Software Engineering Department  
Braude Academic College

**Capstone Project Phase B**

**Optimization of routers placements in wireless mesh networks**

**Supervisor**

Prof. Miri Weiss-Cohen

**Authors**

Marom Blumenfeld Omer Rogel

[Github link](https://github.com/MaromB/Capstone-Project)

**Contents**

1. **Abstract…………………………………………………………..……...… 3**
2. **Introduction …………………………………………………………..…… 3**

Project goals………………………….…………..………………….4

Application of Goals in the Project...………………………………4

Project Structure and Development Process…………….………5

User identification.….…………………………….……………...…6

1. **Project research…………………………….……………………......…… 7**

The process of research…………………………....…………..…..8

Research challenges………………………………………....……10

Research results……………………………………………………11

Research conclusions……………………………………………...14

Learning lessons……………………………………………………16

Project achievements………………………………………………16

1. **User guide …………………………………………..……….…….…….. 17**
2. **Maintaining guide ………………………………………………………...21**
3. **References ………………………………………………………………...23**

**1. Abstract**

When it comes to improving the performance of a wireless mesh network, the positioning of routers is critical. To provide the greatest network accessibility, it is essential that mesh routers be installed in appropriate locations. One way to improve the network’s performance is to increase the coverage while utilizing an optimum number of routers. The network coverage may be optimized using several different techniques that have been developed over time in the scientific literature. In this project, we explored the use of Genetic Algorithms (GA) and Particle Swarm Optimization (PSO) algorithms for optimizing router placement in WMNs.

GA and PSO are popular metaheuristic optimization techniques inspired by natural phenomena. GA operates by iteratively improving a population of candidate solutions using principles of natural selection and genetics. PSO, on the other hand, simulates the social behavior of bird flocking or fish schooling to optimize a population of candidate solutions by adjusting their positions in the search space based on the best-performing solutions found so far. In our research we constructed a program, with the language python, that optimize the placements of routers in a desired area while using the GA or PSO algorithm.

We were able to explore and compare their effectiveness in optimizing router placement in WMNs and These algorithms provided a systematic and efficient approach to address the complex optimization problem, leading to improved network performance and usability.

**2. Introduction**

Wireless Mesh Networks (WMNs) have emerged as a promising technology for providing wireless connectivity over a wide area by utilizing a network of mesh routers. These networks offer flexibility, scalability, and cost-effectiveness compared to traditional wired networks, making them ideal for various applications, including smart cities, industrial automation, and emergency response systems.

In this book, we provide a detailed description of the project's goals, the solutions we devised, and the methodologies and tools we employed to achieve them. We discuss the research and development processes, outlining the steps taken to build the system and the interactions with our client throughout the project.

We will delve into the details of the GA and PSO algorithms, their integration for router placement optimization, and the experimental results demonstrating the effectiveness of the proposed approach.

Furthermore, we present a user guide, offering operational instructions for utilizing the system in various scenarios. Additionally, we provide a maintenance guide, ensuring the continued use and support of the system post-completion.

**Project goals**

Our project aimed to achieve several key goals, each contributing to the overarching objective of enhancing Wireless Mesh Networks (WMNs) through innovative optimization techniques and user-centric design.

The primary goal was to develop a sophisticated algorithmic solution for optimizing the placement of routers in WMNs. By leveraging Genetic Algorithms (GA) and Particle Swarm Optimization (PSO), we sought to find the optimal configuration that would enhance network coverage, minimize interference, and improve overall network throughput.

We aimed to create a software program that not only offered powerful optimization algorithms but also featured a user-friendly interface. This interface would allow users to interact with the program intuitively, providing insights into network performance and router placement strategies in a clear and accessible manner. By providing users with a tool that could optimize router placement based on various parameters, we aimed to make the process of setting up and managing WMNs more efficient and effective.

Furthermore, our project aimed to contribute to the field of optimization algorithms and wireless networking. By exploring the use of GA and PSO in router placement optimization, we contribute to the body of knowledge in this area.

**Application of Goals in the Project**

in our project, we applied our goals by structuring our work around them, ensuring that each aspect of our project contributed to achieving these goals. Here's how we applied our goals.

We started by researching existing optimization algorithms and identifying GA and PSO as suitable techniques for our problem. We then implemented these algorithms in our program, carefully tuning them to optimize router placement based on coverage, interference, and traffic load.

we designed a user-friendly interface that allowed users to input network parameters and visualize the optimized router placement. we focused on making the program intuitive and easy to use. We implemented features such as interactive visualizations, real-time feedback, and user-friendly controls to enhance the overall user experience.

