Optimized Sensor Placement using Evolutionary Algorithm

Abeera Alam
Department of Computer Engineering
Habib University
Karachi, Pakistan
aa05430@st.habib.edu.pk

Sandesh Kumar
Department of Computer Science
Habib University
Karachi, Pakistan
sk05567@st.habib.edu.pk

Syeda Saleha Raza
Department of Computer Science
Habib University
Karachi, Pakistan
saleha.raza@sse.habib.edu.pk

***Abstract*—This paper presents a genetic algorithm (GA)-based approach to optimize sensor placement in a room for environmental monitoring. The optimization problem is formulated as a multi-objective optimization problem, where the objective is to maximize the coverage of the monitored area. The GA algorithm is applied to search for the optimal sensor placement by generating a population of candidate solutions and evolving the population through selection, crossover, and mutation operations. The fitness of each candidate solution is evaluated based on sensor's coverage at each point in the room. The proposed method is evaluated on multiple simulated rooms with various sensor configurations keep in consideration decrease in coverage of a sensor due to obstacles and range.**

***Index Terms* — Evolutionary Algorithm, Sensor Placement Optimization, Computational Intelligence, Environmental Monitoring**

I. Introduction

Optimizing sensor placement is a critical element in constructing engineering systems including HVAC, security and environment systems. By enabling optimal system usage, reducing downtime, and avoiding catastrophic failures, optimized sensor placement (OSP) improves the overall performance of a system [1]. Placing the sensors to maximize the room coverage based on the number of sensors that are available ensures that the resources allocated to a space are utilized in the most efficient manner. Without optimizing their placements, the number of sensors used may be deemed low to cover the entire room and so more sensors would need to be added. Moreover, the number of obstacles in a room may hinder the performance of sensors and therefore it becomes important to consider their effect as well on the sensors' coverage.

This paper uses the evolutionary algorithm approach to solve this problem. A configuration of a room is taken, this includes the size of the room and obstacles placed in the room, and multiple configurations of this room are then generated with the assigned number of sensors placed at different locations for the initial population. The genetic algorithm then performs crossover, mutation, fitness evaluation, survivor selection to converge to an optimal solution. This solution represents the grid configuration of the room where sensors are placed optimally.

Traditional optimization methods, such as gradient descent or linear programming, may not be suitable for this problem due to the large search space and nonlinear relationships between the sensor placement and coverage.[4] Therefore, GA is an attractive approach because it can handle a large number of variables and constraints and can search for a globally optimal solution in a relatively short time. Our GA-based approach considers various factors, such as coverage area, obstacle effects, and the number of sensors available, resulting in a more comprehensive and balanced solution than traditional methods. Therefore, using a GA-based approach is a promising method for optimizing sensor placement in a room for environmental monitoring.

II. Background

A. Genetic Algorithm Optimization

Genetic Algorithms draw inspiration from nature and mimics the evolutionary process. They were first introduced in 1960 by John Holland as a way to efficiently compute solutions to complex problems that are not practical to compute using traditional methods. Traditional methods for sensor placement optimization are often computationally expensive and time-consuming, involving exhaustive search or heuristic algorithms. Evolutionary process have 4 components that it performs on an initialized population - parent selection, crossover, mutation and survivor selection. [2]

The algorithm works by selecting twice the number of off springs that are needed to be generated using a selection scheme. Crossover is performed on these parents; some composition of parent1 is exchanged with parent2 and vice versa to generate off springs. Mutation is performed on these off springs, mutation works by making a small modification on these off springs to introduce diversity and prevent early convergence. The off springs are then added to the population and some members with a low fitness value are removed from the population through survivor selection. This process is repeated for a number of generations until the fitness values converge to an optimal solution.

GAs have proven to be useful in many fields, including engineering, finance, and biology, due to their ability to handle complex, multi-objective optimization problems with large search spaces. In the engineering field, GAs have been applied to a wide range of problems, such as optimal design of structures, control systems, and sensor placement.

