Assignment 2: Learning Network Flow Optimization using Excel and GLPK Solvers

Prepared by **[Jay Przybyla](mailto:jjpriz@gmail.com) and [Xuesong Zhou](mailto:xzhou99@gmail.com)**

**Learning Objectives:**

1. Learn to use a transportation network with node-link structure for network flow optimization
2. Understand the three sets of constraints for network flow optimization:
   1. Flow balance constraint
   2. Demand/Supply constraint
   3. Flow capacity constraint.
3. Understand mathematical programming models for shortest path problems, minimum cost flow problems with capacity constraints, and maximum flow problems.
4. Solve the three above mentioned problems using Excel Solver.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Related files and package:**

1. You can [create](https://docs.google.com/document/d/19a757G2A-yhz4pzT4gh1fn6turFJlOlEKEL7X5B2n1U/edit) or [import](https://docs.google.com/document/d/1nHKOo8buhCReS41IUFAdzhXIAHmijU1TfdR98i-HQk8/edit)  or load an existing network from NEXTA.

2. You can find Excel file with network flow optimization code under

<https://github.com/xzhou99/learning-transportation/tree/master/GAMS_code%20-space-time-network>

network flow optimization.xlsx

3. You can install open-source Linear Programming Kit ([GPLK](http://www.gnu.org/software/glpk/)) and its GUI at <http://sourceforge.net/projects/gusek/> or <http://glpklabw.sourceforge.net>

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Task 1: Become Familiar with Graph and Network Terminology / Understand Node – Link Structure*

A **graph** G = (N,A) is a finite nonempty set of **nodes** (or vertices) and a set of node pairs A called **arcs** (or links or edges). n=│N│, m=│A│ where *n* is the number of nodes and *m* is the number of arcs. The **degree** of a node is the number of incident arcs. Indegree and outdegree of a node.

A **walk** is a sequence of nodes (n1, n2, ...,nk) in which each adjacent node pair is an arc.

A **path** is a walk with no repeated nodes. A **shortest path** is the shortest or least cost path from a **source** node (origin) to a **destination** node.

A **tour** is similar to a path, but a tour is closed path, meaning that the first node and the final node on the path are the same node on the network. Formally, a **cycle** is a walk (n1, n2,...,nk) with n1 = nk, k≥3, and with no repeated nodes except n1 = nk.

A graph is **connected** if a path exists between each pair of nodes. An unconnected graph can be separated into two or more connected components.

A **cut** is a partition of the node set N into two parts, s and =N-s. Each cut defines a set of arcs consisting of those arcs that have one endpoint in s and another endpoint in . We refer to this set of arcs as a cut and represent it by the notation [s, ].

A **tree** is an acyclic connected graph. A **tree** is an acyclic connected graph.

The number of arcs in a tree is always one less than the number of nodes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**PROBLEM (1) Node – Link Structure**

Locate input excel sample file under NEXTA Internal\_release\importing\_sample\_data\_sets\sample\_data\_set.xls

Use NEXTA32.exe and click on Menu -> File -> Import-> Single Excel file Load sample spreadsheet, click on import all table button, show the network.

1. Show the link length in miles for link (1->2)
2. Check input\_link.csv file, confirm that they are consistent with the input.
3. Manually find the shortest path from node 1 to node 4.
4. Construct a walk, a tour, a tree and a cut starting from node 1 on the 6-node network.
5. Are link sets (1->3), (5->6) and (6->4) a cut in this network?

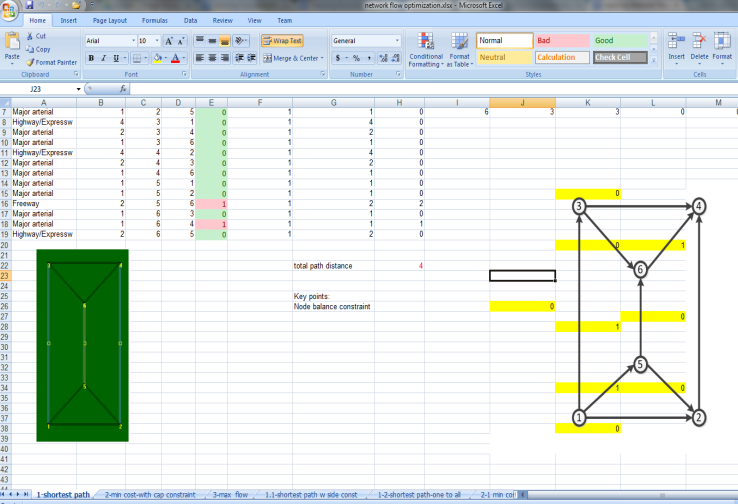
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Task 2: Finding the Shortest Path*

## Overview: Shortest Path Problems in Transportation Models

*Reference file: NEXTA release\ sample\_data\_sets\Optimization\ network flow optimization.xlsx*

*Sheet 1: shortest path*



Given a transportation network, the shortest path problem aims to determine a minimal travel time path from any node to any other node ci,j: distance from node r to node s

There are many possible extensions from the classical shortest path problem by extending either the link cost function or the network structure.

Extensions

* Dynamic (i.e. time-dependent) shortest path (travel time is dependent on leaving time t: ci,j(t))
* Stochastic shortest path (uncertainty in estimates of travel time: ci,j+ historical information, prediction)
* Multi-objective shortest path (travel time vs. travel cost, travel time vs. risk)
* Turn penalties and prohibitions
* Transit frequencies and expected waiting time

Reference: [Stefano Pallottino and Maria Grazia Scutella.](http://citeseer.ist.psu.edu/pallottino98shortest.html) *[Shortest path algorithms in transportation models: classical and innovative aspects](http://citeseer.ist.psu.edu/pallottino98shortest.html)*[. In P. Marcotte and S. Nguyen, editors, Equilibrium and Advanced Transportation Modelling, pages 245--281. Kluwer, 1998.](http://citeseer.ist.psu.edu/pallottino98shortest.html)

<http://citeseer.ist.psu.edu/pallottino98shortest.html>

|  |
| --- |
| Optimization problems  Node Index: i,j; link index (i,j).  Variables *:* xij = 1 if link (i,j) is selected in the shortest path.  Min z=∑ij cij x ij  Subject to  Flow balance constraints:  ∑j x rj = 1 for origin r  ∑i x is = -1for destination s  ∑i x ij - ∑k x jk = 0for non-source non-destination node j  x ij≥0 |

We now try to find the shortest path on the six-node network from node 1 to node 4. The steps of constructing an optimization model are

