning capacities, the system minimises bottlenecks, lowers idle times, and boosts the overall productivity of the railway network.

## LIMITATIONS OF THE DECISION SUPPORT SYSTEM

- **Sensitivity to Data Quality:** The quality of the input data has a significant impact on how accurate judgements are. Results may not be as reliable if station or segment information is inaccurate or inconsistent.
- **Single Algorithm Utilization:** The system exclusively relies on the Ford-Fulkerson algorithm. While effective for certain network flow problems, it may not address all complexities, potentially limiting its applicability in diverse scenarios.
- **Assumed User Ability:** To understand and interpret the results, the system makes certain assumptions about the user's level of competency. A decision assistance system that is not used to its full potential could be the consequence of inadequate user training.
- **Lack of Real-time Information:** The system might not be able to get real-time information that would have shown current station capacity changes or network outages. This may have an impact on how quickly the system adapts to changing circumstances.
- **External Data Integration:** The system's capacity to incorporate real-time external factors like weather and regional events that could affect railway operations may be limited by its potential for non-smooth integration of external data sources.

## CONCLUSION

The planned decision support system is a key component that might transform the way train networks are operated in the ongoing effort to modernise railway operations. Even though it is based on the difficulties the Great Western Railway (GWR) faced, the real issue is that this novel system has the ability to provide strategic insights, maximise train movement, and skilfully traverse the ever-changing transportation environment.

Developed in collaboration with HoptimaL, the Decision Support System is an example of technological progress, with the powerful Ford-Fulkerson algorithm serving as its foundation. This programme functions as a digital navigator, navigating the complicated web of railway infrastructure, like a GPS system for complex urban road networks.

The system has many advantages, including an easy-to-use interface that fits in with the workflows for route managers, dynamic link capacity changes that suit changing demand scenarios, and algorithmic processing that provides decision-makers with effective tools. But there are several obstacles in the way of implementation, such as the need to prioritise data quality, the use of a single method, and possible difficulties integrating real-time information.

A light of hope in the constantly evolving transportation world, where complexity and demand spikes are the standard, is the decision support system. This shift will see cutting-edge solutions seamlessly integrated into day-to-day operations, opening the way for a future where efficiency and passenger fulfilment exist happily on forward-moving tracks.

# APPENDIX

## APPENDIX A

### Algorithm: Ford-Fulkerson Algorithm

' Initialisation

Initialise the Flow of each link to zero and the ResCap to its capacity

Set all nodes to be non-connected and non-visited. Set all ParentNode and ConnectingLink as -1

Set the Origin node to be connected and visited

Set StopFlag = False, Current = Origin, and NumConnected = MaxFlow = 0

' Continue until there is no path with remaining capacity from the origin to destination node

while StopFlag  $\nless$  True do

Set Current as visited

' Update all neighbours of the current node as connected

for i = 1 To NumLinks do

if Node1[i] = Current And Node2[i]  $\nless$  Origin And Node2[i] is not connected And ResCap[i] > 0 then

Set Node2[i] as connected, its ParentNode as Node1[i], and its ConnectingLink as link i

NumConnected = NumConnected + 1

end if

end for

' If destination is connected, backtrack to find the links and minimum ResCap on the path from origin to destination

if Destination is connected then

Set NodeFlag = Destination, MinFlow =  $+\infty$ , and Count = 1

while NodeFlag  $\nless$  Origin do

LinksOnPath[Count] = ConnectingLink[NodeFlag] ' Add link to the path

if ResCap[LinksOnPath[Count]] < MinFlow then

MinFlow = ResCap[LinksOnPath[Count]] ' Update MinFlow if necessary

end if

NodeFlag = Parent[NodeFlag] ' Move to the previous node on the path

Count = Count + 1

end while

MaxFlow = MaxFlow + MinFlow ' Update the maximum flow from origin to destination

' Increase flows and decrease ResCap values on the path by MinFlow

for i = 1 To Count - 1 do

Flow[LinksOnPath[i]] = Flow[LinksOnPath[i]] + MinFlow

ResCap[LinksOnPath[i]] = ResCap[LinksOnPath[i]] - MinFlow

end for

' Reset the node attributes and repeat

Set all nodes to be non-connected and non-visited. Set all ParentNode and ConnectingLink as -1

NumConnected = 0

Current = Origin

' If there is a connected but non-visited node, set this as the current node and repeat

else if NumConnected > 0 then

for i = 1 To NumNodes do

if Node i is connected and non-visited then

```

Current = i and NumConnected = NumConnected - 1
exit for
end if
end for
' Else, there is no path from origin to destination; terminate the algorithm
else
StopFlag = True
end if
end while

```

## APPENDIX B

### Data Structures Used in Ford-Fulkerson Algorithm:

Table B.1: NodeInfo Structure

Field	Data Type	Description
nodeID	Long	Unique identifier for each node
stationName	String	Name of the station associated with the node
ParentNode	Long	Identifier of the parent node in the network
Connected	Boolean	Indicator of node connection status
Visited	Boolean	Indicator of whether the node has been visited
ConnectingLink	Long	Identifier of the connecting link in the network

Table B.2: LinkInfo Structure

Field	Data Type	Description
linkID	Long	Unique identifier for each link in the network
node1ID	Long	Identifier of the first node connected by link
node2ID	Long	Identifier of the second node connected by link
capacity	Long	Maximum capacity of the link in the network
flow	Long	Current flow through the link
Rescap	Long	Residual capacity of the link

Table B.3: Variables in FordFulkerson Subroutine

Variable	Data Type	Description
NodeFlag	Long	Temporary variable used in algorithm for node tracking
MinFlow	Long	Minimum flow on a path during algorithm execution
MaxFlow	Long	Maximum flow from origin to destination

Current	Long	Identifier of the current node in the network
NumConnected	Long	Number of connected nodes in the network
StopFlag	Boolean	Flag indicating termination of the algorithm
Nodes()	NodeInfo	Array storing information about nodes in the network
Links()	LinkInfo	Array storing information about links in the network
Count	Long	Counter variable used in algorithm for iteration
i	Long	Loop iterator variable