B. Similar Work

There has been considerable research in this domain. Similar to our approach, [5] discusses a method to design a cost-optimal sensor system for a certain diagnosability degree using an evolutionary approach. The efficiency of a diagnosis system depends on the relevance of the information it can retrieve from the diagnosed plant, and the efficiency of a sensor system can be measured by the diagnosability degree it provides. The paper presents an application example. [6] presents a genetic algorithm approach to finding the optimal placement of lamps in a room, such that most points are maximally lit while minimizing the number of lamps. The approach has applications beyond just rooms and lamps, like placing cameras, cellphone towers, satellites or in any other setting that involves emitters of some signals and needs to minimize their number whilst also maximizing coverage.

III. Problem Description

Our design aims to optimize the placement of sensors in a given room configuration. The room is modeled as an N x M grid with Y number of obstacles of varying dimensions. For our implementation, we have limited the placement of sensors to the boundaries of the room. In each iteration of the algorithm, the sensor positions are updated and the overall sensor strength at each point is evaluated. The configurations that maximize the sensor strength are considered to be more fit.

Figure 1 shows an example of a 10x30 room configuration with 4 hexagonal obstacles placed inside it.

To execute the optimization, we have set up the following parameters:

- Population Size: the number of individuals in each generation of the algorithm.
- Number of Generations : the total number of generations that the algorithm will run for.
- Mutation Rate: the probability that a mutation will occur in an individual's DNA during reproduction.
- Number of Offsprings: the number of offspring that will be produced from the fittest individuals in each generation.

By varying these parameters, we can achieve different levels of optimization and computational efficiency.

IV. Problem Formulation

A. Chromosome Representation

In our implementation, we represent each chromosome as a list of coordinates where a sensor can be placed. The coordinates are represented as (x, y) pairs, which indicate the position of a sensor in the two-dimensional space of the environment. We have restricted the placement of sensors in our implementation to only be set up at the boundaries of the room.

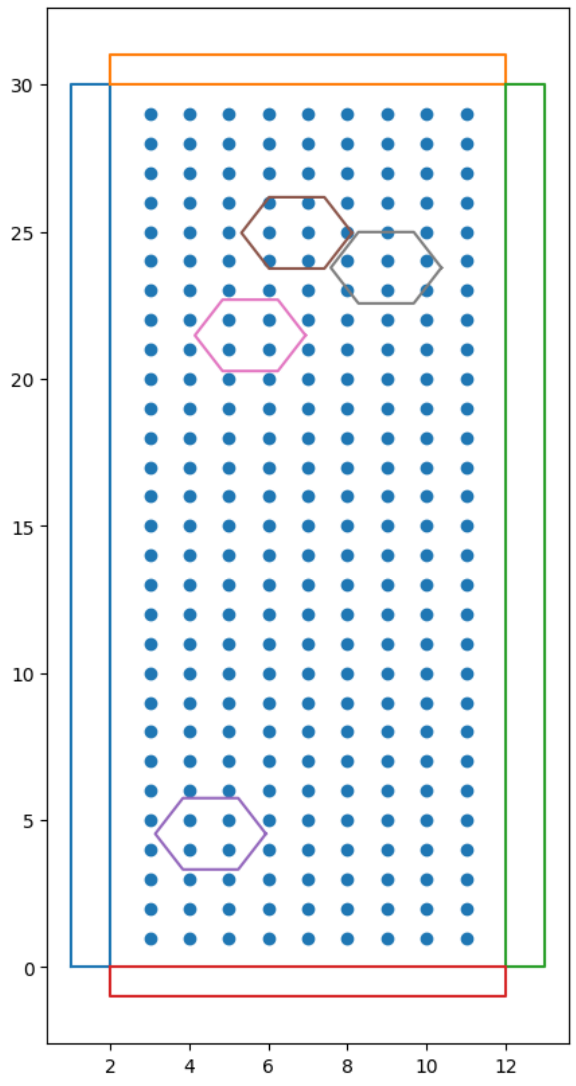

Fig. 1. Room Configuration

B. Crossover

Crossover is a genetic operator used in evolutionary algorithms for recombining genetic material from two parent solutions to produce new offspring solutions. In our implementation, the Crossover function is used to produce a new offspring by combining the sensor placements of two existing objects, parent1 and parent2.

