המכללה האקדמית להנדסה בכרמיאל

בראודה

Software Engineering Department

Ort Braude College

Course 61999: Extended Project in Software Engineering

Capstone Project Phase B

Project Number: 23-1-R-19

Online minimization of switching cost in optical networks

Students:

Ahmad Aamar - 206713034

Email: ahmad.aamar1410@gmail.com

Mohamed abu assad - 316449495

Email: m7mad114@gmail.com

Supervisor:

Shmuel Zaks

Email: shmuel.Zaks@braude.ac.il

1.Project Review

1.1 introduction

In phase A of our research project, we analyzed an online optimal algorithm called "ONLINE-MINADM" that addresses this problem. This algorithm is designed to color lightpaths in optical networks, with the goal of minimizing the number of ADMs used. We studied the algorithm and its upper and lower bounds on two different network topologies: the path topology and the ring topology.

Our analysis revealed that the competitive ratio of the ONLINE-MINADM algorithm is 3/2 for the path topology and 7/4 for the ring topology. This means that for any sequence of lightpaths in the path topology, the total cost of the solution (i.e., the number of ADMs used) is at most 3/2 times the number of ADMs used by the optimal offline algorithm. Similarly, for the ring topology, the total cost of the solution is at most 7/4 times the optimal offline algorithm.

In Phase B of our project, we aim to simulate and evaluate the performance of our proposed online algorithm in various network topologies, specifically in ring and path configurations. Our main objectives are twofold: first, to visually demonstrate the execution of the algorithm, and second, to evaluate its expected performance. While the worst-case competitive ratio is a standard metric for online algorithm analysis, in practical scenarios, it may be more relevant to evaluate the algorithm's expected cost compared to the best possible outcome. By simulating the algorithm in different network topologies, we can gain insights into its behavior in various scenarios and identify potential areas for improvement. Our ultimate goal is to develop a practical, effective, and cost-efficient algorithm for minimizing switching costs in optical networks that can be used in real-world settings.

1.2 algorithm

The ONLINE-MINADM algorithm we studied in Phase A is an online algorithm for coloring lightpaths in optical networks that aims to minimize the number of ADMs used. We analyzed the algorithm's upper and lower bounds on two different topologies: the ring topology and the path topology. In the ring topology, the algorithm achieved a competitive ratio of 7/4, meaning that its performance was within 1.75 times the optimal offline algorithm's performance in the worst case. Similarly, in the path topology, the algorithm achieved a competitive ratio of 3/2. These competitive ratios provide a theoretical guarantee on the performance of the algorithm in these specific network topologies.

main steps for the ONLINE-MINADM algorithm:

1.Initialize an empty set of lightpaths and an empty set of ADMs.

2.For each incoming lightpath request, assign a color that is different from the colors assigned to the set of lightpaths that share the same link as the incoming request.

3.If there is no color available, add a new ADM to the network and assign it a new color.

4.Repeat steps 2-3 for all incoming lightpath requests.

5.The algorithm outputs the set of lightpaths and ADMs used to serve all incoming requests.

Additional example:

The input is a graph g={V,E} such that V={v0,v1,v2,v3,v4,v5} , E={ e0=(v0,v1)

For sequence of paths is: p1, p5, p6, p2, p4, p3, p7

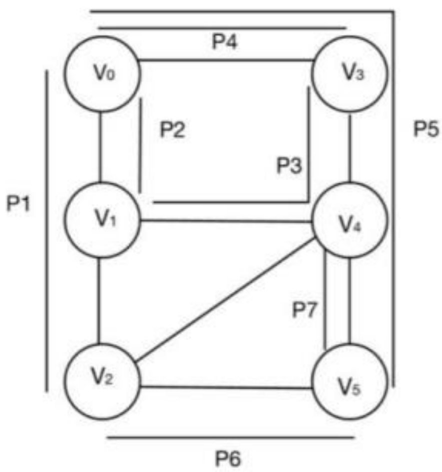


Fig.7. the input.

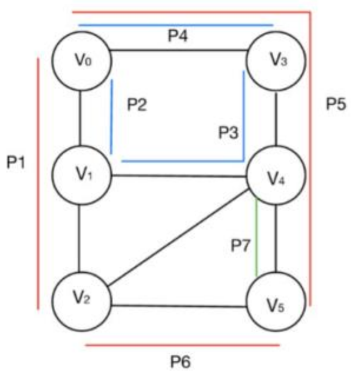


Fig.8. optimal solution

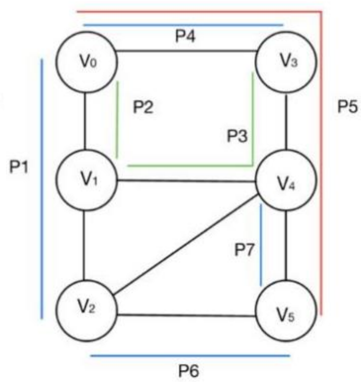


Fig.10. Online-MINADM solution

In the example where the optimal solution is 8 ADMs and the ONLINE-MINADM algorithm solution is 10 ADMs, we can see that the algorithm's solution is not optimal but still relatively efficient. While the optimal offline algorithm uses the minimum possible number of ADMs, it requires knowledge of all future requests, which is not practical in online scenarios. On the other hand, the ONLINE-MINADM algorithm works in an online manner and can adapt to incoming requests without having complete knowledge of the future. In practice, the ONLINE-MINADM algorithm performs much better than its worst-case performance and achieves a competitive ratio of 7/4. Thus, even though the algorithm's solution is not optimal, it still provides a good approximation of the optimal solution and reduces the switching cost compared to using more ADMs.

1.3.Research process

In phase B of our research project, we focused on implementing the ONLINE-MINADM algorithm and evaluating its performance in various network topologies. To build the software, we first reviewed the paper that introduced the optimal coloring algorithm and thoroughly understood the ONLINE-MINADM algorithm, as we did in phase A. We then searched for programming languages and libraries to use in our implementation and found that Python offered many helpful visualization libraries for displaying network topologies and user interfaces.

As a research project, our primary goal was to implement and evaluate the ONLINE-MINADM algorithm, rather than testing it for practical use. Therefore, the main testing we conducted was to ensure that user input was valid and that the algorithm was working correctly. With our implementation complete, we then proceeded to evaluate the performance of the ONLINE-MINADM algorithm in various network topologies, specifically ring and path configurations, as outlined in our evaluation plan from phase A.

1.4. result

Based on the aforementioned information, we conducted extensive testing of the algorithm on various network sizes and different orders of lightpaths within each network. The purpose was to calculate the average solution of the algorithm for each network and input, allowing us to assess its performance.

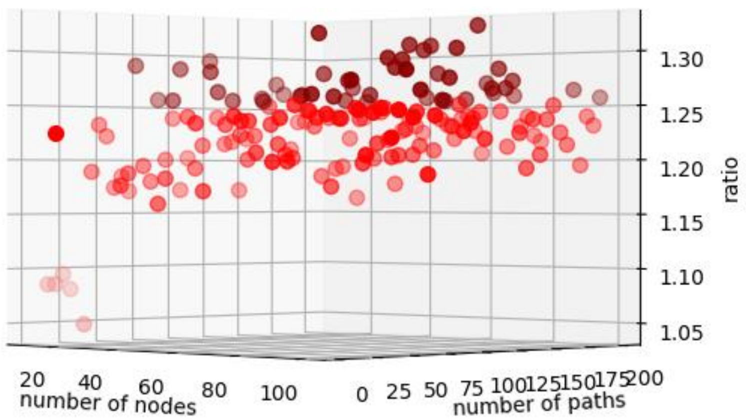
The algorithm was tested on networks ranging from 10 to 100 nodes. For greater precision, users can run the algorithm on additional network sizes. The average ratio, which represents the relationship between the algorithm's average solution and the optimal solution, was computed for both path and ring topologies.

In the case of the path topology, the average ratio obtained was 1.22. This indicates that, on average, the algorithm's solution was 22% worse than the optimal solution. It is noteworthy that the competitive ratio for this topology was initially determined to be 3/2 (1.5). However, our experimental results show a significantly better performance than anticipated.

For the ring topology, the average ratio obtained was 1.27. This indicates that, on average, the algorithm's solution was 27% worse than the optimal solution. Similarly, our experimental results outperformed the competitive ratio of 7/4 (1.75) initially estimated for this topology.

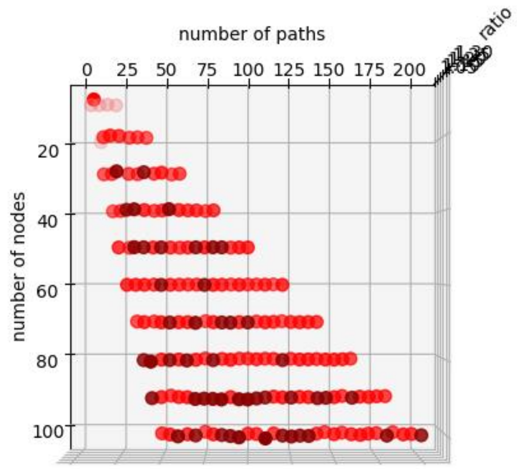
To visualize the results in the path topology, a 3D graph was created. The x-axis represents the number of nodes in the network, the y-axis represents the number of paths in the input, and the z-axis represents the average competitve ratio. In the graph, darker red points represent results with a ratio higher than 1.25, light red points represent ratios between 1 and 1.1, and the remaining points are red.

These findings indicate that our algorithm consistently produces solutions that are notably better than the worst-case scenario predicted by the competitive ratios. It demonstrates the algorithm's effectiveness and suggests that it can provide efficient solutions for various network sizes and input configurations in both path and ring topologies.



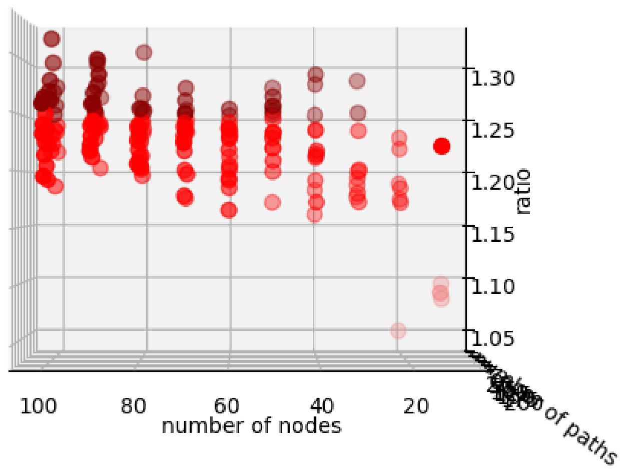
Path topology results

The software provides users with the ability to view the 3D graph from various angles, allowing them to examine it from different perspectives. By exploring the graph from these angles, users can gain valuable insights and draw meaningful conclusions. By focusing on three particularly interesting angles, additional insights can be gleaned, leading to further observations and conclusions.



