תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

Software Engineering Department

Ort Braude College

##### Capstone Project Phase B

**Optimal Design Of Online Problems With Application To Optical Networks**

**22-1-R-14**

Anya Hako garhadanya@gmail.com

Shirin Labay labay.shirin@gmail.com

**Supervisor:** Professor Shmuel Zaks

1. **Project Review and Process Description**

• תיאור של מה שעשיתם )השגתם( בפרויקט: ▪ עבור פרויקט במסלול מחקר: תיאור הפתרון שבניתם: מבנה התוכנה ואופן פעולתה . הסבר על האלגוריתמים / השיטה שננקטה במחקר.

In part B of the project we have implemented the algorithm we discussed in part A. For that we did a user interface for different topologies – path topology and ring topology.

In each topology we created three tabs – the first tab is for the visualization of the algorithms. There the user can build his own network by choosing number of nodes and edges. He can see the process of coloring the network relying on the MINADM algorithm.

The second tab is for showing the optimal solution of a given network. Here the user can see the solution and also can see the process of coloring and see the difference between the online algorithm MINADM and the offline algorithm which is the optimal solution. And finally the last tab is for showing the average analysis. The user can see the 3D graph where the X represents the number of nodes, Y represents the number of paths and Z represents the ratio.

To recall, the ratio is the solution of the online algorithm MINADM / the average ADM used.

Also, the user can input his own network by choosing number of nodes and a range of paths he wants and then it calculates the ratio for each number of paths in the range.

Online MINADM :

Firstly, if there are ADMs at node u and v that are the same color lmda, and these ADMs are not shared by two or more lightpath, the algorithm checks if lmda is feasible for Pi by checking if any of the lightpaths that form a chain colored lamda with endpoint u and v is conflicting Pi. If none of them conflicts Pi then lamd is feasible for Pi, and Pi is assigned the color lmda and is connected to the ADMs at its endpoints.

If there is an ADM at node u or v that is colored lamda, the algorithm checks if lamda is feasible for Pi, if it is then Pi is connected to that ADM. The other endpoint will also have an ADM colored lamda.

If there are two free ADMs with different colors then one of the endpoints is connected to one of them ADMs. The other endpoint gets a new ADM with the same color as the ADM in the first endpoint.

Lastly, if there are no free ADMs, Pi gets a new color lambda that is feasible for Pi. This means there will be new ADMs with color lmbda at nodes u and v.