The function first obtains the y-coordinates of the sensors in both parents as movement in x axis is restricted to ensure that sensors are only placed at the walls of the room. The function then chooses a random crossover point. The chosen point determines where the crossover will occur, and the sensor placements before the crossover point are taken from parent1, while those after the crossover point are taken from parent2. The resulting sensor placements are then combined to produce a new set of sensors, which is used to create a new member for the population.

C. Mutation

The mutation function in our implementation randomly selects one sensor from the individual's list of sensors and changes its y-coordinate to a new random position, making sure it does not collide with any walls and obstacles. This process introduces new genetic material into the population, which helps explore the search space and avoid premature convergence to sub optimal solutions. The mutation is rate is set as 0.5 as a high mutation rate may lead to random and non convergent behavior, while a low mutation rate may limit the exploration of the search space.

D. Fitness Function

The fitness function is designed to evaluate the quality of a particular sensor grid configuration based on its ability to provide effective signal coverage in a given environment. It takes into account the positions of sensors, the location of walls, and the characteristics of the signal propagation.
The fitness value is computed by iterating over all possible positions in the environment and evaluating the signal strength of the sensor grid configuration at each position. The maximum signal strength among all sensors at a particular position is recorded. The fitness value is the sum of the maximum signal strengths at all positions.
The function for calculating signal strength considers both air and wall attenuation. The signal strength decreases logarithmically with the distance travelled through the air or a wall, and a wall has a greater impact on the signal strength. The calculation takes into account the number and location of wall intersections that a signal must travel through to reach a position. The following formula was used for this:
$Signal\ Strength = Initial\ Strength \times (\log(Distance\ from\ wall)) \times Wall\ Fade \times \log(Total\ Distance\ in\ Air) \times Air\ Fade$

V. Experimentation and Results

We evolved 4 variations of grid . All variations are run twice firstly with objects and secondly without objects. Below given are the details and results of each variation.
For first variation, we set up a grid of size 10x30 units. The algorithm was run with initial population of 30 and number of offspring selected was 10. It was evolved for 150 generations and mutation rate was set to 0.5. This setup contains a total of 4 objects placed randomly in the grid. For parent selection, Fitness Proportional Selection was used, whereas for survivor selection, we used Binary Tournament Selection.

First setup was of a grid 10x10 without objects. Figure 2 shows the placement of sensors after evolution. Figure 3 shows intensity of those signals in the grid.

The grid of same size was again evolved with same parameters but with objects. The grid contained 3 objects while evolving. Figure 4 shows the position of sensors and figure 5 the intensity of signals with these parameters. We can see that in both the situations, the algorithm is trying to cover maximum area which is being shown by intensity. This is a good result for our paper.