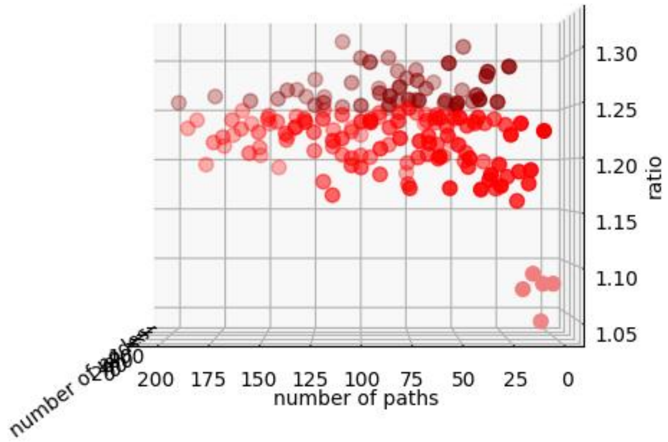
Path topology results, x-y angel

When considering the x-y angle, an analysis of the algorithm's performance was conducted on various networks. The algorithm was executed on networks with 'X' nodes, where the number of paths ranged from 'X/2' to '2X'. For instance, networks with 10 nodes were tested with 5, 10, 15, and 20 paths. For each network configuration, the algorithm was run on different orders of the input paths, and the average results were calculated.



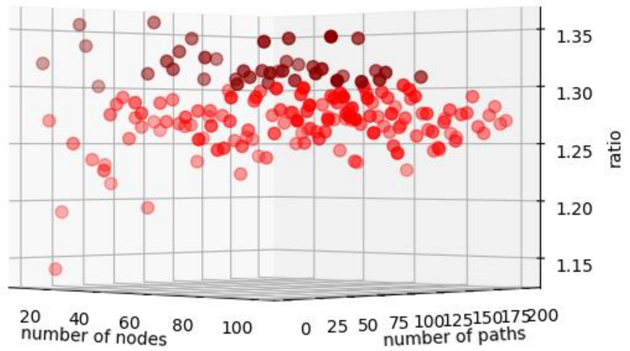
Path topology results, x-z angel

In the x-z angle analysis within the path topology, we examined the relationship between the number of nodes and the average ratio. Notably, the highest ratios were obtained in networks of size 80 and 100. However, it is worth mentioning that no points were found with ratios exceeding 1.5, which aligns with our expectations and indicates that the algorithm's performance remains within the expected bounds in this context.



Path topology result, y-z angel

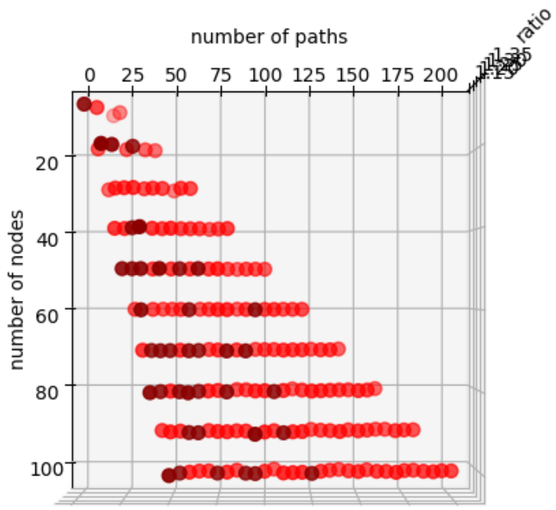
In the y-z analysis of the path topology, we examined the relationship between the number of paths and the resulting ratio. Notably, it was observed that the highest ratios consistently occurred on the right side of the graph, where the number of paths was smaller.



Ring topology results

This graph depicts the results from the ring topology. The subsequent graphs illustrate different angles of analysis within this topology. Similarly to the path topology, the points on these graphs are color-coded in varying shades of red based on the ratio of the result.

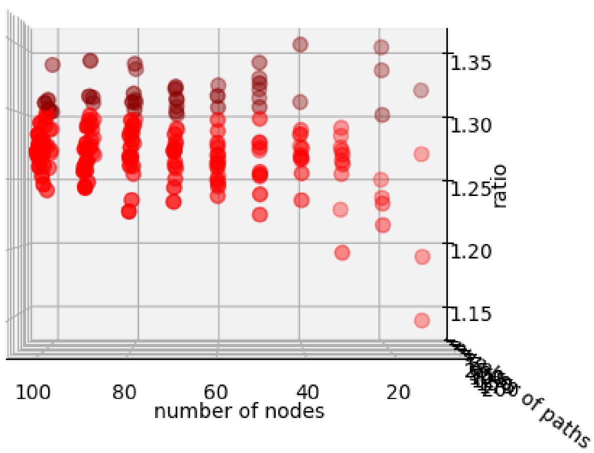
In particular, points with ratios ranging from 1 to 1.1 are represented by a light red color. Points with ratios between 1.1 and 1.3 are depicted as bright red. Lastly, all points with ratios exceeding 1.3 are shown as dark red on the graph. This color classification provides a visual distinction and helps identify the relative magnitudes of the result ratios within the ring topology.



Ring topology results, x-y angel

In the x-y analysis within the ring topology, we explored the relationship between the number of nodes, the number of paths, and the corresponding results. Interestingly, unlike the findings in the path topology, it appears that in almost every column of the graph, there is at least one data point marked as dark red.

This observation aligns with expectations, considering that the competitive ratio of the algorithm in the ring topology is higher, specifically 1.75, compared to the 1.5 ratio in the path topology. Although the threshold for darker red points is higher in the ring topology, the presence of such points in most columns indicates that the algorithm achieves higher ratios in this particular network topology.

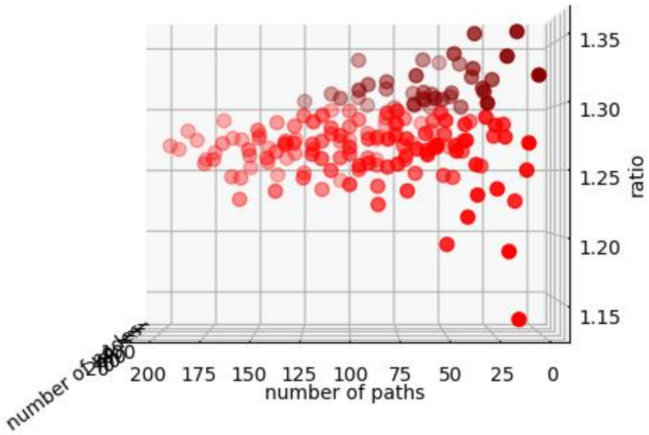


Ring topology results, x-z angel

In the x-z analysis within the ring topology, we examined the relationship between the number of nodes and the resulting ratio. Notably, for nodes ranging from 40 to 100, the minimum ratio obtained was higher than 1.2, indicating that the algorithm's performance remains above the specified threshold in this range.

The graph displays dense points, primarily concentrated within the range of 1.2 to 1.3 for the ratio. This clustering suggests that a majority of the results fall within this range, indicating consistent performance within that particular ratio interval.

As expected, no points with a ratio higher than 1.75 were observed, aligning with the anticipated behavior based on the competitive ratio of the algorithm in the ring topology.



Ring topology results, y-z angel

In the y-z analysis within the ring topology, we explored the relationship between the number of paths and the resulting ratio. Notably, it was observed that the lowest and highest ratios tend to occur on the right side of the graph, where the number of paths ranges from 10 to 50.

1.5 Conclusion

The "ONLINE-MINADM algorithm visualizer" software successfully achieves the primary objectives of this project phase. It effectively implements and visualizes the algorithm studied during phase A, allowing for a comprehensive understanding of its workings. The software also enables the algorithm to be tested on various network sizes and inputs, facilitating an assessment of its average performance.

2. User Documentation

2.1 User Guide

The purpose of the "ONLINE-MINADM algorithm visualizer" software is to provide a visual representation of the algorithm described in [1] and analyze its performance, serving as a valuable addition to the referenced paper.

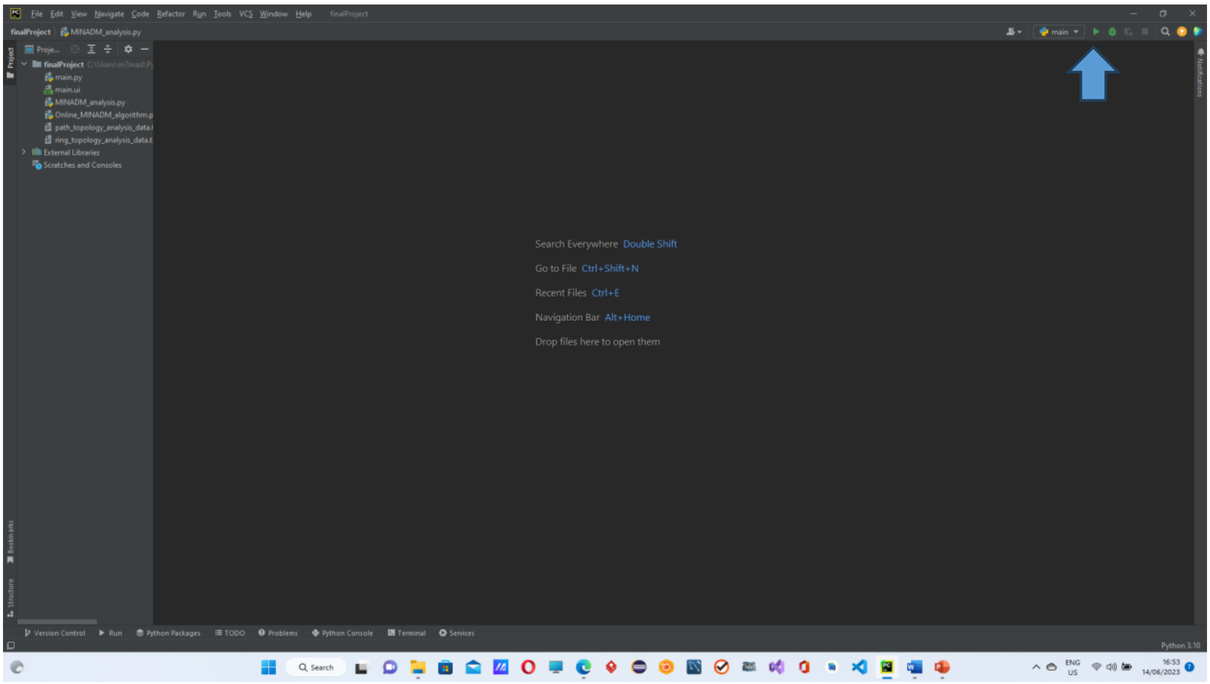
The algorithm is implemented in Python using PyCharm IDE version 2022.3.3. The software utilizes the "networkx" library for drawing networks and "matplotlib" for displaying these visualizations. To create the user interface, the "pyqt5" library is employed.