**GUI Testing**

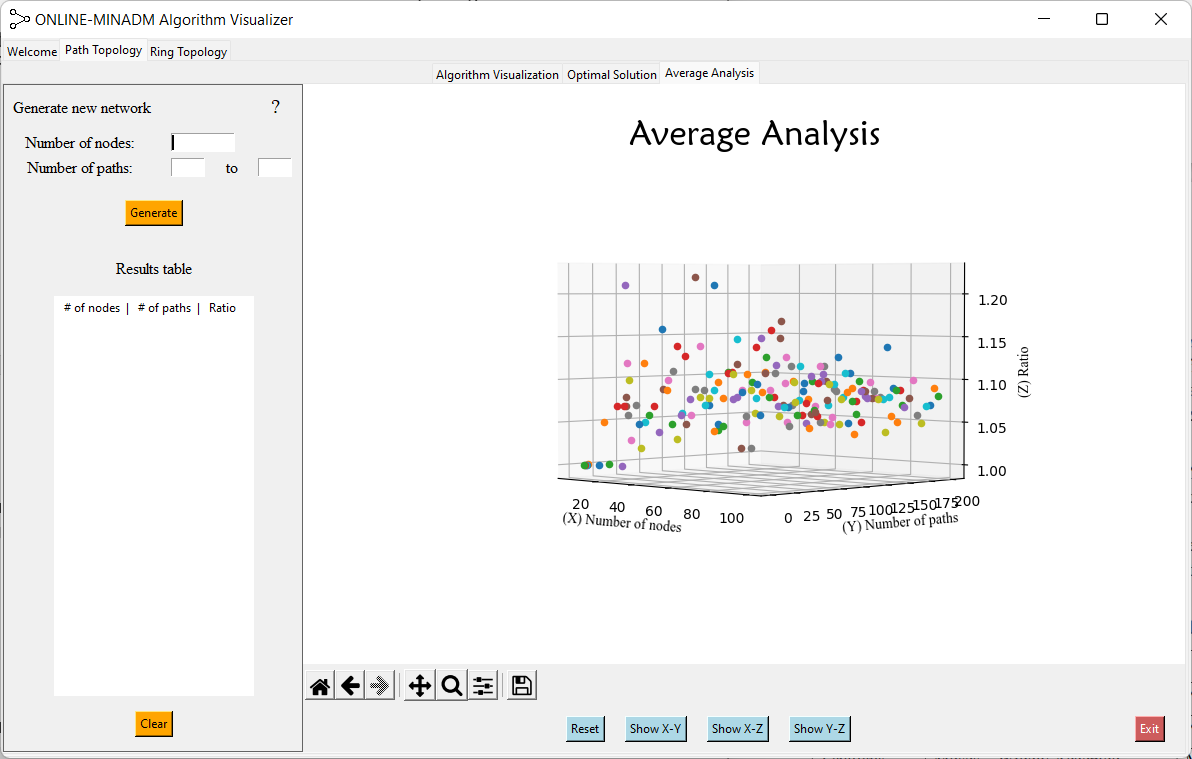
**Outcome**

The researcher's outcome is a tool intended to show the process of the online algorithm and to show the average analysis of the topologies.

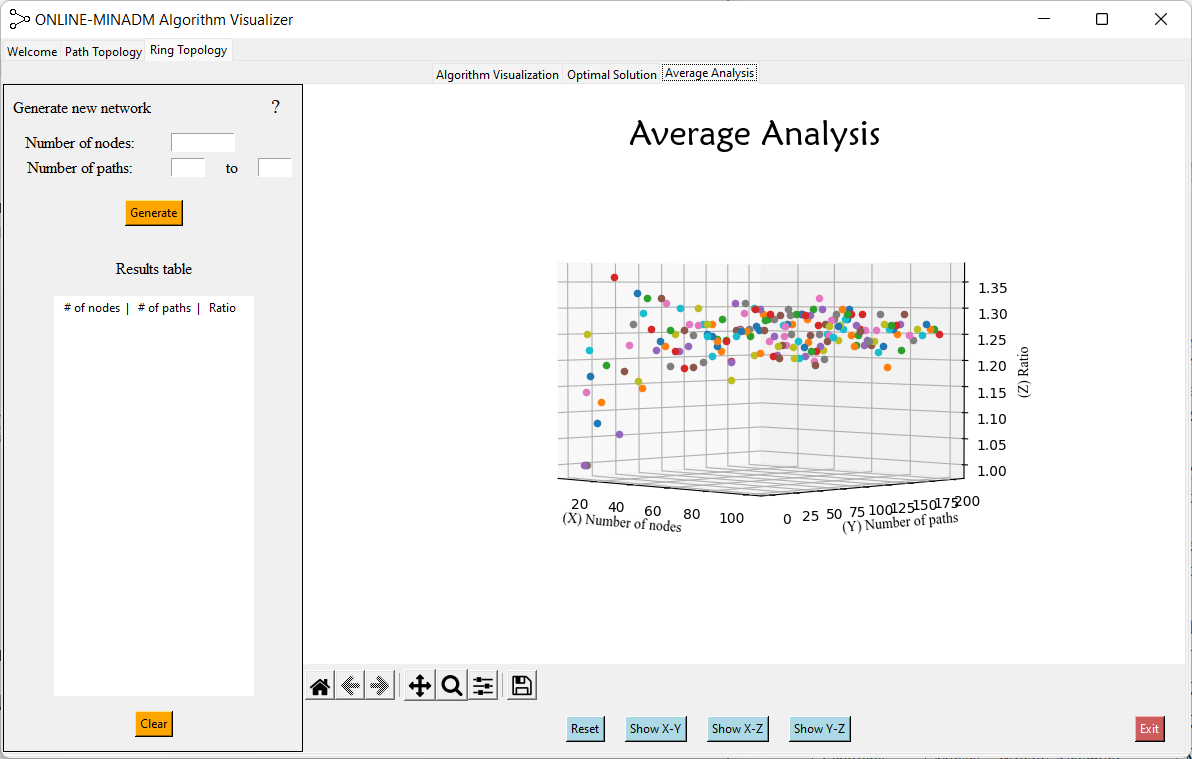
**Analysis and Conclusions**

Average analysis :…………………………………………………….