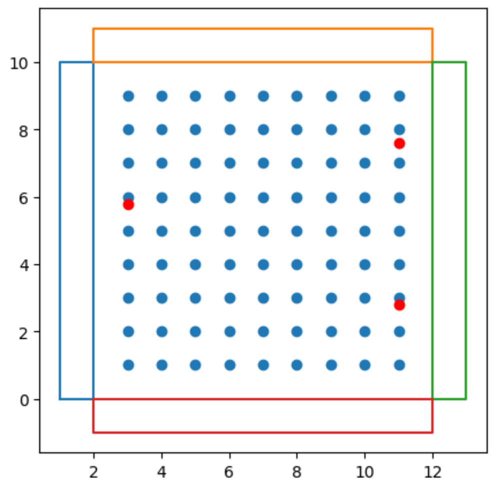

Fig. 2. Positions of sensors in grid of size 10x10 without objects

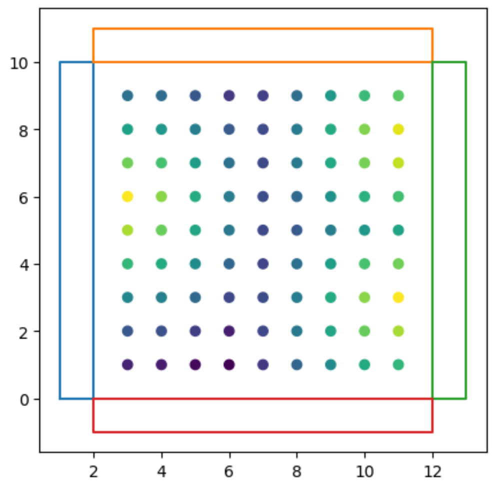

Fig. 3. Intensity of Signal in grid of size 10x10 without objects

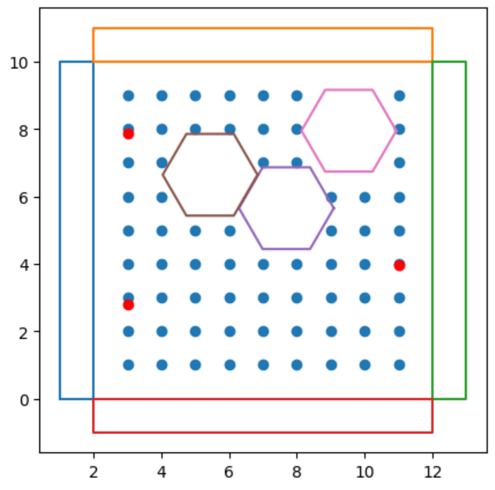

Fig. 4. Positions of sensors in grid of size 10x10 with objects

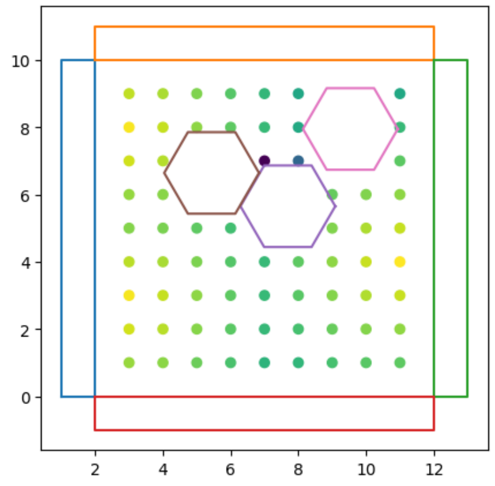
Fig. 5. Intensity of Signal in grid of size 10x10 with objects

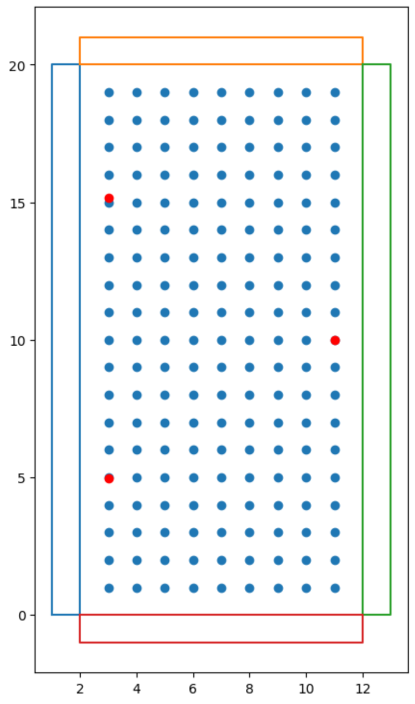
Fig. 6. Positions of sensors in grid of size 10x20 without objects

The second setup was of grid of size 10x20. The grid was evolved for 150 generations again. Figure 6 shows the positions of sensors and figure 7 the intensity of signal when there were no objects in this grid. We can see in the figure 7 that it is trying to maximize the intensity by placing them on different sides and then creating enough distance between them so that maximum area is covered.

Figure 8 and 9 show the results when objects were added. Figure 8 showing number of positions and 9 showing intensity. A total of 3 objects were added while evolving.