The software primarily focuses on path and ring topologies. It allows users to visualize the implemented algorithm on networks of their choice and provides a side-by-side comparison of the algorithm's coloring and the optimal solution for a specific network.

Furthermore, the software conducts an analysis of the algorithm's performance. It achieves this by executing the algorithm on various networks and inputs. Users can contribute to the statistics by adding their own data, and the software recalculates the average accordingly. The performance results for each network are displayed on a 3D graph, providing a comprehensive visualization of the algorithm's performance across different scenarios.

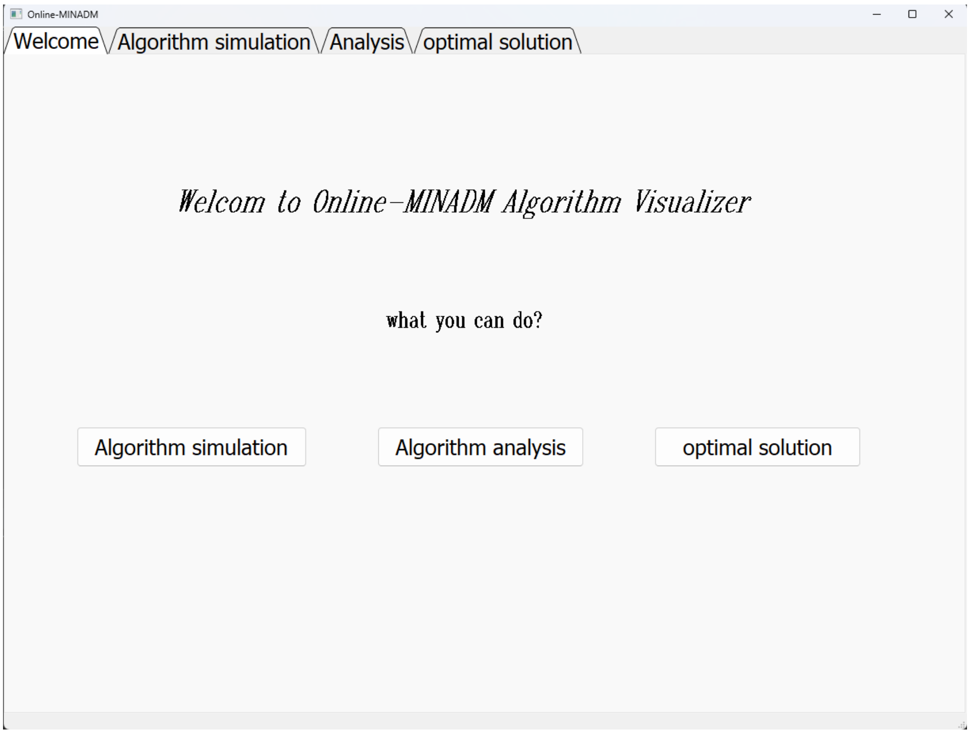
2.2 Operating Instructions:

Executing the software is a straightforward process requiring an IDE to open the code and a simple action of clicking on the "run" icon. Please refer to the screenshot below for visual reference.



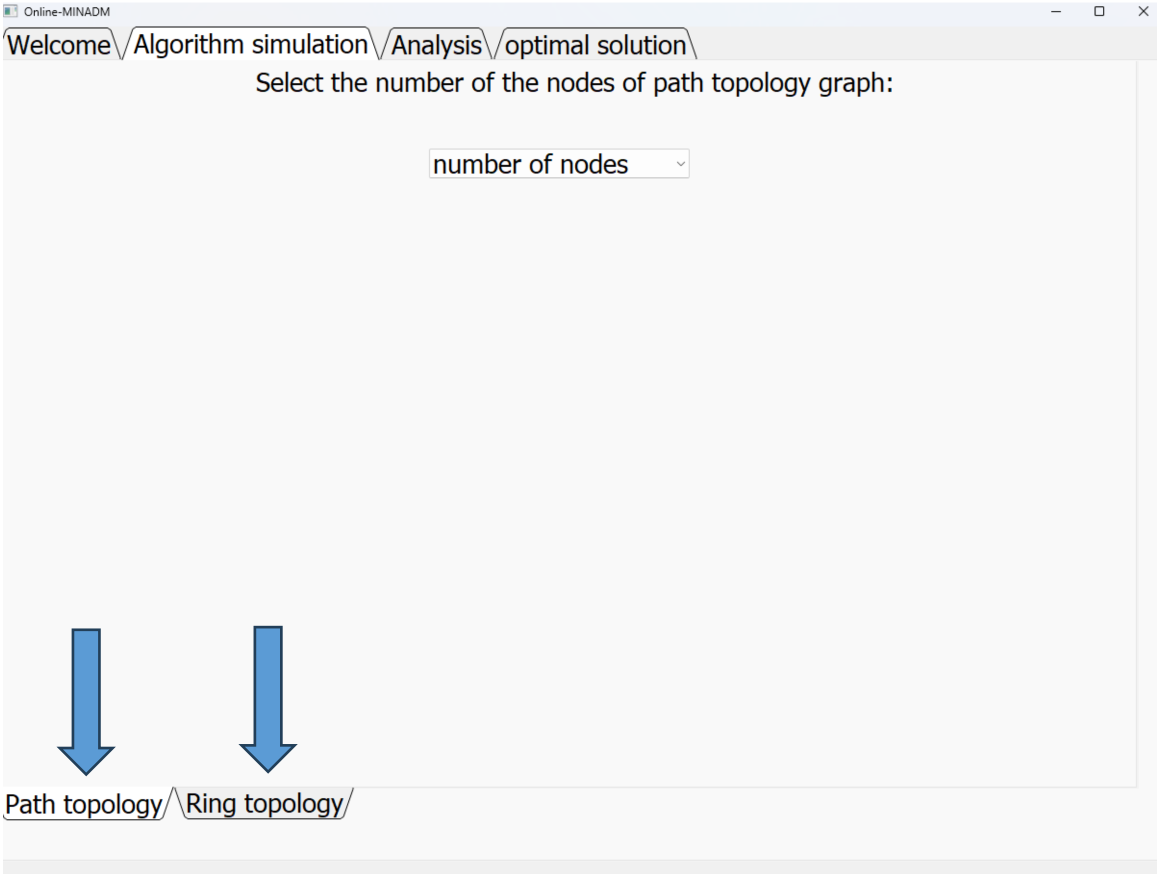
By following this procedure, the software will be launched and ready to use.

This will open the main welcome page of the software

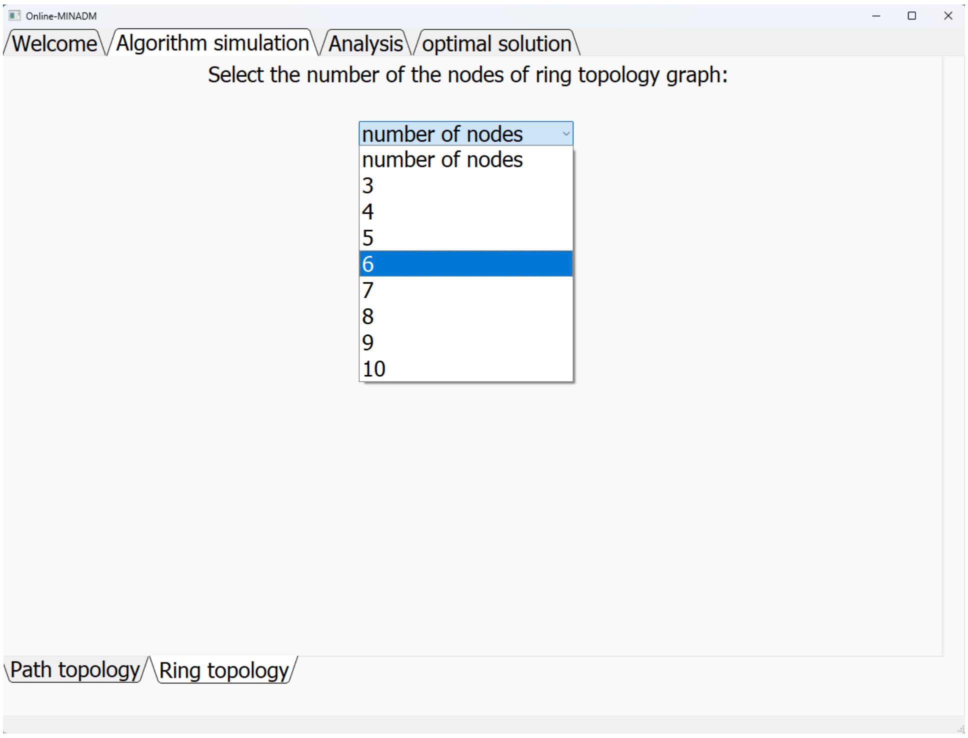


On the welcome page, the user can do choose to open algorithm simolution by pressing the button “Algorithm simulation” or the tab “Algorithm simulation” on the top to start visualizing the algorithm on certain topology. To look at the results or see comparison between the optimal and the algorithms solution, the user can click on the buttons ”Algorithm analysis” or “optimal solution” or by clicking on analysis and optimal solution tabs.