1. setup node-link structure, based on input\_link.csv from NEXTA
2. setup link cost C,
3. setup link variable vector X,
4. define objective function Z,
5. define flow balance constraint using SUMIF function
6. solve the problem using EXCEL Solver

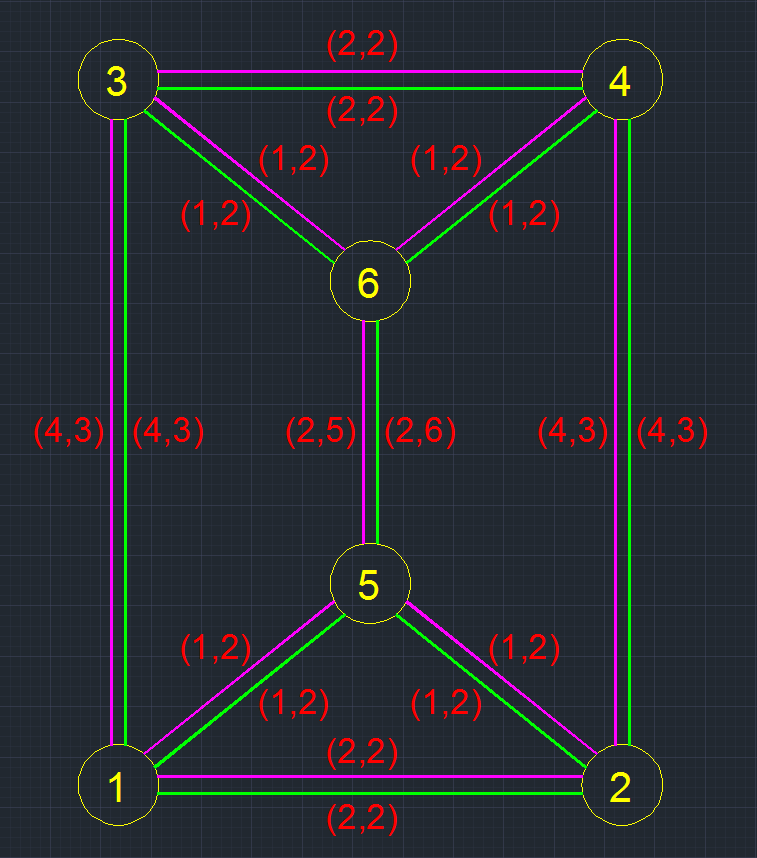
Reference on using Excel Solver: <http://office.microsoft.com/en-us/excel-help/introduction-to-optimization-with-the-excel-solver-tool-HA001124595.aspx>

**Step 1: Create Link Block in Excel**

Step 1.1: Set up the first column in excel to represent the link type.

Step 1.2: Set up the second column in excel to represent the link cost (distance, risk, etc).

Step 1.3: Set up the third and fourth columns in excel to represent FROM node TO node pairs. The third column should represent the beginning node and column four should represent the ending node for a given link direction. The results of the given 6 cell network with the cost and capacity for each link presented in the following format: (cost, capacity):



should look like:

|  |  |  |  |
| --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** |
| Major arterial | 2 | 1 | 2 |
| Highway/Expressway | 4 | 1 | 3 |
| Major arterial | 1 | 1 | 5 |
| Major arterial | 2 | 2 | 1 |
| Highway/Expressway | 4 | 2 | 4 |
| Major arterial | 1 | 2 | 5 |
| Highway/Expressway | 4 | 3 | 1 |
| Major arterial | 2 | 3 | 4 |
| Major arterial | 1 | 3 | 6 |
| Highway/Expressway | 4 | 4 | 2 |
| Major arterial | 2 | 4 | 3 |
| Major arterial | 1 | 4 | 6 |
| Major arterial | 1 | 5 | 1 |
| Major arterial | 1 | 5 | 2 |
| Freeway | 2 | 5 | 6 |
| Major arterial | 1 | 6 | 3 |
| Major arterial | 1 | 6 | 4 |
| Highway/Expressway | 2 | 6 | 5 |

Step 1.4: Set up the fifth column in excel to represent the flow on each link. This is the decision variable for the linear programming in excel therefore you can put “0” as a placeholder in each cell for now. It is helpful to color code this block of cells to remind you they are the decision variables. It is also helpful to conditionally format this block (under Home ribbon under Styles) to make values greater or equal to 1 a different color so they stand out. The updated table should look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** | **Flow** |
| Major arterial | 2 | 1 | 2 | 0 |
| Highway/Expressway | 4 | 1 | 3 | 0 |
| Major arterial | 1 | 1 | 5 | 0 |
| Major arterial | 2 | 2 | 1 | 0 |
| Highway/Expressway | 4 | 2 | 4 | 0 |
| Major arterial | 1 | 2 | 5 | 0 |
| Highway/Expressway | 4 | 3 | 1 | 0 |
| Major arterial | 2 | 3 | 4 | 0 |
| Major arterial | 1 | 3 | 6 | 0 |
| Highway/Expressway | 4 | 4 | 2 | 0 |
| Major arterial | 2 | 4 | 3 | 0 |
| Major arterial | 1 | 4 | 6 | 0 |
| Major arterial | 1 | 5 | 1 | 0 |
| Major arterial | 1 | 5 | 2 | 0 |
| Freeway | 2 | 5 | 6 | 0 |
| Major arterial | 1 | 6 | 3 | 0 |
| Major arterial | 1 | 6 | 4 | 0 |
| Highway/Expressway | 2 | 6 | 5 | 0 |

Step 1.5: Set up the sixth column in excel to represent the SUMIF operator indicator. This will be used later, and should be filled with “1” for each row.

Step 1.6: Set up the seventh column in excel to represent the loaded link cost. For each unit of flow that transverses the link, there is the associated cost. The loaded link cost then is the cost of the link (column 2) multiplied by the flow on that link (column 5). Copy this formula down for all links in the network. The new table should look like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** | **Flow** | **SUMIF Ind.** | **link cost** |
| Major arterial | 2 | 1 | 2 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 1 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 1 | 5 | 0 | 1 | 0 |
| Major arterial | 2 | 2 | 1 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 2 | 4 | 0 | 1 | 0 |
| Major arterial | 1 | 2 | 5 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 3 | 1 | 0 | 1 | 0 |
| Major arterial | 2 | 3 | 4 | 0 | 1 | 0 |
| Major arterial | 1 | 3 | 6 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 4 | 2 | 0 | 1 | 0 |
| Major arterial | 2 | 4 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 4 | 6 | 0 | 1 | 0 |
| Major arterial | 1 | 5 | 1 | 0 | 1 | 0 |
| Major arterial | 1 | 5 | 2 | 0 | 1 | 0 |
| Freeway | 2 | 5 | 6 | 0 | 1 | 0 |
| Major arterial | 1 | 6 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 6 | 4 | 0 | 1 | 0 |
| Highway/Expressway | 2 | 6 | 5 | 0 | 1 | 0 |