The third setup was of Grid of Size 10x30. Figure 10 and 11 show the results when no objects were added. Figure 10

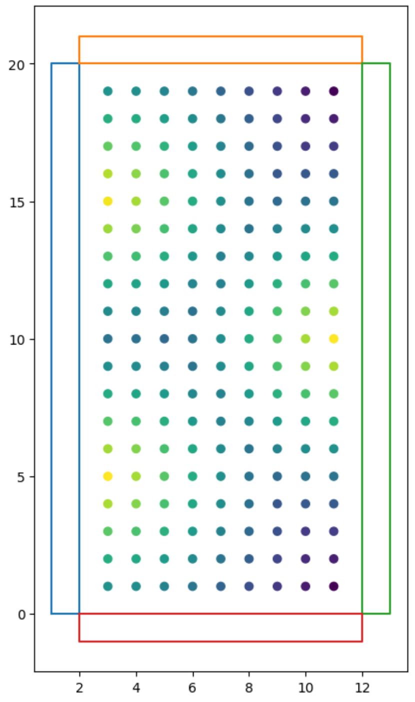
Fig. 7. Intensity of Signal in grid of size 10x20 without objects

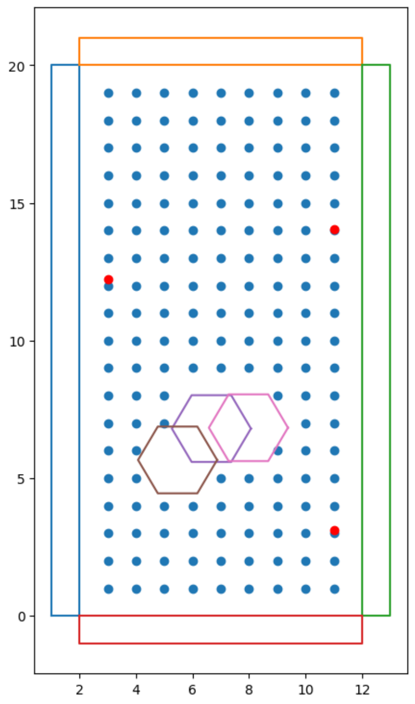
Fig. 8. Positions of sensors in grid of size 10x20 with objects

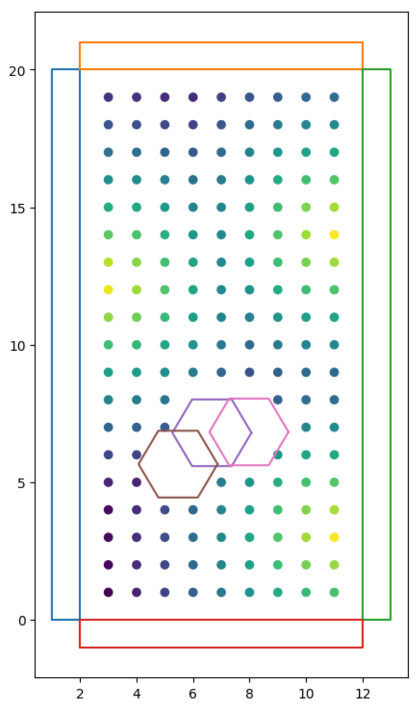
Fig. 9. Intensity of Signal in grid of size 10x20 with objects

showing number of positions and 11 showing intensity.

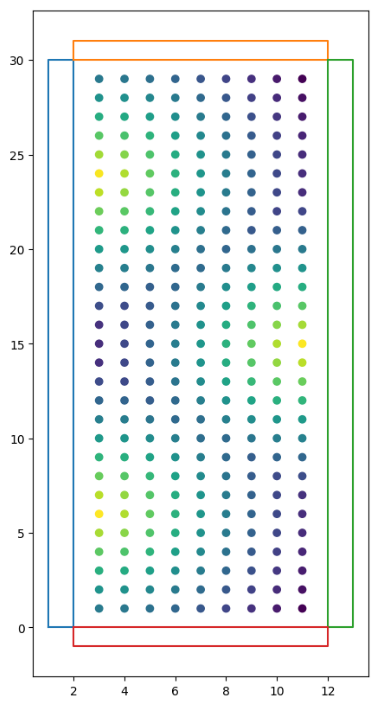
Fig. 11. Intensity of Signal in grid of size 10x30 without objects

Figure 12 and 13 show the results when objects were added. Figure 12 showing number of positions and 13 showing intensity. A total of 4 objects were added while evolving.