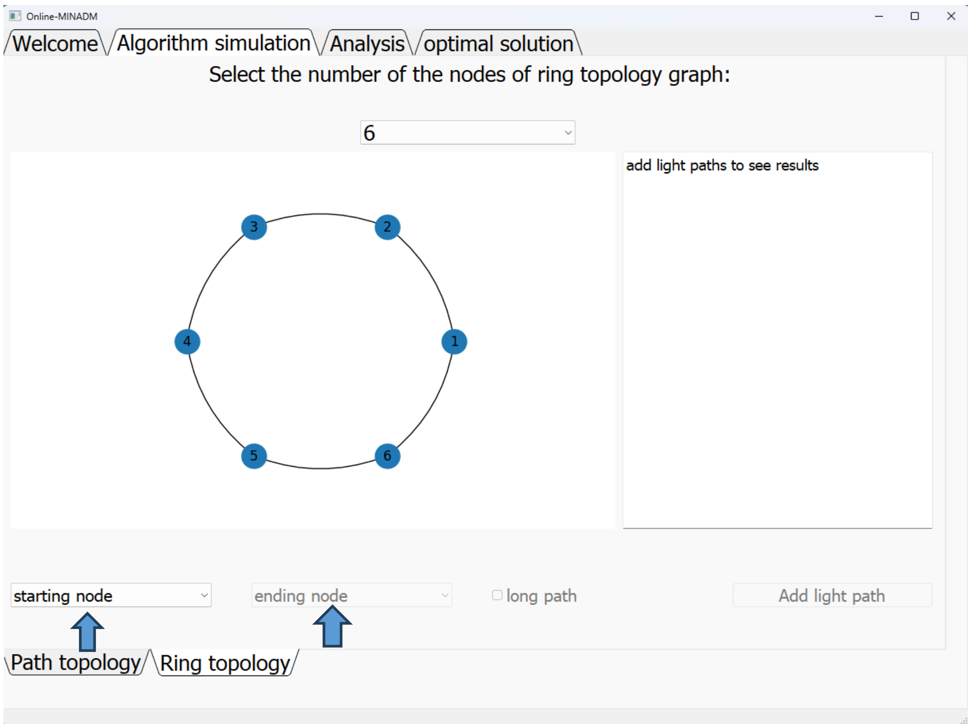
Pressing on “Algorithm simulation” button will open a tab that enables the user to start visualizing the algorithm in ring topology or path topology graph, the user can choose the topology by selecting the appropriate tab.



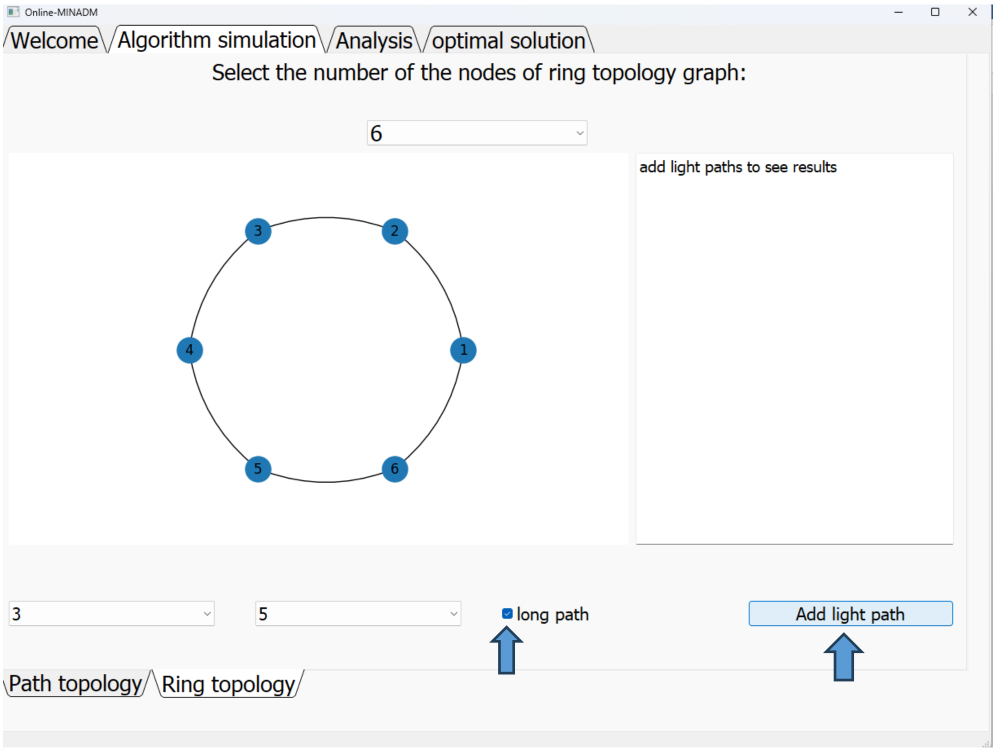
the user can choose the number of nodes by clicking on the combo box and select a number 3-10



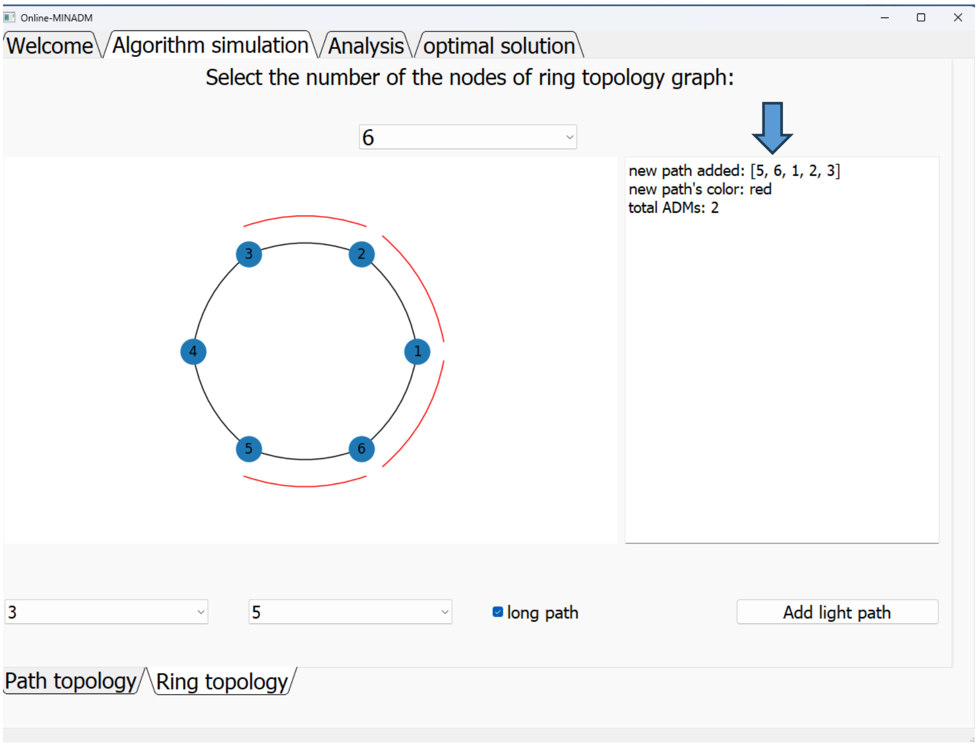
For example, if the user chooses 6, the system will generate new cycle graph with 6 nodes and then the user can start adding light paths



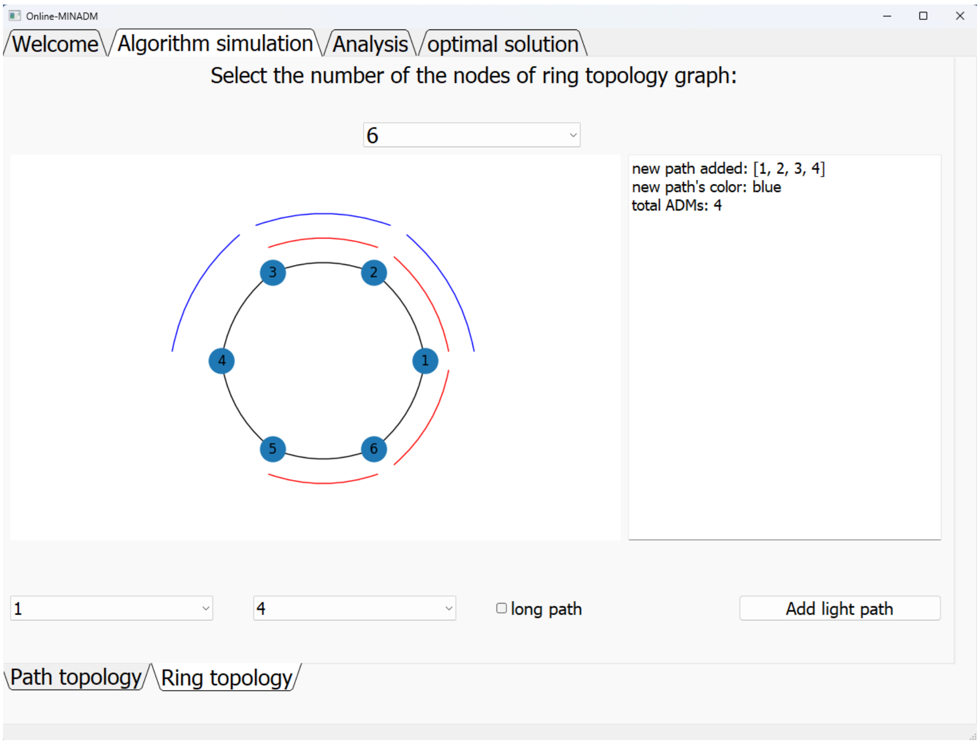
To add new light light path, the user need select a starting node first and then selecting ending node, this fields is mandatory. in addition, the user can choose which path he wants, if he wants the long path he needs to check the check box as shown in the picture bellow, or leave it unchecked if he wants the short path, at the end he needs to cilck on “Add light path” button. For example if the user chooses the starting node and the ending node to be 3 and 5 like the picture bellow and checked the check box and clicked on the button, the program will add new ligth path to the graph that starts from 3, ends in 5 and goes through 2,1,6.