We designed our program to be scalable and adaptable to different network sizes and configurations. We implemented modular and flexible code structures that could easily accommodate changes and additions to the program's functionality. To elaborate more about this feature and how we used it in our program so we are giving the user the ability to select the amount of clients routers and size of area has he wishes but, for research reasons, we have examined two scenarios that we have seen that are being used in other related academical researches, 16 clients and 16 routers and an area of 32 x 32 or 42 x 42 in meters unit.

By applying all the above, our project is a testament to our commitment to our goals, as we applied our knowledge and skills to create a functional and impactful solution for optimizing router placement in WMNs.

**Project Structure and Development Process**

Our project followed a structured development process, starting with planning and research, followed by design, implementation, testing, and deployment. We held regular meetings to discuss progress, brainstorm ideas, and make decisions collaboratively.

We used various diagrams, such as flowcharts, UML diagrams, and network diagrams, to visualize our ideas and communicate them effectively. These diagrams helped us understand the system's architecture, identify potential issues, and make informed design decisions.

In developing our code, we followed best practices such as code reviews, testing, and documentation to ensure its quality and maintainability. We used version control systems to manage our codebase and collaborate efficiently.

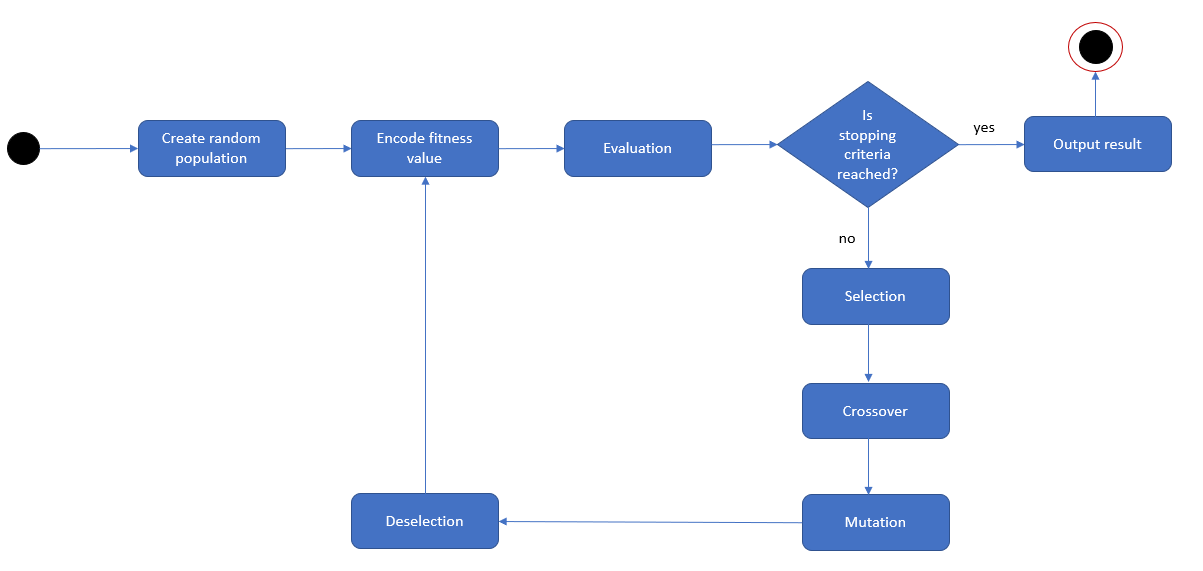


Figure 1 - we can see the activity diagrams for the operation of our GA implementation.A diagram of a flowchart

Description automatically generated

Figure 2 – here we can see the activity diagrams for the operation of our PSO implementation.

**User identification**

Our project aims to benefit a wide range of users and stakeholders involved in Wireless Mesh Networks and related fields.

Network Administrators and Engineers are the primary users who will directly benefit from our optimized router placement solution. By using our program, they can enhance the performance and reliability of WMNs, leading to improved network coverage and reduced interference. This will result in more efficient network operations and better service quality for end-users. In addition, our work contributes to the research community by offering insights into the application of GA and PSO algorithms in optimizing WMNs. Researchers can use our findings and methodologies to further advance the field of wireless networking and optimization algorithms.