Since we placed “0” for the place holders for the Flow, the associated link costs are also 0. These should change as the Flow column is changed from 0 to 1. Test this by changing a few of the Flow variables from 0 to 1 and verify that the link costs changes. The new table should look like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** | **Flow** | **SUMIF Ind.** | **link cost** |
| Major arterial | 2 | 1 | 2 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 1 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 1 | 5 | 0 | 1 | 0 |
| Major arterial | 2 | 2 | 1 | 1 | 1 | 2 |
| Highway/Expressway | 4 | 2 | 4 | 0 | 1 | 0 |
| Major arterial | 1 | 2 | 5 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 3 | 1 | 1 | 1 | 4 |
| Major arterial | 2 | 3 | 4 | 0 | 1 | 0 |
| Major arterial | 1 | 3 | 6 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 4 | 2 | 0 | 1 | 0 |
| Major arterial | 2 | 4 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 4 | 6 | 1 | 1 | 1 |
| Major arterial | 1 | 5 | 1 | 0 | 1 | 0 |
| Major arterial | 1 | 5 | 2 | 0 | 1 | 0 |
| Freeway | 2 | 5 | 6 | 1 | 1 | 2 |
| Major arterial | 1 | 6 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 6 | 4 | 0 | 1 | 0 |
| Highway/Expressway | 2 | 6 | 5 | 0 | 1 | 0 |

Step 1.7: Construct the cost function by summing all the individual link costs in a single cell. This cell will be the target cell that will be used in the excel solver. This step completes the link block. The table, with the same Flow values changed in Step 1.6, should look like:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** | **Flow** | **SUMIF Ind.** | **link cost** |
| Major arterial | 2 | 1 | 2 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 1 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 1 | 5 | 0 | 1 | 0 |
| Major arterial | 2 | 2 | 1 | 1 | 1 | 2 |
| Highway/Expressway | 4 | 2 | 4 | 0 | 1 | 0 |
| Major arterial | 1 | 2 | 5 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 3 | 1 | 1 | 1 | 4 |
| Major arterial | 2 | 3 | 4 | 0 | 1 | 0 |
| Major arterial | 1 | 3 | 6 | 0 | 1 | 0 |
| Highway/Expressway | 4 | 4 | 2 | 0 | 1 | 0 |
| Major arterial | 2 | 4 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 4 | 6 | 1 | 1 | 1 |
| Major arterial | 1 | 5 | 1 | 0 | 1 | 0 |
| Major arterial | 1 | 5 | 2 | 0 | 1 | 0 |
| Freeway | 2 | 5 | 6 | 1 | 1 | 2 |
| Major arterial | 1 | 6 | 3 | 0 | 1 | 0 |
| Major arterial | 1 | 6 | 4 | 0 | 1 | 0 |
| Highway/Expressway | 2 | 6 | 5 | 0 | 1 | 0 |
|  |  |  |  |  |  |  |
|  |  |  |  | Cost | Function | 9 |

**Step 2: Create Node Block in Excel**

Step 2.1: Set up the eighth column in excel to represent the node identification number. This should match the number of nodes your network has. Column 8 should look like:

|  |
| --- |
| **Node ID** |
| 1 |
| 2 |
| 3 |
| 4 |
| 5 |
| 6 |

Step 2.2: Set up the ninth column in excel to represent the number of outbound links from each node. This step is accomplished using the “SUMIF” operator within excel. The SUMIF command requires 3 inputs separated by commas. The command inputs are as follows: SUMIF (A Given Set of Data, the criteria, What Value to add for each Correct IF).

Note: For the criteria parameter in the SUMIF command, we use the following formula:

“=”&H2 The “=” portion converts the text cell into a number and the & is a reference command.

For the first node, the corresponding Outbound link formula should look like:

=SUMIF($C$2:$C$19,”=”&H2,$F$2:$F$19)

The first input is C2:C19 which represents the set of data included in the From Node column. The second input, H2, is the criteria or in this case we are looking for all links that start from node 1, the thirds input is F2:F19, which is the SUMIF Ind. Column. This means that for each row where the From Node is 1, the operator will add 1.

Step 2.3 Set up the tenth column in excel to represent the number of inbound links from each node. Follow the same steps as in Step 2.2, but use the To Node column as the given set of data.

The Node table should look like:

|  |  |  |
| --- | --- | --- |
| **Node ID** | **Outbound Links** | **Inbound Links** |
| 1 | 3 | 3 |
| 2 | 3 | 3 |
| 3 | 3 | 3 |
| 4 | 3 | 3 |
| 5 | 3 | 3 |
| 6 | 3 | 3 |

Step 2.4: Set up the eleventh column in excel to represent the network demand. The number of units to go from one node to another is indicated by placing those numbers by the respective nodes. A positive number indicates the origination and a negative number, the destination. For example, If there was to be one unit of flow to pass from node 1 to node 4, then a positive one would be placed for node 1 and a negative 1 placed for node 4. The corresponding node table should look like:

|  |  |  |  |
| --- | --- | --- | --- |
| **Node ID** | **Outbound Links** | **Inbound Links** | **Demand** |
| 1 | 3 | 3 | 1 |
| 2 | 3 | 3 | 0 |
| 3 | 3 | 3 | 0 |
| 4 | 3 | 3 | -1 |
| 5 | 3 | 3 | 0 |
| 6 | 3 | 3 | 0 |

Step 2.5: Set up the twelfth column in excel to represent the network flow balance. The flow balance for each node is equal to the total Flow on the outbound links – the total flow on the inbound links – the demand for that node. For the first node, the corresponding Flow Balance formulas should look like:

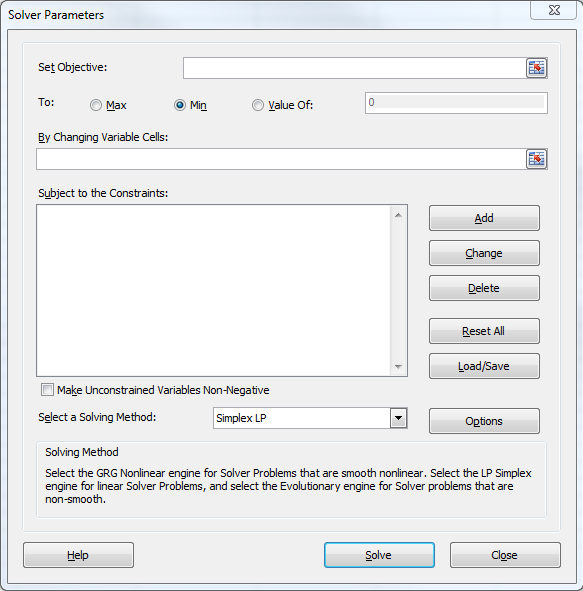
=(SUMIF($C$2:$C$19,”=”&H2,$E$2:$E$19) – SUMIF($D$2:$D$19,”=”&H2,$E$2:$E$19))-K2

