Maxwell Sanders

**To Compile and Run**

If you want to compile it, then you will probably have to compile it in NetBeans. I’m not sure whether or not it could compile in other environments, but I know that it should compile in NetBeans. This can be done by importing the whole project folder titled “Netbeans Project” into NetBeans.

After it is compiled it should be able to run like any other Java file.

* The file has a considerable amount of GUI overhead, so it might take a second or so to boot up.

**Basic functionality**

To understand what my program does, it goes ahead and runs the algorithm to completion saving every step along the way. This produces some memory overhead but does allow the user to click back and forth through steps.

The master tab will display (upon initialization)

* The distance vector table
* The steps it takes to completion
* The time it took to get to completion (if a link cost is changed, the time it takes to fix that change, not necessarily the time to full completion anymore)
* The current step
* Buttons to go to the initial and end steps
* Buttons to move forward and backward steps

Each node tab will display

* The editable cost to go to each link
* Buttons to move forward and backward steps
* A button to recalculate from an edit
* The current step

The file chooser tab, which can be covered in the next section

**Select the Input File**

To select the input file, click over to the file chooser tab.

From there select the file that is formatted properly, and then click open.

Weird quirks:

-There is no error-checking, selecting an improperly formatted file will crash it

-The File Chooser will not disappear even after selecting a file

-Cancel has no functionality

These quirks are all due to the fact that the GUI is a pre-created NetBeans GUI, and I couldn’t figure out how to make it appear and disappear or how to get rid of the cancel button.

**Initial link state tables set up in GUI**

Once the file is selected it will be parsed and upon clicking back on Master you will be greeted by:

The distance vector table at step 0

How many steps it takes for the network to become stable

How much time it took for the network to become stable

Each of the Node tabs also will display their initial cost to all of the other nodes

**System detects a stable state**

My program calculates the stable state the moment the file is read in by calculating each step, and then the moment a step is the same as the step before it, the program stops and declares the system to be stable.

**Runs algorithm correctly in single step mode**

Since all of the steps are calculated and saved one by one, the user is able to go through each step of the algorithm and see how the Distance Vector table changes as well as go backwards through the steps for better accessibility.

**Runs algorithm without stopping, displays time**

Since all of the steps are run at the beginning, the user can see the time it takes to reach stability.

The user is also able to skip to the final step to see how the network ends up stabilizing

**Change a link cost and run from the previous state**

Clicking on any of the Node tabs, the user can adjust any value of any of the nodes and press recalculate to run the program starting from that step.

Upon recalculation you can go back to the Master tab to see how long it took to stabilize from that point, as well as how many steps the network has now taken towards trying to stabilize.

**Observations**

Upon line failure – The algorithm took a few more steps to re-stabilize the network without that line, and the total costs of the network goes up slightly.

Upon line repair – The algorithm is able to return to the way it was before line failure and the cost of the network returns to the original.

**Bugs and limitations**

As stated in the file choosing portion the **file chooser** has some bugs.

* It is incapable of being closed. I couldn’t figure out how to make it disappear after use, so it just stays there.
* Cancel doesn’t do anything. The NetBeans GUI had FileChooser as a default, but that comes with the fact that I can’t delete things off of it since it is all one piece.
* Files that aren’t properly formatted crash the program. The program is set to read data in a certain way and when it isn’t then the program will crash trying to shove garbage data into arrays.

**The timing**. The timing isn’t the timing to full completion, it is the timing to completion from the point of calculation.

* If you open a file time will display the time it takes to stabilize the network
* If you edit a link, time will display the time it takes to stabilize the network from that link change, not how long it will take to stabilize the network in total.

**Total steps**. Total steps of the network are the total steps it took from file read to stabilize the network. This includes all of the steps added from link changing. Upon a link change total steps will not reflect how many steps it took to stabilize from the link, it will still only display the total steps.

**References**

I had to use NetBeans Java Swing GUI, so all of the documentation for that GUI I found based on the tooltips inside of the IDE.

I used the algorithm from the classroom slides.

To figure out how to do the timing, I looked at StackOverflow <https://stackoverflow.com/questions/1770010/how-do-i-measure-time-elapsed-in-java>