Professionals working in industries that rely on WMNs, such as smart cities and industrial automation can benefit from our project. They can use our optimized router placement solution to improve the efficiency and effectiveness of their network infrastructure, leading to cost savings and better service delivery. That group leads us to our main targeted audience, the End-users.

While not directly interacting with our program, end users of services powered by WMNs, such as residents of smart cities or users of emergency response systems, will indirectly benefit from the improved network performance. They will experience better connectivity and more reliable services in their daily lives.

In creating our program, we have aimed to make it user-friendly and intuitive, catering to the needs of network administrators and engineers who may not have expertise in optimization algorithms. We have also focused on providing detailed insights and visualizations to help users understand the impact of router placement on network performance. Overall, our goal is to empower our users to make informed decisions that improve network efficiency and user experience.

**3. Project Research**

Our suggested solution addresses the challenge of efficiently deploying routers in WMNs to improve network coverage, reduce interference, and Suggested d optimize traffic load. Our approach combines the strengths of GA and PSO to find the optimal locations for router placement, considering various factors such as network topology, signal strength, and traffic patterns.

The core of our solution lies in the development of novel algorithms that leverage the principles of GA and PSO to iteratively improve the placement of routers. The GA component uses genetic operators such as selection, crossover, and mutation to evolve a population of candidate solutions towards the optimal placement. On the other hand, the PSO component models the movement of particles in a search space, with each particle representing a potential router placement. By adjusting the velocity and position of particles based on local and global best solutions, PSO guides the search towards promising regions of the solution space.

In constructing our program, we have designed a modular and scalable architecture that allows for easy integration of new algorithms and optimization techniques. The program consists of several components, including a network topology generator, a router placement optimizer, and a visualization module. The network topology generator creates a virtual environment simulating a real-world WMN, while the router placement optimizer uses GA or PSO to find the optimal router locations. The visualization module provides graphical representations of the network topology, router placements, and performance metrics, allowing users to analyze and evaluate the effectiveness of the optimization process.

The program operates in several stages. First, it generates a random initial population of router placements. Then, it evaluates the fitness of each placement based on predefined criteria such as network coverage, interference, and traffic load. Next, it applies GA or PSO algorithms to evolve the population towards the optimal solution. This process continues iteratively until a satisfactory solution is found or a predefined termination condition is met. During all of those stages the user will be able to see the current best solution, according to the fitness function, and additional visualizations that will give deeper insights about the process of the program.

**The process of research**

Our research process began with a thorough literature review to understand the existing approaches to router placement optimization in WMNs. This involved studying academic papers, technical articles, and relevant documentation to identify the state-of-the-art algorithms and methodologies. We also examined existing software tools and simulation platforms commonly used in the field of wireless networking research.

For constructing our visualized interface, we used packages such as Matplotlib and Tkinter. The GUIs were designed to display the results of the optimization algorithms in a clear and understandable format, providing users with insights into the optimal router placement for their wireless mesh networks. We Started with constructing the first window in which the user inputs his desire routers, clients and area, after that we have continued to our second window in which the user can visualize the progress of the algorithm and that required us getting into the following step which is developing the algorithms.