Path:



Ring:



**Maintenance Guide**

**UML diagram**

**Use Case Diagram:**

**Activity Diagram:**

**Version Control**

The project manages changes to files and revisions through GitHub. The repository of the project, together with a description and installation instructions, can be found in:

**https://github.com/AnyaHako1/algorithm-visualizer-project**

**User Documentation**

**General description**

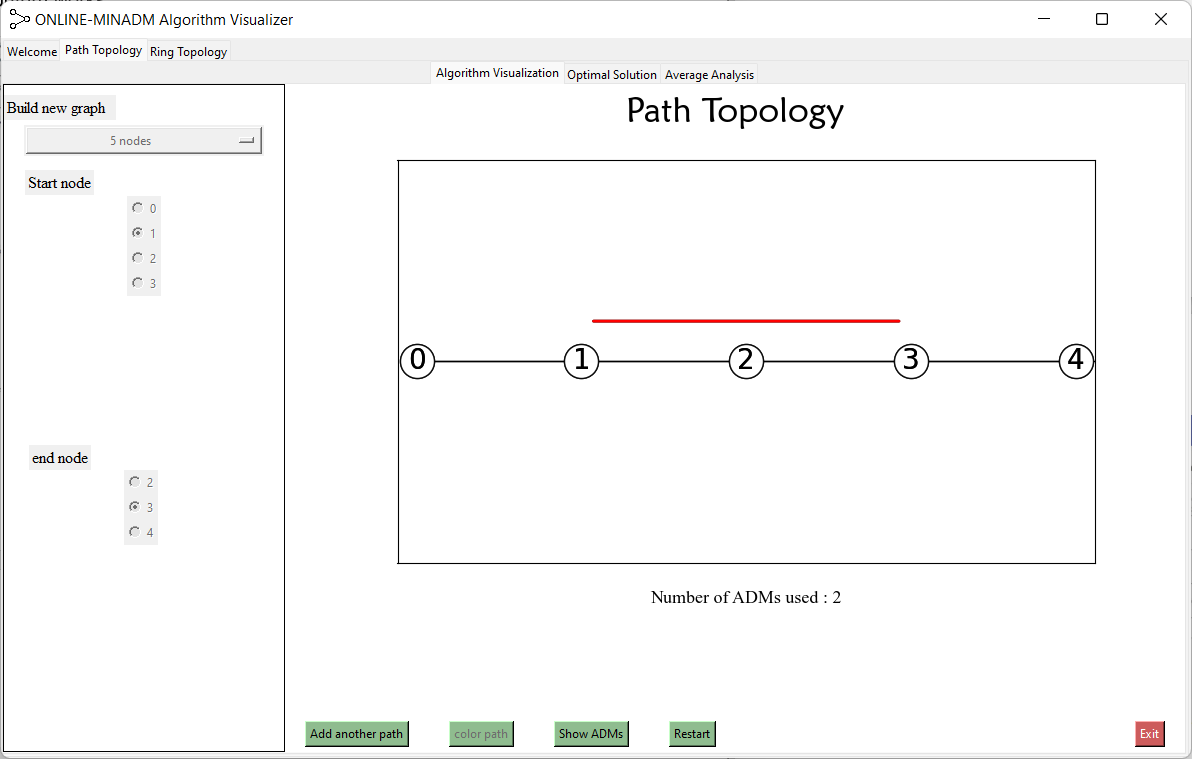
The software is a visualization of an online algorithm that shows the process of the algorithm and how it works. The GUI lets users to use the software intuitively. In the front page you can choose which topology you want to see, it is either a path topology or a ring topology.

The first page of the software is shown below.