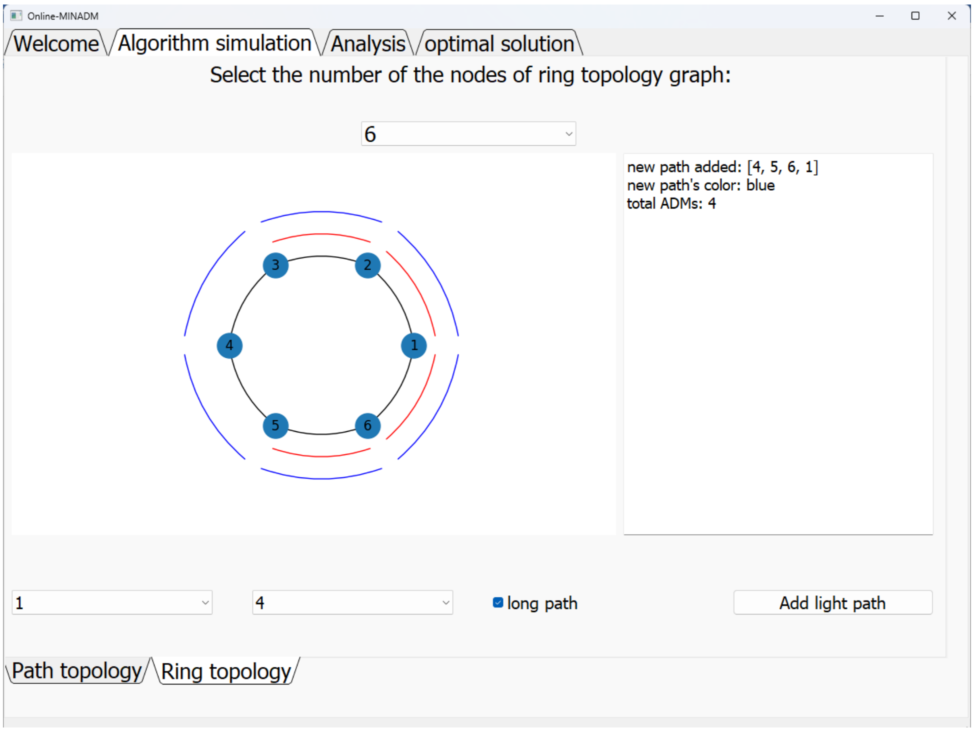
After clicking on “Add light path” button, the program will take the data from the input fields and generate a path based on the data and run the “Online MINADM” algorithm on it, and draw it on the with the proper color. Moreover, on the right side the user can see the results of the algorithm.



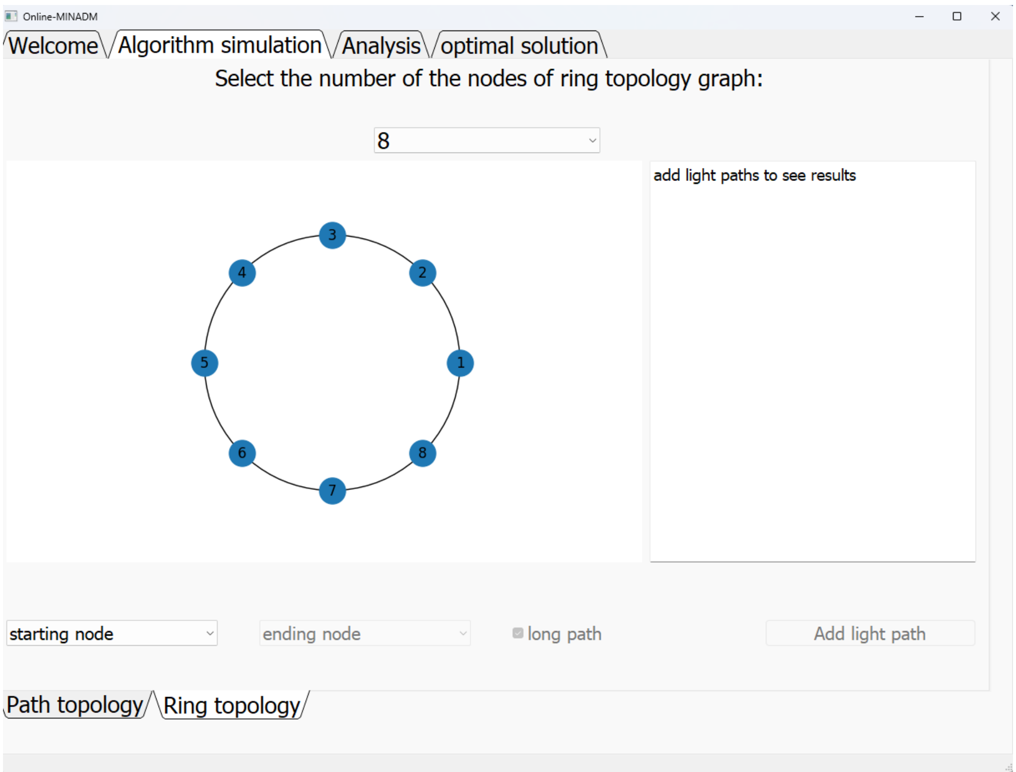
By rpeating the same process, the user can add as many light paths as he wants.



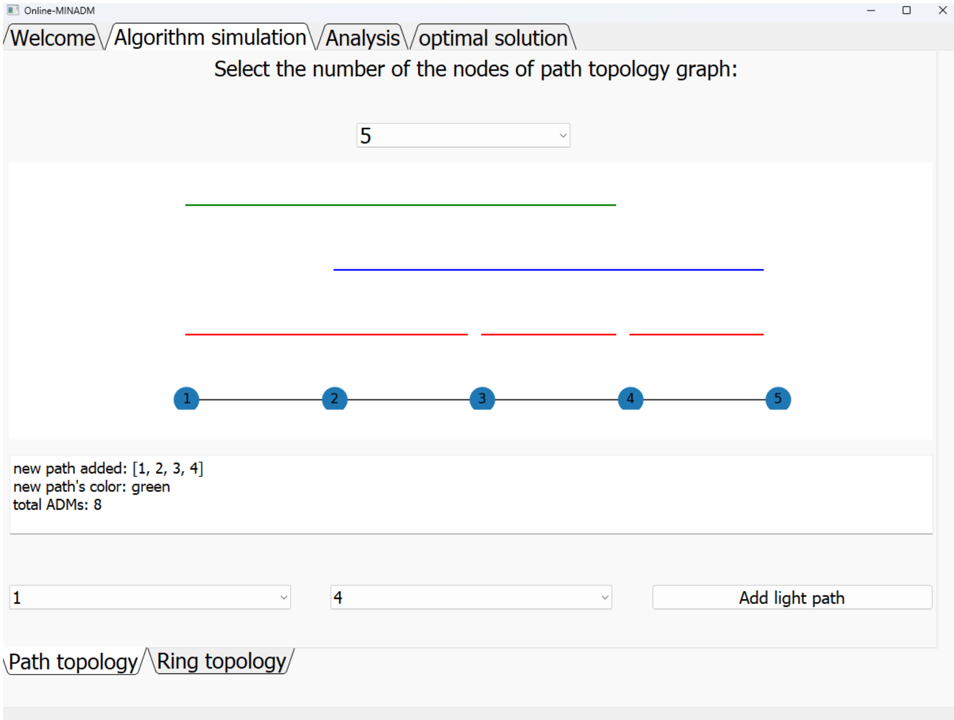
If the user adds new light path that there is no proper color for it, the program will add it with new color as we can see in the picture above



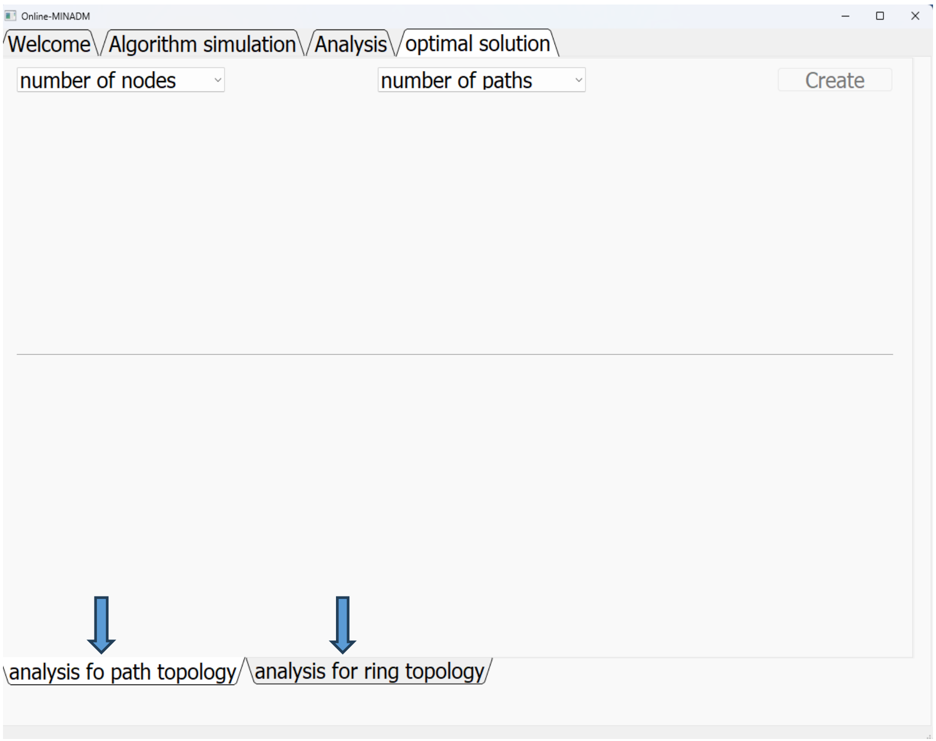
The user can start new experiment by rechoose the number of node from the combo box and the program will automatically clear the graph and draw new clear one.