The corresponding node table should look like:

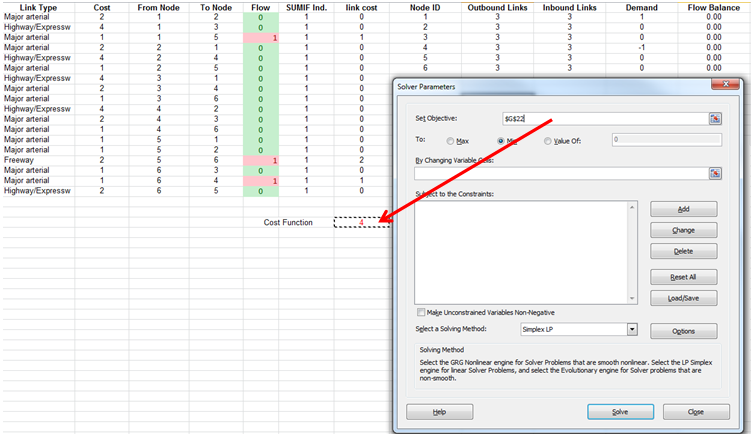
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Node ID** | **Outbound Links** | **Inbound Links** | **Demand** | **Flow Balance** |
| 1 | 3 | 3 | 1 | 0.00 |
| 2 | 3 | 3 | 0 | 0.00 |
| 3 | 3 | 3 | 0 | 0.00 |
| 4 | 3 | 3 | -1 | 0.00 |
| 5 | 3 | 3 | 0 | 0.00 |
| 6 | 3 | 3 | 0 | 0.00 |

**Step 3: Use Excel Solver to Find Optimal Solution**

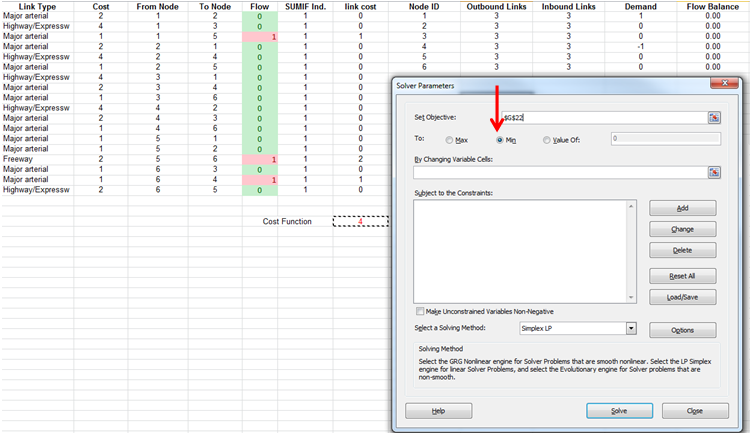
Step 3.1: Now that the link and node tables are organized, we can use the excel solver to find the optimal shortest path. Under the Data ribbon, click on the Solver button. (If the Solver does not appear, it will need to be added as an Excel add-on.) The solver interface will appear:



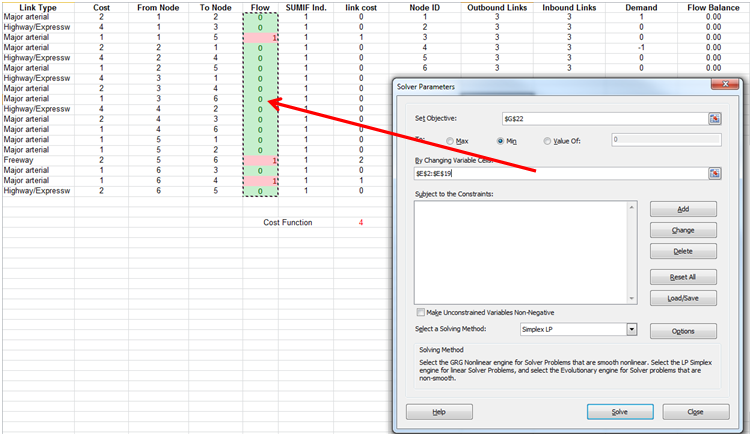
Step 3.2: For the “Set Objective” option, choose the Cost Function cell.



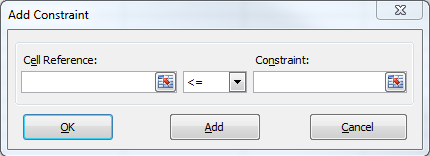
Step 3.3: Select “To: Min” - Meaning we want to minimize the Cost Function.



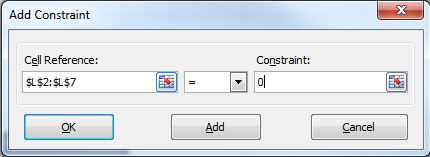
Step 3.4: For the “By Changing Variable Cells” option, choose the Flow column.



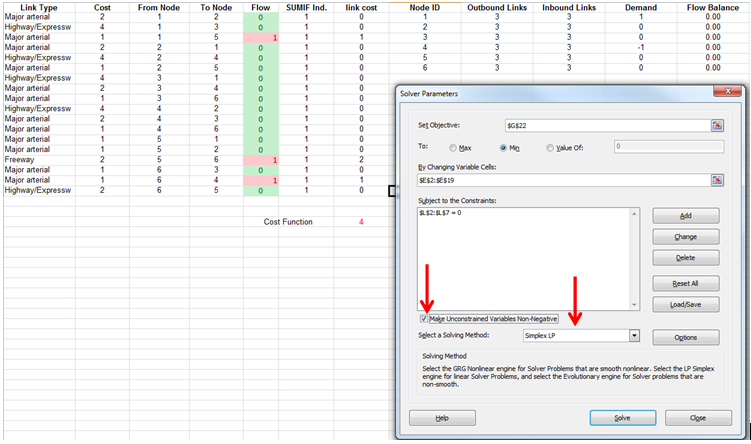
Step 3.5: For the “Subject to the Constraints” option, select “Add” and the following window will appear.