We utilized a variety of tools and technologies for algorithm development and testing. We used the programming language, Python, which is commonly used in the field of optimization and simulation and that allowed us to prototype and evaluate different algorithms quickly and efficiently. Some of the imports that were used in order to construct our algorithms were: random, threading, time, cv2, math and more. These libraries provided us with efficient tools for handling complex mathematical operations and implementing the genetic algorithm and particle swarm optimization. We also had to decide how to store the clients the routers and the area in terms of data structures we have eventually constructed our program as object oriented and stored the clients inside a class, so each client has an X and Y values for locations and a Boolean value the checks if the client is within a range of the routers. The router class also have an X and Y locations as well as a radius, that is being measured in meters, an ID, an amount of coverage valuable that indicates how many clients are within a certain router and a velocity variable for our PSO algorithm. And finally, our area class that has a height and a width, that are being measured in meters, and 2 lists, one for routers and one for clients. After establishing a base, with our object-oriented parameters, we need to start contracting the algorithms themselves. we first started with GA and then with PSO. The first step into developing this algorithm was evaluating thickness function after randomizing the clients and routers on the selected area. As both of our algorithms need to establish a certain value for how well the result is, the value of our fitness function, we had to decide for certain parameters that will be tested and checked as well as how much weight we give to each one of those parameters. In depth research and a discussion between both of us, the team members of the project, we established several key parameters that we wanted to examine while evaluating the fitness function. The first one was the coverage that specify the number of clients that have access to a router and therefore to the network. The second one is the giant component size that examine the connectivity of our mesh network according to the position of the routers between one another and their ability to communicate. 3rd was a penalty component that checks if our routers are too close to one another and that will decrease the value of the fitness function, routers that are too close together may cause interference. and 4th and finally we also considered a bonus parameter that, in contrast of the penalty, it will complement the fitness function when the distance between the routers is close enough for communications but not as much for interference. After evaluating the fitness score, we had to decide what were the proper weights for each one. To decide that we had to go through several trials with different weights for understanding how effective and how significant each value should be to get our optimal solution. We first started with an equal distribution of 25% for each, after several trials we have seen that the penalty component did not help while notify us when routers are too close to one another, and we decided to deduct this parameter and to give better weight to the bonus parameter. The trials after that gave us a visual representation of our optimal solutions that seemed too close together and that led us to give more weight for the coverage component. As expected, adding more weight to the coverage component gave us more coverage but the routers were not necessarily connected in a way that they would have the ability to communicate. After teamwork discussion, we have decided that instead of giving more weight back to the giant component variable we will give more weight to a new parameter that we are adding, an optimal-coverage parameter. The optimal-coverage parameter will calculate the optimal distribution of the routers in terms of area and not in terms of covering the clients, we will compare that optimal solution to our distribution of routers and that will be another value with another weight inside of our fitness function. Those fine tuning and further research have led us to the final fitness weights distribution that is different for each algorithm as followed:

The fitness function for our genetic algorithm:

And the fitness function for our particle swarm algorithm:

Did G represent the giant component size, C represents the coverage, B represents the bonus component and O represent the optimal coverage ratio for the routers.

With this depended-research development we started testing our results for several cases and conditions such as different size areas that are not only squared but also images. The results were gathered and are being presented in a table under research results later in the article.

Since our project primarily involved developing optimization algorithms, we did not have extensive interactions with users during the development process. However, we designed the interface with usability in mind, ensuring that users can easily input their network parameters and visualize the optimized router placements. Overall, our approach focused on creating a robust and efficient solution that can benefit a wide range of users in the field of wireless mesh networking.

**Research challenges**

To elaborate on the challenges faced during the research process, several key aspects came into play, including analytic problems, engineering challenges, and decision-making regarding data structures and algorithm implementation. Here's a more detailed overview:

Understanding the underlying principles of genetic algorithms (GAs) and particle swarm optimization (PSO) required a deep dive into algorithmic complexities. Deciding how to adapt these algorithms to the specific requirements of optimizing router placement in WMNs was a significant challenge. To address this, we conducted an in-depth study of genetic algorithms (GAs) and particle swarm optimization (PSO) algorithms. We consulted academic literature and research papers to understand the core principles behind these algorithms.

Implementing the GA and PSO algorithms in Python presented its own set of challenges. Ensuring the algorithms were efficient, scalable, and integrated seamlessly into the overall program structure was crucial. This involved optimizing the algorithms for performance and considering factors like memory management and computational efficiency. To tackle this, we adopted a systematic approach to algorithm implementation. We first developed a detailed design document outlining the structure of our program and the integration of GA and PSO algorithms. We then implemented the algorithms in Python. Regular code reviews and testing helped us identify and resolve any issues early in the development process.

Choosing the right data structures to represent routers, clients, and other components of the WMN was critical. This decision impacted the efficiency of the algorithms and the overall performance of the system. we chose to represent routers, clients, and other network components using Python classes. This allowed us to organize the data efficiently and simplify the implementation of the algorithms. We used to dictionary and lists to store network configurations and algorithm parameters, ensuring quick access and manipulation of data.

We also had to tackle the careful implementation to ensure correctness when it comes to the parameters that are being measured during the phase of evaluation of the fitness function and of course the weight that we give to each one of those parameters. After careful tuning and examination of the results for those tuning, we have managed to establish a satisfying selection of parameters and their weights for a desired fitness function that will lead to an optimal solution.

Developing a user-friendly interface that would allow users to interact with the optimization algorithms was another challenge. Deciding on the right libraries and frameworks for the interface, such as matplotlib and Tkinter, required careful consideration. Matplotlib allowed us to visualize network configurations and algorithm outputs, providing users with a clear understanding of the optimization process. Tkinter was used to create an interactive interface for users to input parameters and visualize results.