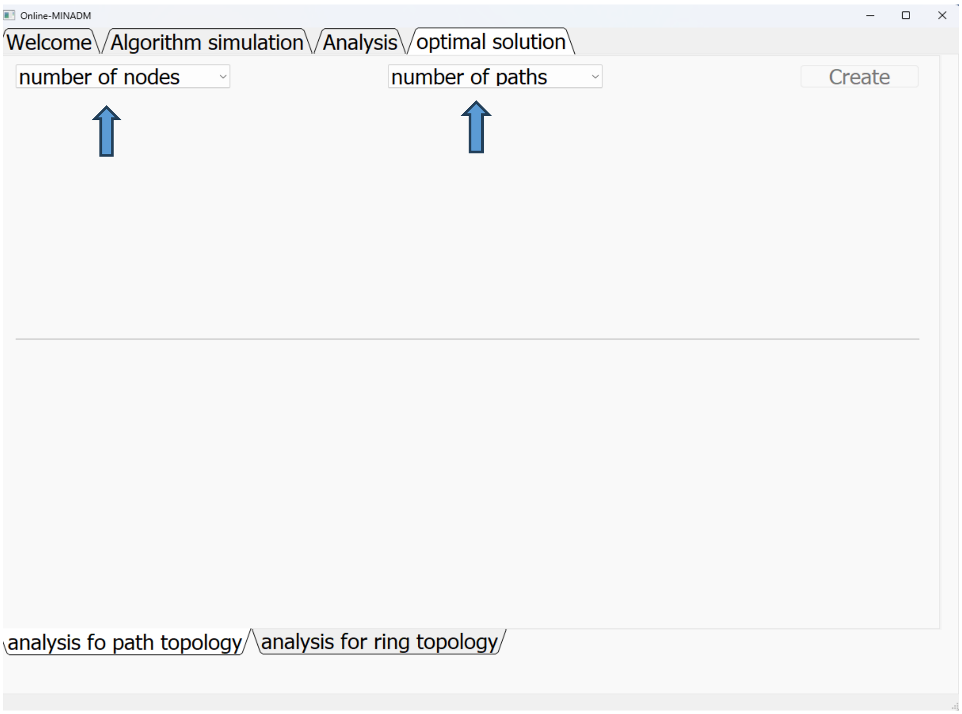
The user can do the same thing on path topology graphs by entering the “path topology” tab.



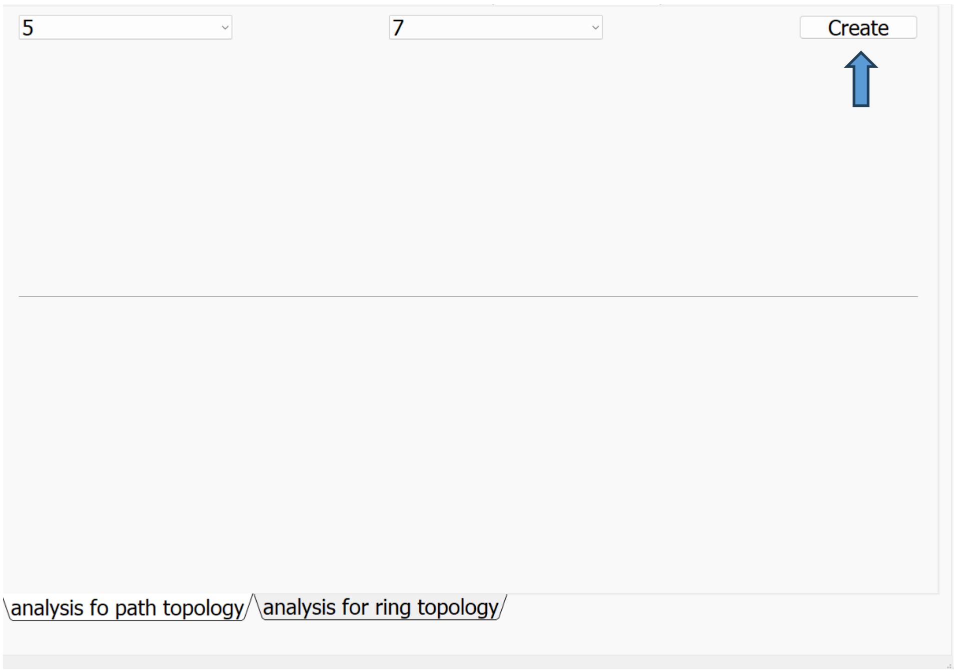
The user can enter to the “optimal solution” tab to do comparison between optimal solution and online solution. Inside this there is to tabs one for ring topology and the another for path topology.



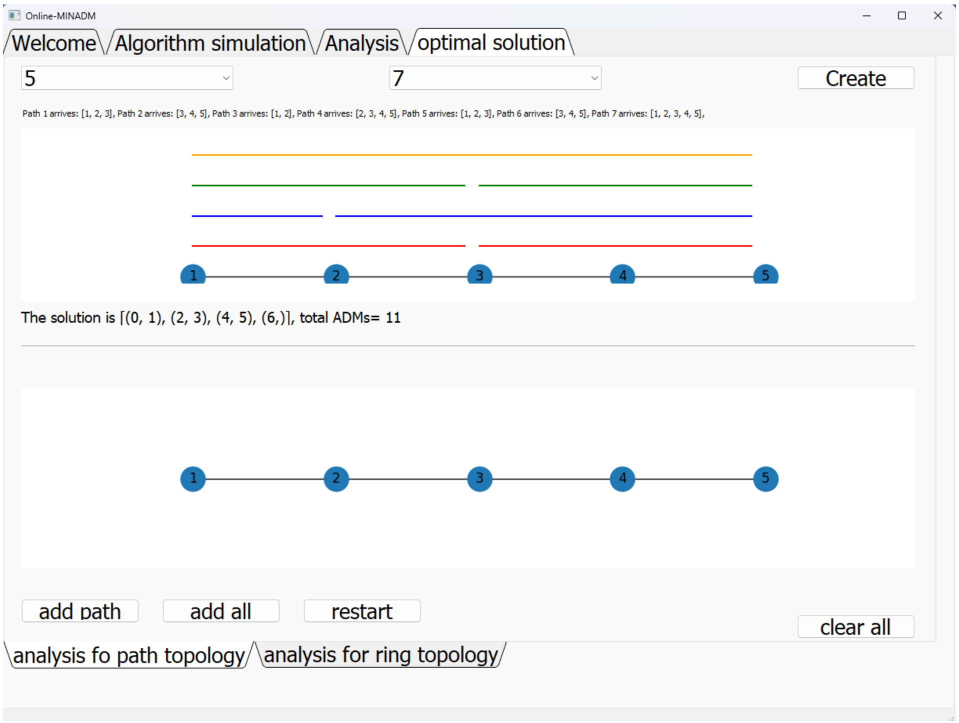
To begin, the user needs to choose the number of nodes and the number of paths



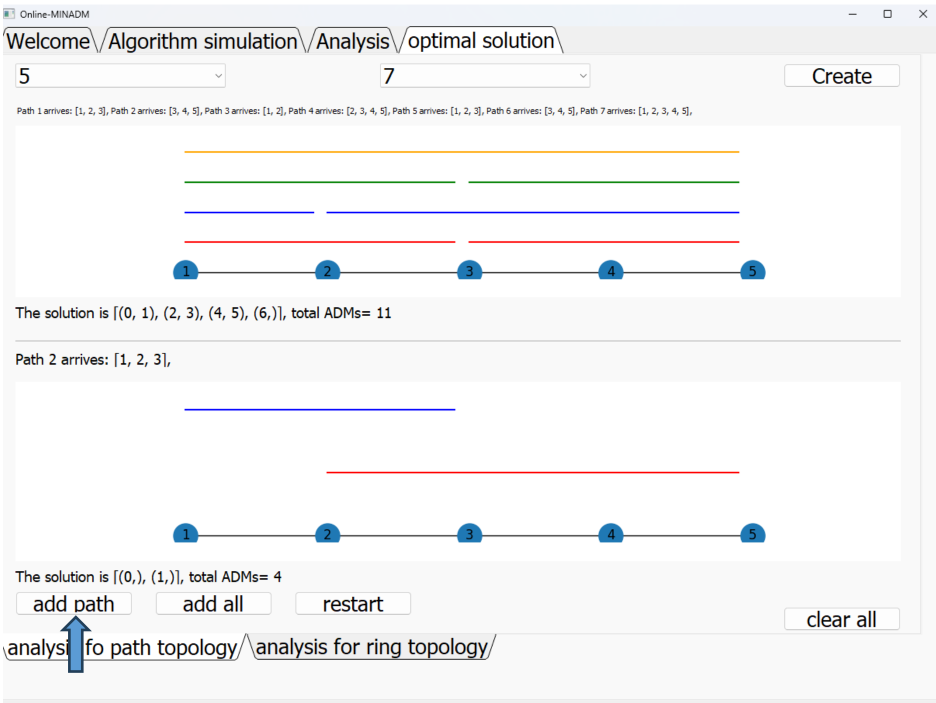
After that the program will enables the “Create” button, so the user can press it to generate new graphs.



In this case the program will generate and draw 2 path graphs with five nodes, the first one for the optimal solution and the second one for online solution. The optimal graph will contain 7 light paths that colored in the optimal way, as shown in the picture bellow.

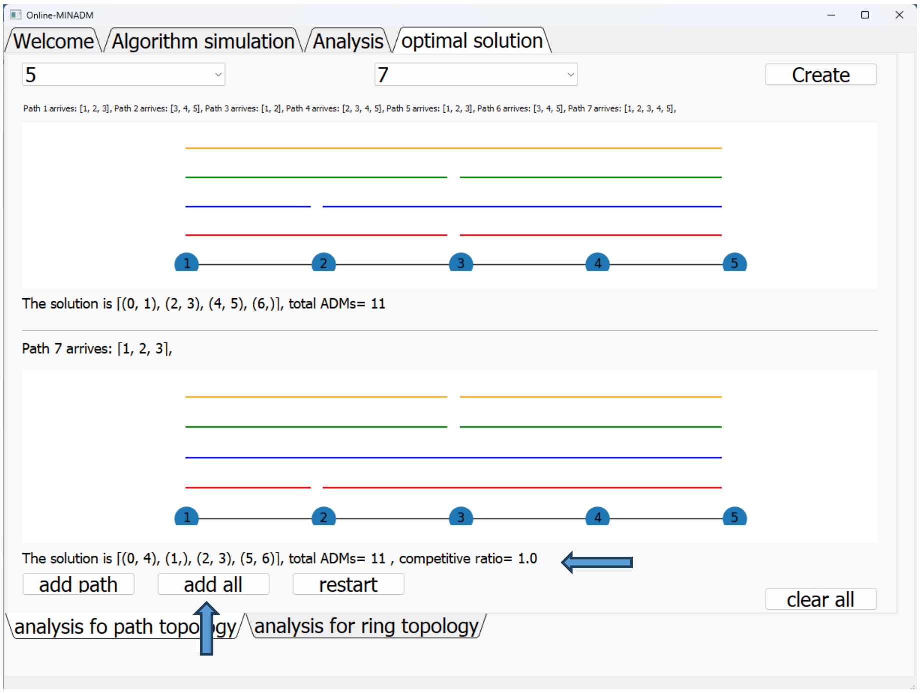


The user can add paths one by pressing on “add path” button, so the program will take a random path and add it to the graph.