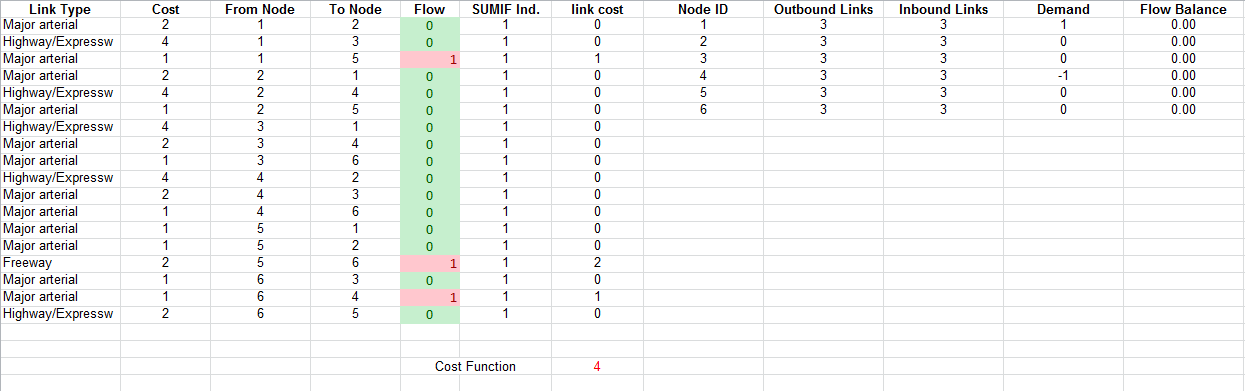
For “Cell Reference” choose the Flow Balance column. Choose that it is Equal To. For the “Constraint” choose 0.



Step 3.6: Be sure “Make Unconstrained Variables Non-Negative” and “Simplex LP” are Marked and Selected, respectively.



Step 3.7: Press solve and the shortest path solution will be shown by the indication of 1’s instead of 0’s in the Flow column. The optimal solution for this network is shown below:



**PROBLEM (2) Safe Hazardous Material Transportation**

The [transportation of hazardous materials](http://phmsa.dot.gov/hazmat) is an important research topic due to the associated risks and impacts. There are a number of ways to reduce the risk of transportation hazardous material, such as driver training or better transportation route planning. This problem is interested in designing the safest route to ship hazardous materials on a transportation network.

For a hazmat shipment, we need to find a safe path from an origin to a destination that involves the smallest probability of collision. A path has a sequence of links, where driving on this path can be viewed as a probabilistic experiment, and the probability of being involved in a collision during one trip is equivalent to the opposite of the probability of no collision occurring for the duration of the trip. Translating this relationship into an equation produces the following function:



where P(crash) is the probability of collision per unit exposure, α is the measure of exposure (time or mileage), and P(crash)α is the total probability of collision due to exposure

Please check statistics of crashes per million vehicle miles traveled at:

<http://www.mhd.state.ma.us/default.asp?pgid=content/traffic/crashrate&sid=about>

Can you find the safest path on this 6-node network?

Hints : (1) Decompose links to 0.1-mile or 0.5-mile segment (on edge *e)*  to rebuild the network.

(2) The objective function can be transformed to  and further converted to sum of log(risk(e)) through a log transformation.

(3) Use log(risk(e)) as the edge cost function to solve the problem.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*Task 3: Minimum Cost Flow with Capacity Constraints*

Reference file: NEXTA release\ sample\_data\_sets\Optimization\ network flow optimization.xlsx

Sheet 2: min cost-with cap constraint

|  |
| --- |
| Optimization problems  Node Index: i,j; link index (i,j).  Flow demand d  Variables *:* xij = 1 if link (i,j) is selected in the shortest path.  Min z=∑ij cij x ij  Subject to  Flow balance constraints:  ∑j x rj = d for origin r  ∑i x is = -dfor destination s  ∑i x ij - ∑k x jk = 0for non-source non-destination node j  x ij capacity(i,j)  x ij≥0  Reference: http://en.wikipedia.org/wiki/Minimum-cost\_flow\_problem |

This lesson will be similar to the Shortest Path Problem, except that we are going to add Capacity Constraints for each link. Assume the following capacities for each link:

|  |  |  |
| --- | --- | --- |
| **From Node** | **To Node** | **Capacity** |
| 1 | 2 | 2 |
| 1 | 3 | 3 |
| 1 | 5 | 2 |
| 2 | 1 | 2 |
| 2 | 4 | 3 |
| 2 | 5 | 2 |
| 3 | 1 | 3 |
| 3 | 4 | 2 |
| 3 | 6 | 2 |
| 4 | 2 | 3 |
| 4 | 3 | 2 |
| 4 | 6 | 2 |
| 5 | 1 | 2 |
| 5 | 2 | 2 |
| 5 | 6 | 6 |
| 6 | 3 | 2 |
| 6 | 4 | 2 |
| 6 | 5 | 5 |

**Step 1: Create Link Block in Excel**

Step 1.1: Follow all the steps in Lesson 1, except with the addition of the capacity column (from above) at the end of the link block. The link block should look like:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** | **Flow** | **SUMIF Ind.** | **link cost** | **Capacity** |
| Major arterial | 2 | 1 | 2 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 1 | 3 | 0 | 1 | 0 | 3 |
| Major arterial | 1 | 1 | 5 | 1 | 1 | 1 | 2 |
| Major arterial | 2 | 2 | 1 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 2 | 4 | 0 | 1 | 0 | 3 |
| Major arterial | 1 | 2 | 5 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 3 | 1 | 0 | 1 | 0 | 3 |
| Major arterial | 2 | 3 | 4 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 3 | 6 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 4 | 2 | 0 | 1 | 0 | 3 |
| Major arterial | 2 | 4 | 3 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 4 | 6 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 5 | 1 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 5 | 2 | 0 | 1 | 0 | 2 |
| Freeway | 2 | 5 | 6 | 1 | 1 | 2 | 6 |
| Major arterial | 1 | 6 | 3 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 6 | 4 | 1 | 1 | 1 | 2 |
| Highway/Expressway | 2 | 6 | 5 | 0 | 1 | 0 | 5 |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  | Cost Function | 4 |  |  |

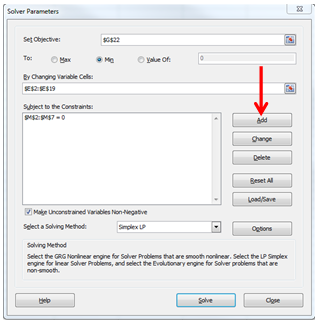
**Step 2: Create Node Block in Excel**

Step 2.1: The node block for this lesson should be the same as in lesson 1. Repeat the steps or copy and paste. The node block should look like:

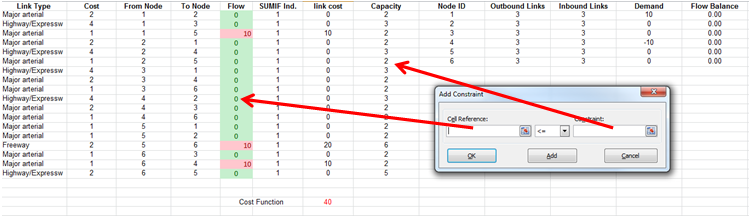
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Node ID** | **Outbound Links** | **Inbound Links** | **Demand** | **Flow Balance** |
| 1 | 3 | 3 | 1 | 0.00 |
| 2 | 3 | 3 | 0 | 0.00 |
| 3 | 3 | 3 | 0 | 0.00 |
| 4 | 3 | 3 | -1 | 0.00 |
| 5 | 3 | 3 | 0 | 0.00 |
| 6 | 3 | 3 | 0 | 0.00 |

**Step 3: Use Excel Solver to Find Optimal Solution**

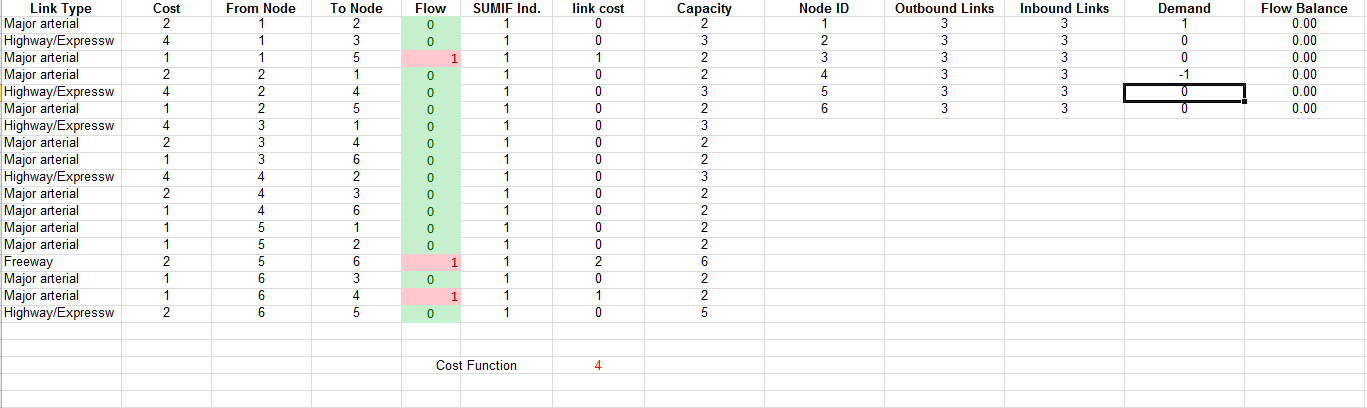
Step 3.1: The solver will be used the same was as in lesson 1, except with the addition of the link capacity constraint. With the solver window open, and the same selections as in lesson 1, click on the “Add” button under “Subject to the Constraints.”



Step 3.2: The Add Constraint window will open. For Cell Reference, choose the Flow column. The operator should be “less than or equal to.” For the Constraint, choose the Capacity column. This is indicating that the optimization will be subject to the Flow for each link being less than or equal to the capacity for that link.



Step 3.3: Press solve and the shortest path solution will be shown by the indication of 1’s instead of 0’s in the Flow column. The optimal solution for this network is shown below:



Note: The solution did not change from lesson 1 because the capacity constraint was not violated in that solution. You can adjust the demand and observe how the solutions change when the capacity constraint is included and when it is not.

**PROBLEM (3) The Transshipment Problem on 6-Node Network**

Utilizing the same 6-node network as before, a transshipment problem can be analyzed. This problem builds off the same principles as the shortest path problem, but included multiple origins and destinations. This is handled by changing the Flow Balance column in excel to match the given supply and demands. Travel time is used as the link cost and objective function. Since the given costs are assumed to be travel times for vehicles, the problems asks to increase that travel time 25% to account for the slower moving trucks.

Assume the following given inputs:

There are plants (P) with supply s(i) on nodes 1 and 2; s(1) = 3; s(2) = 3;

There are warehouses (Q) with demand d(j) on nodes 3 and 4; d(3)=4; d(4) = 2;

The travel time for the trucks is calculated by using 1.25% of the given link costs.

Include the given link capacity constraints.

Objective:

The objective is to find the cheapest shipping plan to satisfy all the demands.

Requirements:

1. Modify a min cost flow model to construct an optimization model for the transshipment problem.
2. Solve the problem using Excel Solver
3. **[unequal supply and demand]** Can you find the solution from Excel if the supply on node 4 is increased to 2.5?
4. **[link blockage]** Keeping the supply on node 4 = 2, what happens if the capacity on link (1-3) is reduced to 2?

**PROBLEM (4) Optimal Delivery Problem on 6-Node Network**

Utilizing the same 6-node network as before, a delivery optimization problem can be solved. This problem also builds off the shortest path problem. In this scenario, we assume a vehicle needs to travel from an origin to a destination with several stops requested within the network. The goal for this problem is to make all the stops while minimizing the costs.

Assume the following given inputs:

A truck needs to go from node 1 to node 4.

There are delivery requests at nodes 2, 3 and 4.

The travel time for the truck is the given link costs.

Objective:

The objective is to find a delivery plan to minimize the total driving cost while satisfying all requests.

Requirement:

1. Modify a min cost flow model to construct an optimization model for the transportation problem
2. Solve the problem using Excel Solver

**PROBLEM (5) Most Reliable Path on 6-Node Network**

Utilizing the same 6-node network as before, we can expand the problem to include probabilistic link path reliability. In reality, network links are subject to incidents which can affect the reliability of the link travel time. For instance, a traffic collision, weather, reoccurring congestion, etc., can cause a link to experience longer than usual travel times. This problem asks you to include a link path reliability into the shortest path problems from above.

Assume the following given inputs:

Let Pij be the probability that a link is working, and that all link are independent.

The probability that a path P is working is product of working probability over all links along a path.

Let Pij = # of lanes/5 for each link

Objective:

The objective is to find the shortest and most reliable path.

Requirement:

Construct a shortest path problem to find the most reliable path.