Balancing the workload between team members was crucial for the project's success. Ensuring that both team members were equally involved and contributing effectively required effective communication and collaboration. To manage workload distribution between team members, we divided tasks based on expertise and availability, ensuring that each team member had a clear understanding of their responsibilities. Regular check-ins and updates helped us stay on track and address any issues that arose during development.

Despite these challenges, our project team was able to overcome them through collaboration, research, and iterative development.

**Research results**

Our primary goal was to create a system that improves network coverage, reduces interference, and enhances overall network performance. After extensive research, algorithm development, and testing, we successfully achieved our goals by implementing a robust solution. Our application allows users to input network parameters such as the number of routers, number of clients, and network topology. The GA and PSO algorithms then optimize the placement of routers to maximize network coverage and minimize interference.

Another important discussion was around the user interface and how to make it intuitive and user-friendly. The interface provides real-time feedback on the optimization process, allowing users to visualize the network layout and monitor the algorithm's progress.

Discussing the output that were resulted from the algorithms, we have received coverage outputs for different wireless mesh network inputs regarding customers and routers that were decently satisfying. Most of the tests that we have run for our final product in the project as an output of a satisfying fitness function as been described in the images and data tables below.

A screenshot of a computer screen

Description automatically generated

Figure 3 - In the image above, you can see the result of our program after running the genetic algorithm. on the left side you can see the resulted coverage and on the right side you could see an axis that represents the fitness functions pair iterations,  
And in the following image you can see the same result after running our program with the PSO algorithm.

A screenshot of a computer

Description automatically generated

Figure 4 – here we can see the result of our program after running the PSO algorithm.

In both images we can see the coverage that was received was well established, around 75% or higher and a fitness score that is about 80% or higher.

And in the data tables below there are several iterations for the selected area for our research with the selected number of clients and routers. Note that the number of iterations that we chose is 300 iterations and that is due to the observation that, after several trials, we have noticed that beyond that amount it requires a lot more iterations in order to get an improvement that sometimes is not as exceptional. Let us begin by presenting the results for our GA algorithm.

|  |  |  |  |
| --- | --- | --- | --- |
| size | Coverage | giant component size | fitness value |
| 42x42 | 87.5% | 6 | 84.3% |
| 42x42 | 100% | 3 | 82.1% |
| 42x42 | 82% | 4 | 81.1% |
| Average | **89.3** | **|4.3| = 4** | **82.5** |

|  |  |  |  |
| --- | --- | --- | --- |
| size | Coverage | giant component size | fitness value |
| 36x36 | 93% | 6 | 88.9% |
| 36x36 | 81% | 7 | 87.8% |
| 36x36 | 89% | 6 | 88.3% |
| Average | **87.6** | **|6.3| = 6** | **88.3** |

And now let’s present the data tables for our PSO algorithm for optimal solution.

|  |  |  |  |
| --- | --- | --- | --- |
| size | Coverage | giant component size | fitness value |
| 42x42 | 81% | 8 | 88.5% |
| 42x42 | 77% | 8 | 87.8% |
| 42x42 | 75% | 9 | 88.9% |
| Average | **77.6** | **|8.3| = 8** | **88.4** |

|  |  |  |  |
| --- | --- | --- | --- |
| size | Coverage | giant component size | fitness value |
| 36x36 | 85% | 8 | 91.1% |
| 36x36 | 83% | 9 | 92.8% |
| 36x36 | 83% | 13 | 93.4 |
| Average | **83.6** | **10** | **92.4** |

**Research conclusions**

The research phase of our project yielded several key conclusions that have significantly influenced the development and direction of our work.

Specifically, each one of our algorithms help us get to several conclusions regarding our development, research, and results.

Discussing our genetic algorithm, We have found that while trying to get one large mesh network for all of our routers we’ve got several mesh networks that were spread around the area while trying to create a mesh network for each client distribution that were randomly Distributed before, and that leads us to the finding that the GA algorithm was leaning towards the coverage of the clients and not into the creation of a one big mesh network. That finding Leads us to another conclusion that if there is a need to use the GA algorithm for distributing a network it can come into good use When one is trying to establish several mesh networks for a desired area that could have several mesh clients within. Regarding the final output, we have noticed that when looking in the sub-network that was created, we could see that this distribution of the routers was quite fair although there seemed to be overlapping routers that could cause interference and should be removed to a different location if one is planning a real-life application of this solution. While observing the process of creating our final solution for a certain input, we have also noticed that the algorithm could take several steps between the first initialized guest and the final solution so there could be several optimal solutions that are getting better and better with an average of discovering a new solution each 25 to 50 Iterations for the first 200 Iterations.