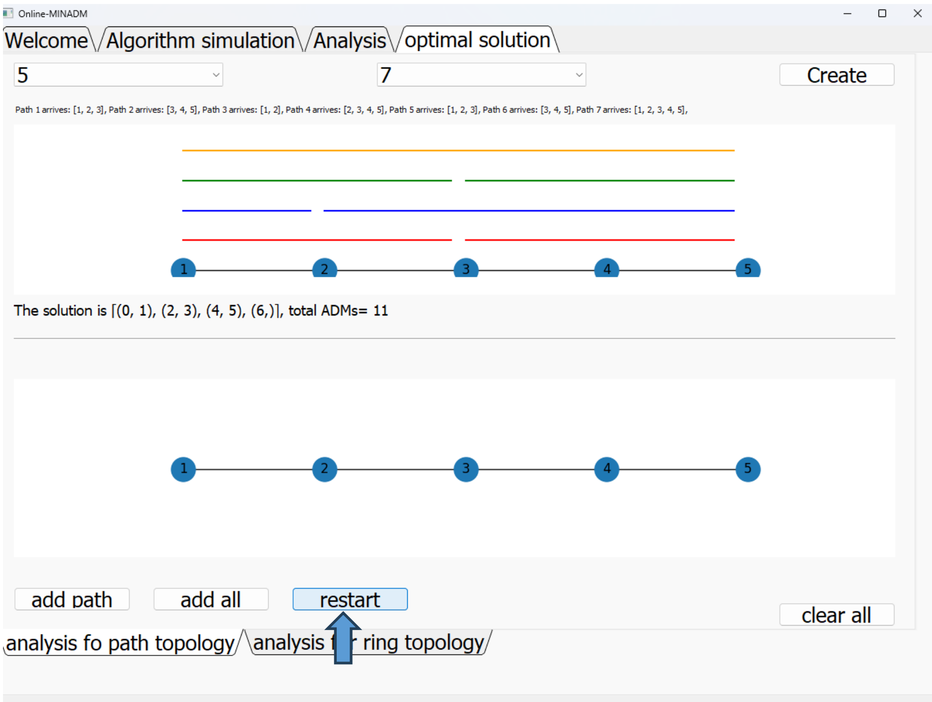


Also, the user can add all the paths in one click, by pressing on “add all” button.

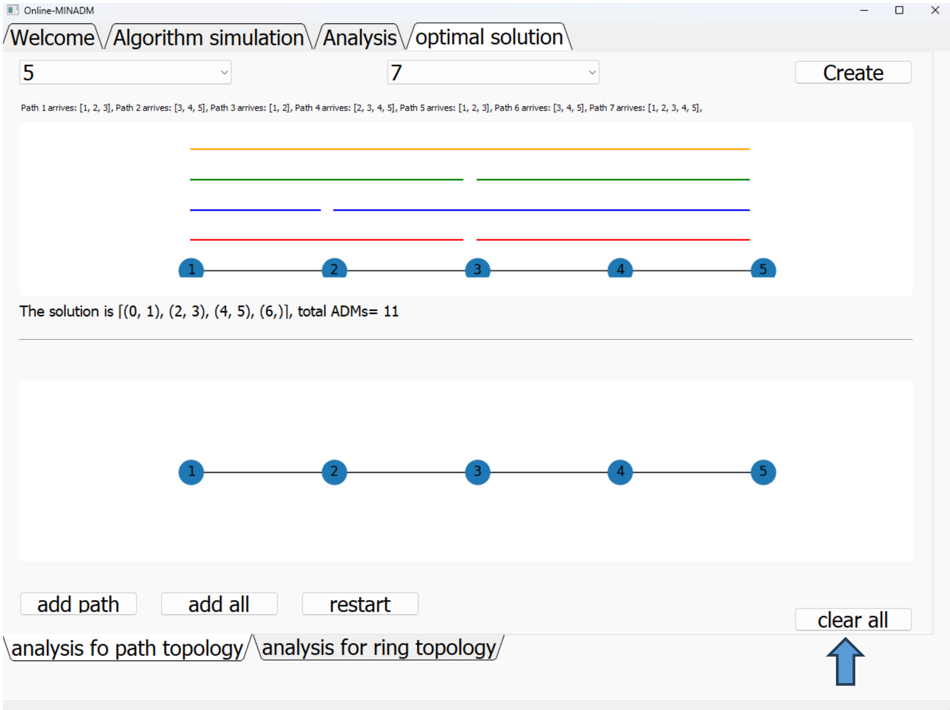
At the end, the program will calculate the competitive ratio and display it.



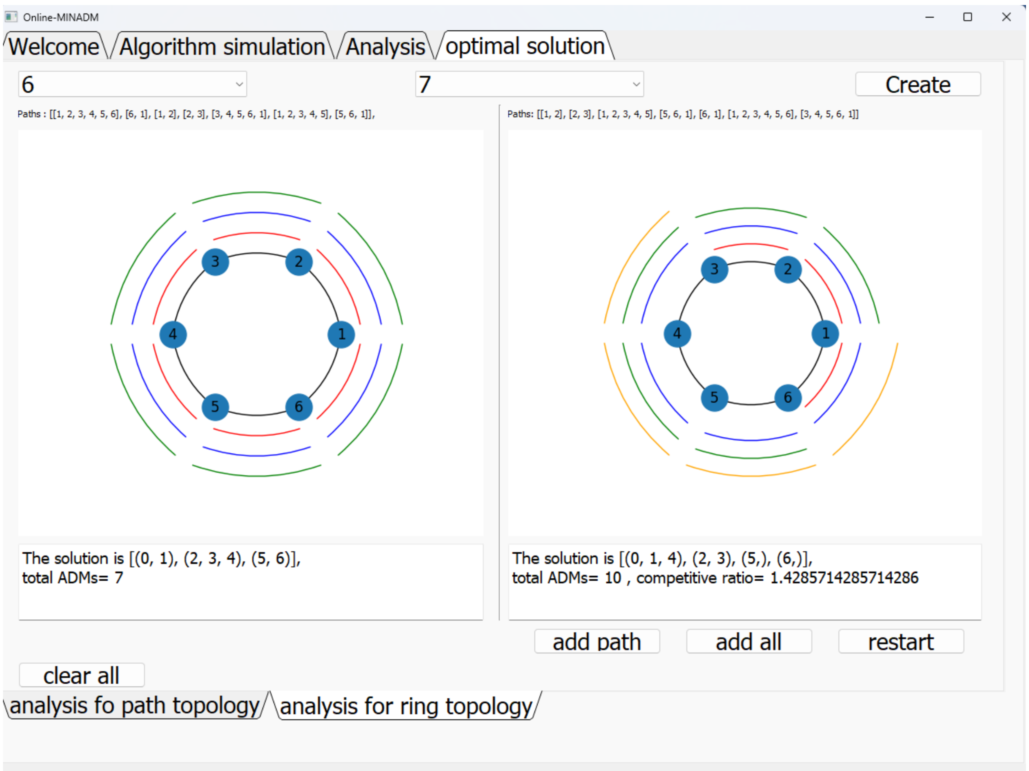
The user can reset the online graph by pressing on “restart button” and start again.



To start new experiment, the user can press on “clear all” button and repeat the same process or just recreate another graphs by pressing on “create” button and the program will clear all first automatically.



The user can do the same thing in ring topology by enter to "analysis for ring topology" tab.



In the "Analysis" tab, the software presents a 3D graph that showcases the relationship between the number of nodes (x-axis), the number of paths (y-axis), and the ratio between the average performance and the optimal solution (z-axis). The points in the graph are color-coded into three different shades of red based on the ratio values.

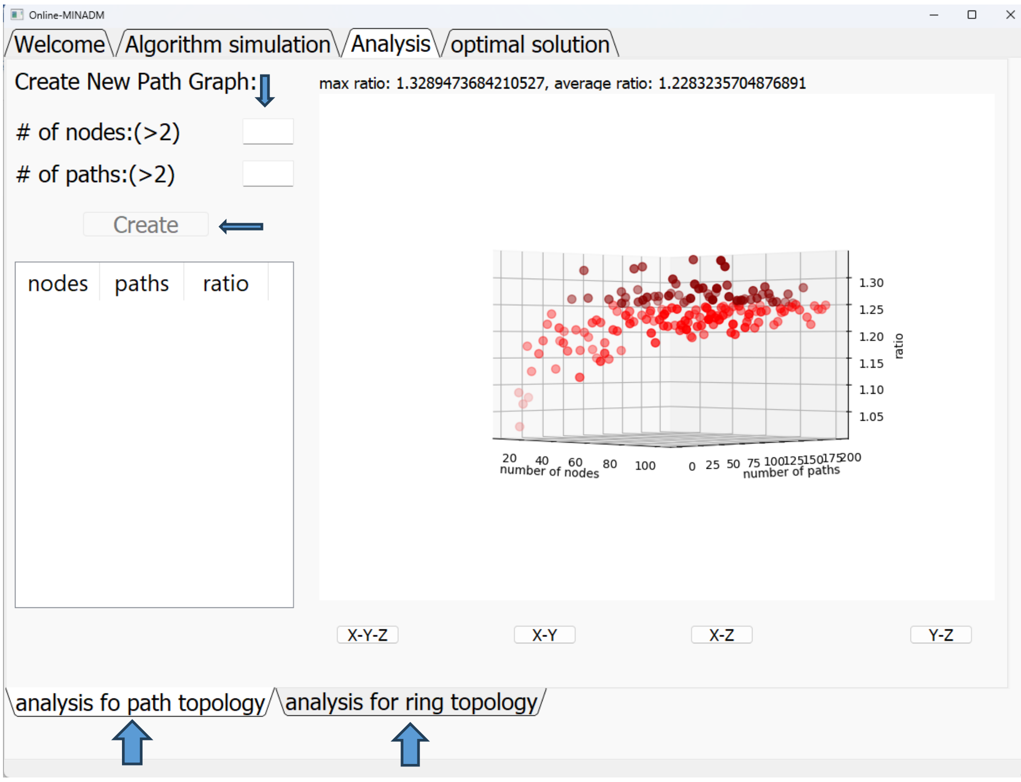
The color distinction provides visual differentiation for understanding the ratio variations. Typically, lighter shades of red may represent lower ratio values, while darker shades of red may indicate higher ratio values. This color-based representation allows users to identify and interpret the performance characteristics of the algorithm across different combinations of node and path numbers within the given network topology.