Hints: -->

-->

Use as C(i,j)

**PROBLEM (6) Chinese Postman Problem**

Utilizing the same 6-node network as before, we can expand the problem to address the Chinese Postman or Route Inspection Problem. In this problem, the goal is to have the postman visit every link in a network at least one while minimizing the distance traveled. Extending the shortest path problem, this can be determined using the optimization tools. In this problem, we assume a patrol vehicle needs to travel on each link at least once during a round while minimizing the total distance traveled.

Assume the following given inputs:

A patrol vehicle is required to travel each link at least once for each of his rounds.

Objective:

The objective is to find a path for the patrol vehicle while minimizing the total travel distance.

Requirement:

Modify a min cost flow model to construct an optimization model for the Route Inspection Problem.

Hints:

Min z=∑ij cij x ij

Subject to

∑i x ij - ∑k x jk =b

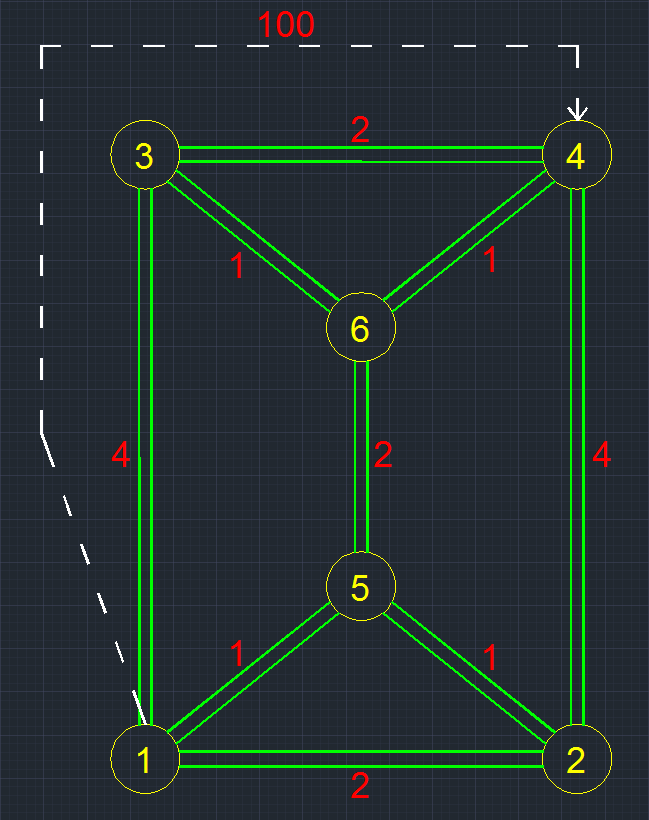
x ij j≥1

*Task 4: Maximum Flow*

Reference file: NEXTA release\ sample\_data\_sets\Optimization\ network flow optimization.xlsx

Sheet 3: max flow

This Task will utilize the same network as before. The capacity constraint will be included and will be the same as in lesson 2. To examine the total number of units that can transverse the network from node 1 to node 4 we utilize a modeling trick where a artificial, one-directional, high cost, high capacity link is introduced. The artificial link will extend from the source node to the destination node and should have cost attributes that make it undesirable when compared to the true network links. It should also have enough capacity to accommodate enough flow that we know we have saturated the network capability. For our network, we will assume the artificial arc extends from Node 1 to Node 4, has a capacity of 20, and a cost of 100. The network will look as follows with the artificial link in white:



**Step 1: Create Link Block in Excel**

Step 1.1: Follow all the steps in Lesson 1, except with the addition of the artificial arc link as a new row. The artificial link should have a cost of 100, extend from node 1 to node 4, have a SUMIF indicator of 1, and a capacity of 20. The Flow column can just have a 0 for a placeholder. The Link Flow should be calculated as the others are. The link block should look like:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Link Type** | **Cost** | **From Node** | **To Node** | **Flow** | **SUMIF Ind.** | **link cost** | **Capacity** |
| Major arterial | 2 | 1 | 2 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 1 | 3 | 0 | 1 | 0 | 3 |
| Major arterial | 1 | 1 | 5 | 0 | 1 | 0 | 2 |
| Major arterial | 2 | 2 | 1 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 2 | 4 | 0 | 1 | 0 | 3 |
| Major arterial | 1 | 2 | 5 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 3 | 1 | 0 | 1 | 0 | 3 |
| Major arterial | 2 | 3 | 4 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 3 | 6 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 4 | 4 | 2 | 0 | 1 | 0 | 3 |
| Major arterial | 2 | 4 | 3 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 4 | 6 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 5 | 1 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 5 | 2 | 0 | 1 | 0 | 2 |
| Freeway | 2 | 5 | 6 | 0 | 1 | 0 | 6 |
| Major arterial | 1 | 6 | 3 | 0 | 1 | 0 | 2 |
| Major arterial | 1 | 6 | 4 | 0 | 1 | 0 | 2 |
| Highway/Expressway | 2 | 6 | 5 | 0 | 1 | 0 | 5 |
| Artificial Arc | 100 | 1 | 4 | 0 | 1 | 0 | 20 |

**Step 2: Create Node Block in Excel**

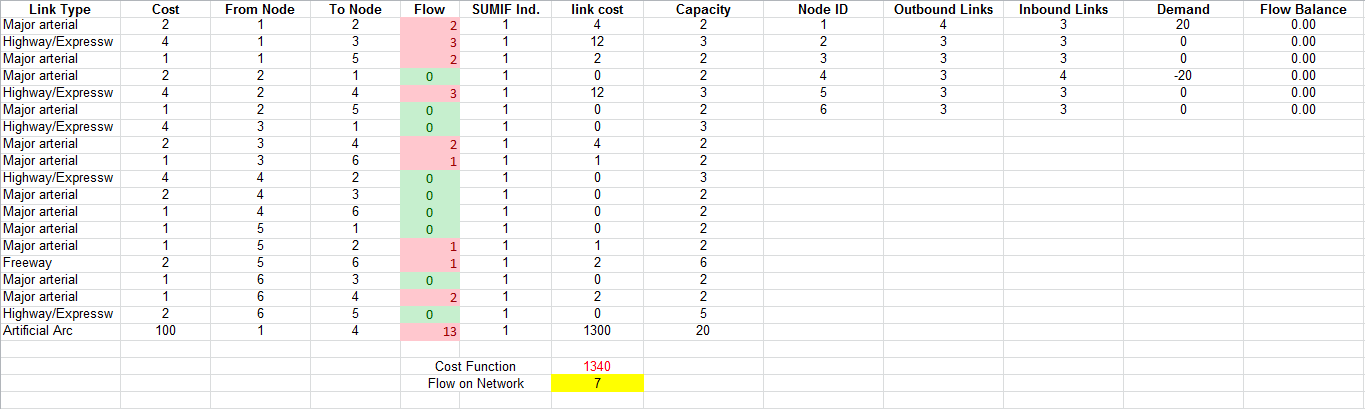
Step 2.1: The node block for this lesson should be similar to that of lesson 1. The first difference is that we are going to put a supply of 20 units for Node 1 and a demand on Node 4 of -20 units. The second change that needs to be made is to adjust the SUMIF operators in the Outbound and Inbound Links column. They need to be adjusted to include the new artificial link. If done correctly, the number of outbound links for node 1 should be increased to 4 and likewise the inbound links for node 4 should be increased to 4. The node block should look like:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Node ID** | **Outbound Links** | **Inbound Links** | **Demand** | **Flow Balance** |
| 1 | 4 | 3 | 20 | -20.00 |
| 2 | 3 | 3 | 0 | 0.00 |
| 3 | 3 | 3 | 0 | 0.00 |
| 4 | 3 | 4 | -20 | 20.00 |
| 5 | 3 | 3 | 0 | 0.00 |
| 6 | 3 | 3 | 0 | 0.00 |