Discussing our particle swarm optimization algorithm, the final solutions that we got for inputs, that were the same for the GA algorithm, have given us better results in terms of giant component size and therefore a well-connected network as well as a better fitness score. We did notice that when spreading the clients randomly the coverage seems to be less than GA algorithm although when manually entering clients, we did manage to get a satisfying result. That information leads us to the conclusion that if one is planning to have a connected mesh network that is mostly focused about the connectivity and less about the coverage it is recommended to choose this algorithm. While looking at the graph of the fitness scores pair iterations we saw that it might takes longer for the PSO algorithm to find a new solution although the new solution will have a better fitness score than the one that can be reached from the GA algorithm, and that concludes us that PSO converge faster than GA. Another conclusion that was specifically related to the PSO algorithm was the fine tuning for the weights of the fitness function. We have noticed that even a change of .05 percent from one parameter to another will affect the behavior of the algorithm and therefore the solutions that we will get. This phenomenon was not something we notice in our GA implementation and fine tuning.

One common conclusion drawn from our study is the necessity for each algorithm to have a unique fitness function with carefully tuned weights. This is because the nature of optimization varies between the GA and PSO. In the beginning of our trials, while trying to find the right tuning for each, we started with an equal tuning for both fitness function and then we started reaching to the correct tuning for the GA algorithm. After that, we tried applying the same configurations for the PSO algorithm and we have noticed that the results were not as good and a different and unique tuning was required. The final conclusion from the above is that although both of the algorithms have the same goal, they will need different care for each sub goal.

As a result, the fitness functions for each algorithm must be tailored to these specific objectives. This highlights the importance of fine-tuning the fitness function for each algorithm to achieve optimal results.

Furthermore, we have general conclusions about conducting an academical research. One of the primary findings was the ease of splitting the work between team members, demonstrating the effectiveness of our collaboration. This experience highlighted the importance of clear communication and defined roles within the team, ensuring that tasks were completed efficiently and effectively.

Searching through academic articles proved to be invaluable for gaining a deeper understanding of the field and the development of our project. It allowed us to leverage existing research and incorporate proven methods and algorithms into our work. Additionally, it provided us with insights into the latest advancements in the field, helping us stay abreast of current trends and technologies.

Python emerged as a well-chosen language for developing our algorithm and interface. Its versatility and extensive libraries, such as NumPy, SciPy, and Matplotlib, provided us with powerful tools for implementing complex algorithms and visualizing our results. The decision to use Python enabled us to focus more on the algorithm's logic and less on low-level implementation details, resulting in a more streamlined development process.

In summary, the research phase not only provided us with the necessary foundation for our project but also offered valuable insights and lessons that will be instrumental in guiding our future work. The GA tended to create multiple mesh networks distributed across the area, prioritizing client coverage over a single large mesh network. This makes it suitable for establishing several mesh networks in a desired area with multiple clients. However, the PSO algorithm yielded better results in terms of giant component size and overall network connectivity. It was more focused on connectivity rather than coverage, making it ideal for scenarios where network connectivity is crucial. Fine-tuning the weights of the fitness function was critical for the PSO algorithm, as even a slight change could significantly impact the algorithm's behavior and the solutions obtained. Additionally, both algorithms showed iterative improvement, with the PSO algorithm requiring more iterations but yielding better fitness scores in the end.

Through effective collaboration, the selection of appropriate tools and technologies, and a thorough exploration of academic literature, we were able to lay a solid groundwork for the development phase of our project.

**Learning lessons**

The research phase of our project provided us with valuable lessons that have shaped our approach and understanding of the project. One of the key lessons was the importance of effective time management and task distribution. While splitting the work between team members proved to be a successful strategy, we realized that it might have been beneficial to spend more time learning and exploring together before dividing the tasks.

In hindsight, another important lesson was the value of clear communication and documentation. Maintaining detailed records of our research findings, decisions, and progress proved to be essential for staying organized and ensuring that everyone was on the same page. Additionally, regular meetings and discussions helped us stay aligned and address any challenges or issues that arose during the research phase. Being open to new ideas and willing to modify our plans, when necessary, allowed us to overcome these challenges and continue making progress towards our goals.