In the "Path Topology" analysis, points on the 3D graph are color-coded based on the ratio values. Points with a ratio (z) between 1 and 1.1 are represented by a light red color. Points with a ratio between 1.1 and 1.25 are depicted as bright red, while points with a ratio greater than 1.25 are shown as dark red. It is important to note that these color shades and ranges were chosen arbitrarily to emphasize the varying levels of ratios in the results.

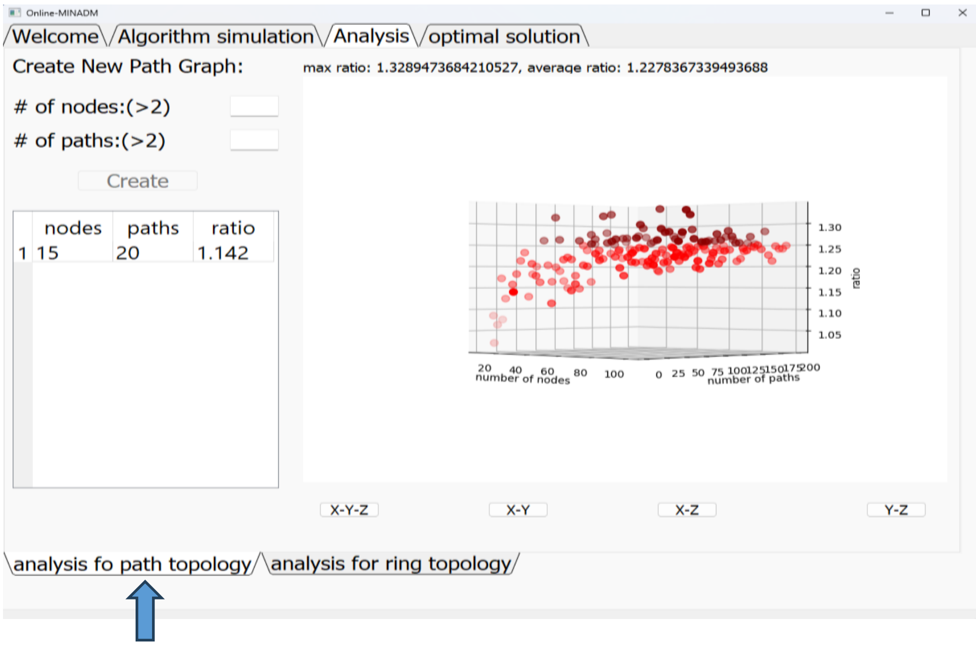
Similarly, in the "Ring Topology" analysis, the points are color-coded based on the ratio values. Points with a ratio between 1 and 1.1 are represented by a light red color. Points with a ratio between 1.1 and 1.3 are depicted as bright red, and points with a ratio greater than 1.3 are shown as dark red.

These results were obtained by running the algorithm on numerous networks of various sizes and calculating the average result. Each point (x, y, z) on the graph represents a network with x nodes, y paths arriving sequentially, and an average ratio of z. The average is calculated by running the algorithm on networks with x nodes while changing the order in which the paths arrive.

The 3D graph can be interactively explored by moving it around using the mouse. Additionally, specific angles of the graph can be viewed conveniently by using the provided buttons.

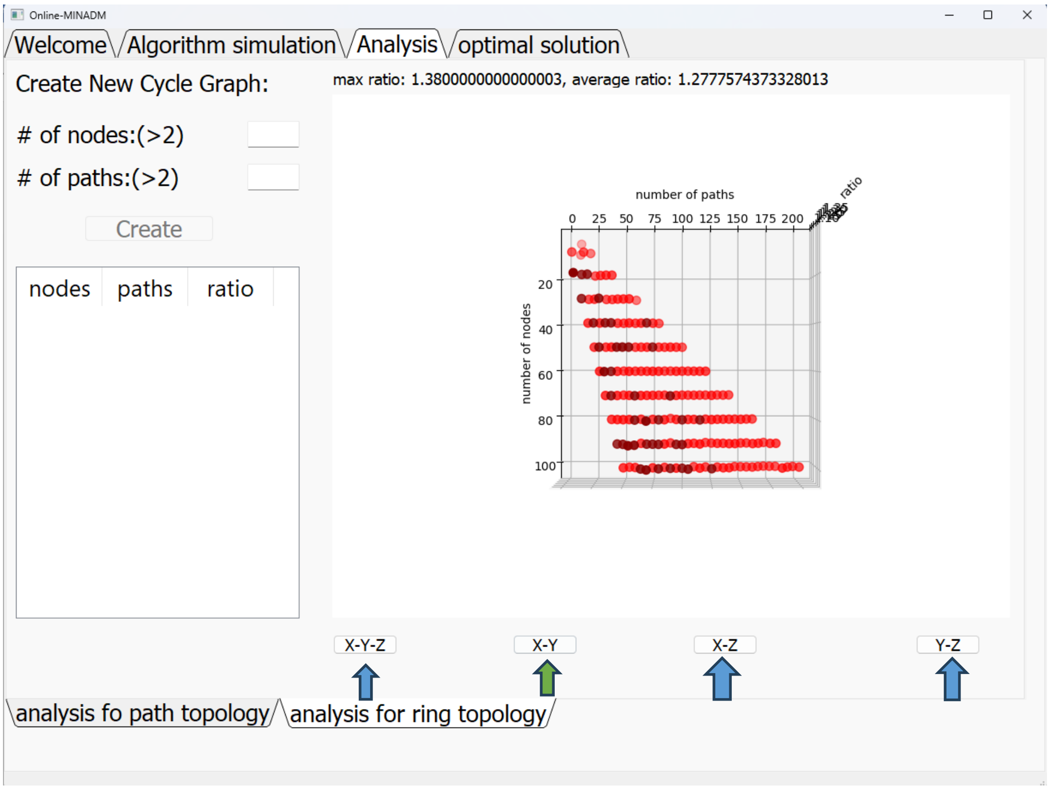


The user can choose between path topology and ring topology by clicking on the desired tab and see the results or add new point to the graph, to add new point to the graph the user needs fill input field (# of nodes, # of paths) and press on “create” button.



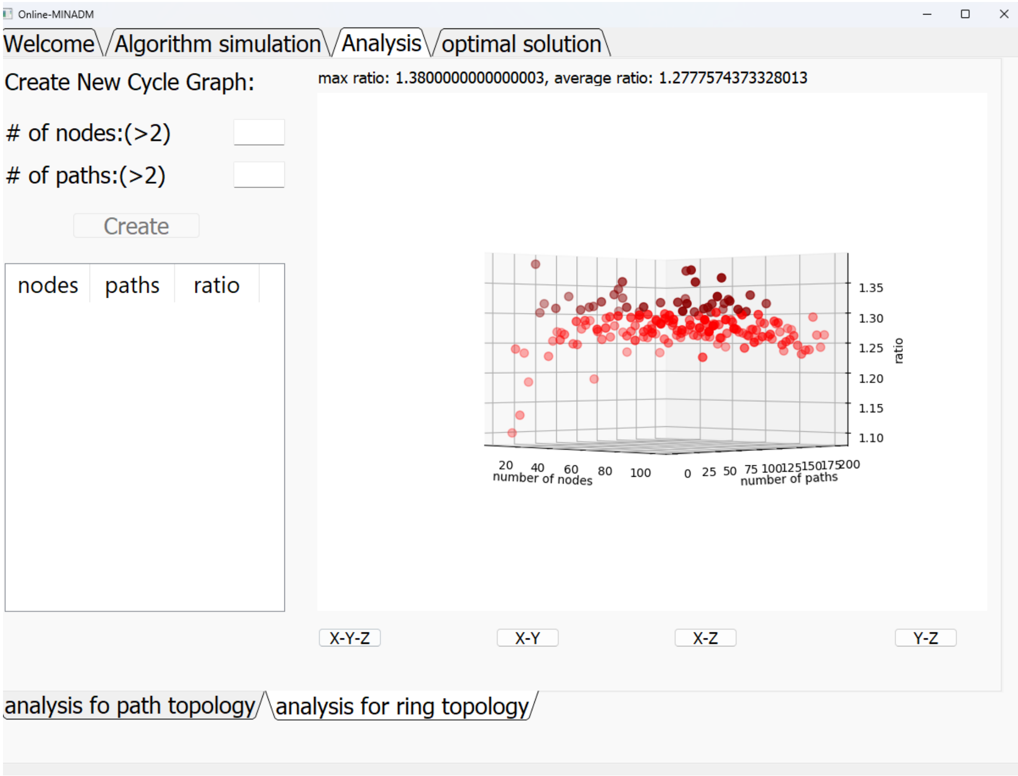
After pressing on the "create" button, the program will read the input fields and generate new graph with the specified number of nodes, create list of paths in length of the specified number of paths, the paths are arranged in the optimal so the algorithm can calculate the optimal solution.

After calculating the optimal solution the program will do 10 random runs then calculate the average and add the data to table and to the graph.



By clicking on the buttons ("X-Y-Z", "X-Y, "X-Z", "Y-Z"), the user can see different angels of the graph

The user can do the same for ring topology by click on the tab "analysis for ring topolog"


Online-MINADM
Welcome
Algorithm simulation
Analysis
optimal solution
Create New Cycle Graph:
# of nodes:(>2)
# of paths:(>2)
Create
nodes
paths
ratio
max ratio: 1.3800000000000003, average ratio: 1.2777574373328013
number of nodes
number of paths
ratio
X-Y-Z
X-Y
X-Z
Y-Z
analysis fo path topology
analysis for ring topology

3. References

[1] *Shalom, M., Wong, P. W,* & *Zaks,* S. *(2007, September). Optimal on-line colorings for minimizing the number of ADMs in optical networks. In International Symposium on Distributed Computing (pp. 435-449). Springer, Berlin, Heidelberg.*