**Step 3: Use Excel Solver to Find Optimal Solution**

Step 3.1: The solver will be used the same was as in lesson 2. The “By Changing Variable Cells” option should be changed to include the new artificial arc. Likewise the capacity constraint should be adjusted to include the new arc.

The maximum flow for the network is defined by the demand at Node 4 minus the flow on the artificial arc. It may be helpful to create a cell with this formula in it. The optimal solution for this network is shown below. The maximum flow for this network regardless of cost is 7.



**Problem 7:**

Use GNU LP solver and Chicago network with capacity constraints.

<https://github.com/xzhou99/STALite/tree/master/dataset/5_Chicago_sketch>

Reference:[Modeling in GNU MathProg language - a short introduction](http://www.im.pwr.wroc.pl/~pziel/lectures/om/glpk_notes.pdf)

<http://en.wikibooks.org/wiki/GLPK>

<http://www.cs.cmu.edu/~ckingsf/bioinfo-lectures/linearp.pdf>

Source code:http://trac.astrometry.net/browser/trunk/projects/archetypes/glpk-4.31/examples/spp.mod?rev=9312

|  |
| --- |
| /\* SPP, Shortest Path Problem \*/  /\* Written in GNU MathProg by Andrew Makhorin <mao@mai2.rcnet.ru> \*/  /\* Given a directed graph G = (V,E), its edge lengths c(i,j) for all  (i,j) in E, and two nodes s, t in V, the Shortest Path Problem (SPP)  is to find a directed path from s to t whose length is minimal. \*/  param n, integer, > 0;  /\* number of nodes \*/  set E, within {i in 1..n, j in 1..n};  /\* set of edges \*/  param c{(i,j) in E};  /\* c[i,j] is length of edge (i,j); note that edge lengths are allowed  to be of any sign (positive, negative, or zero) \*/  param s, in {1..n};  /\* source node \*/  param t, in {1..n};  /\* target node \*/  var x{(i,j) in E}, >= 0;  /\* x[i,j] = 1 means that edge (i,j) belong to shortest path;  x[i,j] = 0 means that edge (i,j) does not belong to shortest path;  note that variables x[i,j] are binary, however, there is no need to  declare them so due to the totally unimodular constraint matrix \*/  s.t. r{i in 1..n}: sum{(j,i) in E} x[j,i] + (if i = s then 1) =  sum{(i,j) in E} x[i,j] + (if i = t then 1);  /\* conservation conditions for unity flow from s to t; every feasible  solution is a path from s to t \*/  minimize Z: sum{(i,j) in E} c[i,j] \* x[i,j];  /\* objective function is the path length to be minimized \*/  data;  /\* Optimal solution is 20 that corresponds to the following shortest  path: s = 1 -> 2 -> 4 -> 8 -> 6 = t \*/  param n := 8;  param s := 1;  param t := 6;  param : E : c :=  1 2 1  1 4 8  1 7 6  2 4 2  3 2 14  3 4 10  3 5 6  3 6 19  4 5 8  4 8 13  5 8 12  6 5 7  7 4 5  8 6 4  8 7 10;  end; |

|  |
| --- |
| /\* MAXFLOW, Maximum Flow Problem \*/  /\* Written in GNU MathProg by Andrew Makhorin <mao@mai2.rcnet.ru> \*/  /\* The Maximum Flow Problem in a network G = (V, E), where V is a set  of nodes, E within V x V is a set of arcs, is to maximize the flow  from one given node s (source) to another given node t (sink) subject  to conservation of flow constraints at each node and flow capacities  on each arc. \*/    param n, integer, >= 2;  /\* number of nodes \*/    set V, default {1..n};  /\* set of nodes \*/    set E, within V cross V;  /\* set of arcs \*/    param a{(i,j) in E}, > 0;  /\* a[i,j] is capacity of arc (i,j) \*/    param s, symbolic, in V, default 1;  /\* source node \*/    param t, symbolic, in V, != s, default n;  /\* sink node \*/    var x{(i,j) in E}, >= 0, <= a[i,j];  /\* x[i,j] is elementary flow through arc (i,j) to be found \*/    var flow, >= 0;  /\* total flow from s to t \*/    s.t. node{i in V}:  /\* node[i] is conservation constraint for node i \*/    sum{(j,i) in E} x[j,i] + (if i = s then flow)  /\* summary flow into node i through all ingoing arcs \*/    = /\* must be equal to \*/    sum{(i,j) in E} x[i,j] + (if i = t then flow);  /\* summary flow from node i through all outgoing arcs \*/    maximize obj: flow;  /\* objective is to maximize the total flow through the network \*/    solve;    printf{1..56} "="; printf "\n";  printf "Maximum flow from node %s to node %s is %g\n\n", s, t, flow;  printf "Starting node Ending node Arc capacity Flow in arc\n";  printf "------------- ----------- ------------ -----------\n";  printf{(i,j) in E: x[i,j] != 0}: "%13s %11s %12g %11g\n", i, j,  a[i,j], x[i,j];  printf{1..56} "="; printf "\n";    data;    /\* These data correspond to an example from [Christofides]. \*/    /\* Optimal solution is 29 \*/    param n := 9;    param : E : a :=  1 2 14  1 4 23  2 3 10  2 4 9  3 5 12  3 8 18  4 5 26  5 2 11  5 6 25  5 7 4  6 7 7  6 8 8  7 9 15  8 9 20;    end; |