The research phase taught us valuable lessons about teamwork, time management, communication, and adaptability.

**Project achievements**

When we embarked on this project, we set out with several key goals in mind. One of our primary objectives was to explore and research two powerful optimization algorithms, Particle Swarm Optimization (PSO) and Genetic Algorithms (GA). Through extensive study and experimentation, we gained a deep understanding of these algorithms and their applications in optimizing router placement in wireless mesh networks.

Another goal was to develop a functional program that could effectively demonstrate the use of these algorithms in real-world scenarios. We aimed to create a user-friendly interface that would allow users to input their network parameters and visualize the optimal router placement suggested by the algorithms. By achieving this goal, we hoped to provide users with a practical tool for designing efficient wireless mesh networks. Additionally, we aimed to evaluate the performance of the PSO and GA algorithms in terms of their ability to optimize router placement. Through rigorous testing and analysis, we were able to assess the strengths and weaknesses of each algorithm and determine their suitability for our network scenarios.

**4. User guide**

The user guide for our program compresses four initial steps firstly we would choose the number of clients that is being desired followed by the second step that is to select the number of routers, The third step is selecting the desired algorithm, selecting in between GA and PSO, and then the final step is to select the desired area or create one with width and height dimensions. Another additional step is the ability to choose between randomizing the position of the clients or to choose it manual on top of the area.

A screenshot of a computer

Description automatically generated

Selecting the number of routers, clients, and the desired algorithm.

Selecting between a pre uploaded image or creating an area.

A blue screen with a blue text box and a blue square with a blue text box and a blue square with a blue square with a white square with a black text box and a white square with

Description automatically generated

For preloaded images we can randomly distribute clients or manually select their location

A blue screen with a blue text and a blue square

Description automatically generated with medium confidence

For minimal distribution select the location of the clients by clicking on the image.

A screenshot of a computer

Description automatically generated

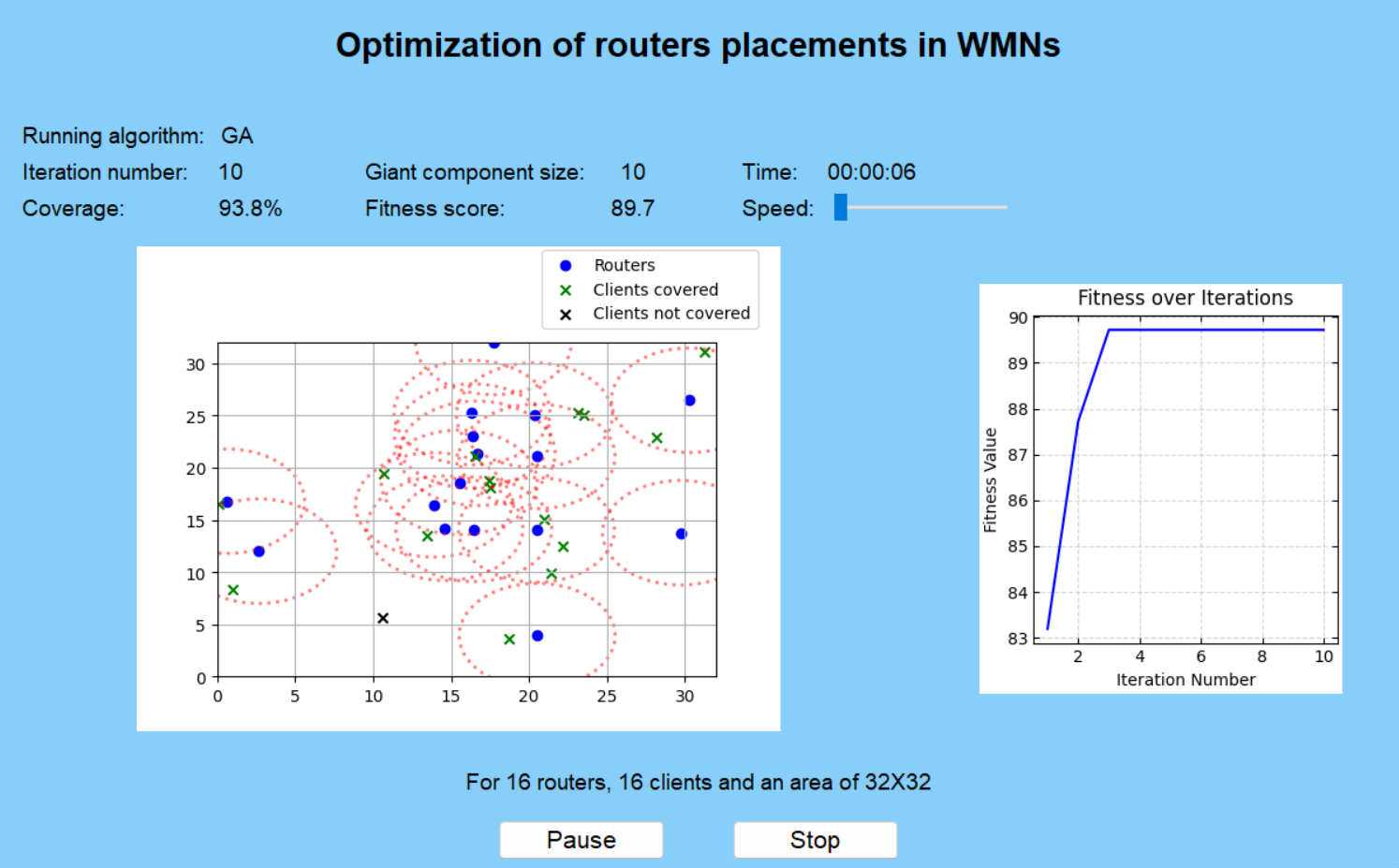
Here as well there is an option for random or manual distribution of clients.