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

There are couple of things that you can do in the system, such as building your own network. It is done by choosing number of nodes you want in the network and choosing edges. You can see the number of ADMs used in the model you built and you can see the process of the coloring of the network. As shown below, this is a 5 nodes network with an edge (1, 3), and when clicking on "Show ADMs" you can see the number of ADMs used until now. After that you can add more paths to the network and see the process of the algorithm.



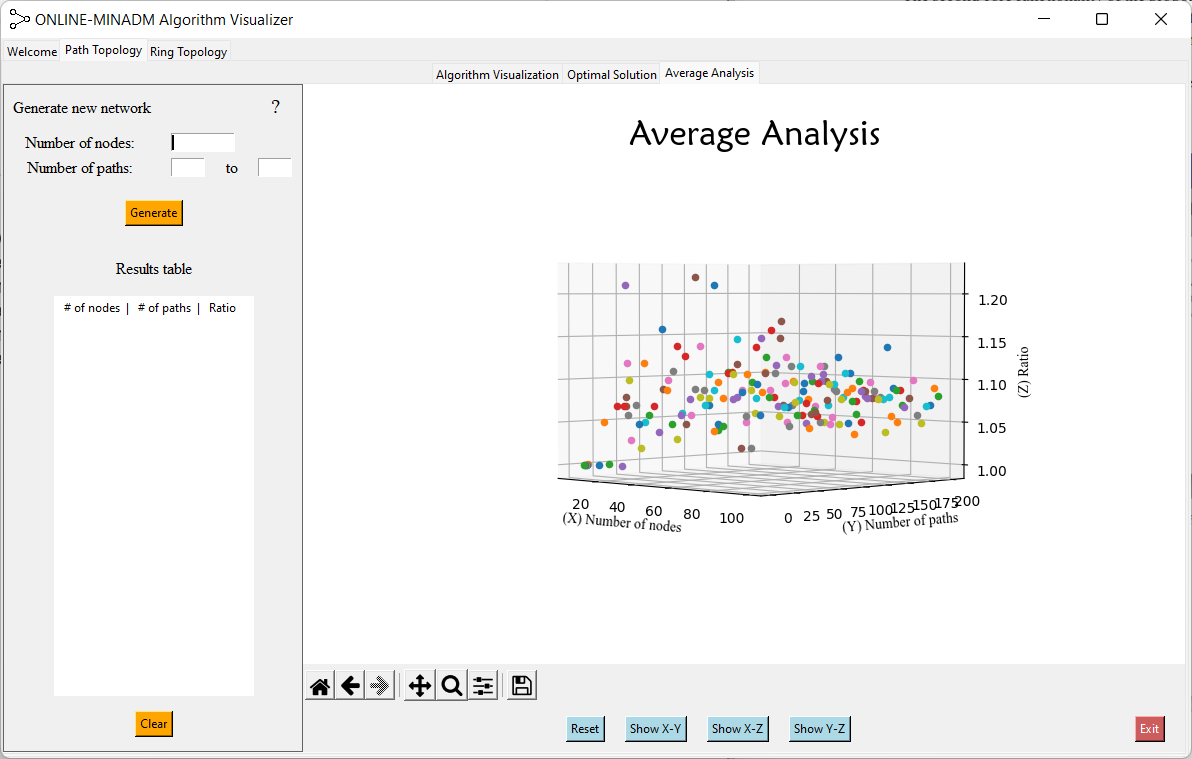
The second core functionality of the system is to see the optimal number of ADMs used in against the number of ADMs used using the algorithm.

As shown below, the optimal number of ADMs used in this network is 8 and using the algorithm the number of ADMs is 10. Also, you can see where it used more ADMs relying on the colors. More than that, you can see the process when clicking on the buttons below.



The third functionality of the system is to show average analysis. This is shown through a 3D graph where the X represents the number of ADMs, Y represents the number of paths and Z represents the ratio.

In the left side we have a GUI where the user can input his own number of nodes and a range of paths and it outputs on the table the results for each path what is the ratio. Also, the user can change the graph and see the different angles of the graph by clicking on the button in the bottom.



**Data Organization**

This project does not use a database.

Requirements and Installation

* Hardware
* Operating system: windows 11 64-bit
* NVIDIA GPU
* Software
* Python 3.10
* PyCharm

**Future changes**

**References**