For a rectangle shape select and input the desired height and width.

A screenshot of a computer

Description automatically generated

After selecting the width and height with manual distribution select the position of the clients by clicking on the image



visualized results

After starting the selected optimization algorithm, we will be able to monitor the progress of the algorithm as well as the best solution so far according to the fitness score and additional parameters such as selected input time coverage and more.

**5. Maintenance guide**

To preserve the continuous use of the product there are several aspects that we can consider for future use. Talking about the fundamental aspects of our program it was based on one programming language Python that was being developed in the environment PyCharm, besides that the final product can be extracted to an EXE file so to use the product all you have to do is just run the program and the rest will follow, The interface and the algorithm behind it.

**Updates and improvements**

In our project, there are several areas that we can update and improve to enhance its functionality, performance, and usability. Some of these areas include:

Continuously improving the genetic algorithms (GA) and particle swarm optimization (PSO) algorithms to enhance their efficiency and effectiveness in optimizing router placement in wireless mesh networks. This can involve refining the algorithm parameters, exploring new algorithm variations, and incorporating machine learning techniques for better optimization. For example, Our program is fully functional while checking the desired variables that we have selected for our fitness function which are the giant component size, coverage, and a penalty. For future improvements, We can modify the variables for our fitness function as desired, it is possible to update and change them so the fitness function will check alternative features in our outputs as well as weights.

That leads us to another section that can be updated and improve which is the performance optimization. We can reduce computation time while creating the algorithms calculations in a more efficient way, those improvements can be applied with different data structure and libraries that Python provides us for faster calculations another important aspect that needs to be taken into account when it comes to the computation time is the visualization part. We can highly improve the computation time and reach an optimal solution faster if we just want to view the final output without seeing the progress of the algorithm from the first time it’s been initialized till the appearance of the final solution.

Enhancing the user interface (UI) of the application to make it more intuitive, user-friendly, and visually appealing. This can include adding interactive features, improving data visualization, and optimizing the UI for different screen sizes and devices. We can also handle errors and logging mechanisms to provide better feedback to users and developers. This can involve implementing robust error handling strategies, and error messages.

**6. References**

[1]Rastin Pries, Barbara Staehle, Dirk Staehle, Phuoc Tran-Gia.(2010, January). **On Optimization of Wireless Mesh Networks using Genetic Algorithms.**

[2]Shinji Sakamoto, Tetsuya Oda.(2016, April). **Implementation and Evaluation of a Simulation System Based on Particle Swarm Optimization for Node Placement Problem in Wireless Mesh Networks.**

[3]Tetsuya Oda, Aoto Hirata, Nobuki Saito ,Kazuho Kanahara.(2020, May). **Approach of a Solution Construction Method for Mesh Router Placement Optimization Problem.**

[4]SYLIA MEKHMOUKH TALEB, YASSINE MERAIHI, ASMA BENMESSAOUD GABIS.(2022, April). **Solving the Mesh Router Nodes Placement in Wireless Mesh Networks Using Coyote Optimization Algorithm.**