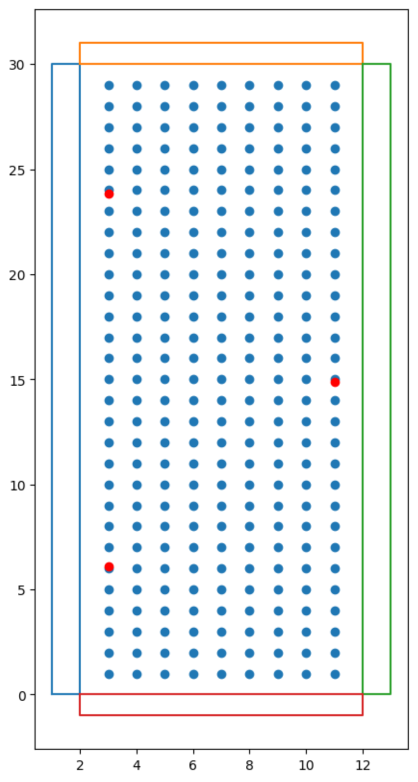
Fig. 10. Positions of sensors in grid of size 10x30 without objects

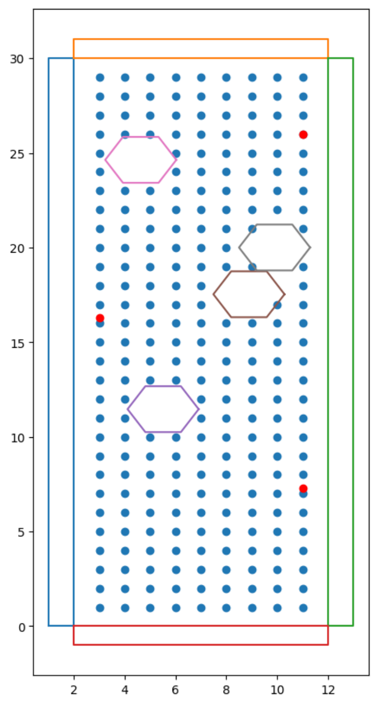
Fig. 12. Positions of sensors in grid of size 10x30 with objects

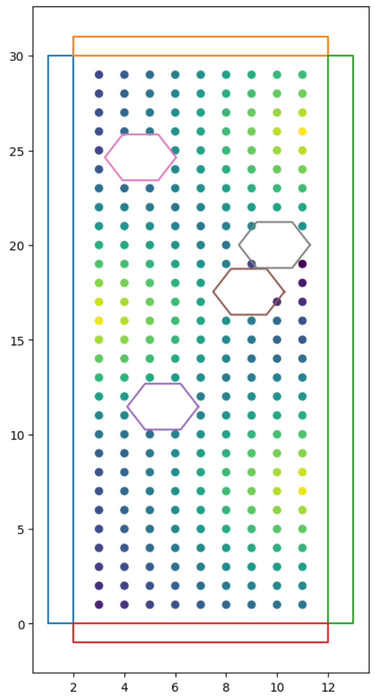

Fig. 13. Intensity of Signal in grid of size 10x30 with objects

The last setup was of Grid of size 5x20. Figure 14 and 15 show the results when no objects were added. Figure 14 showing number of positions and 15 showing intensity.

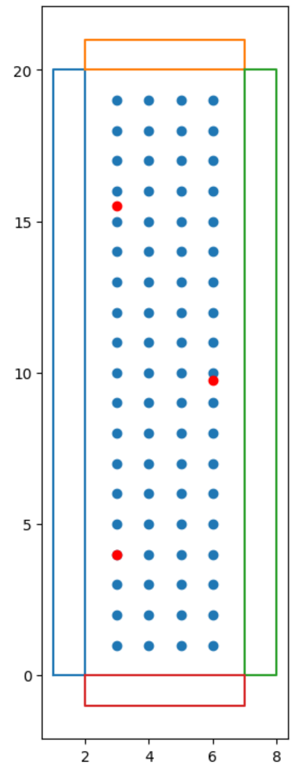

Fig. 14. Positions of sensors in grid of size 5x20 without objects

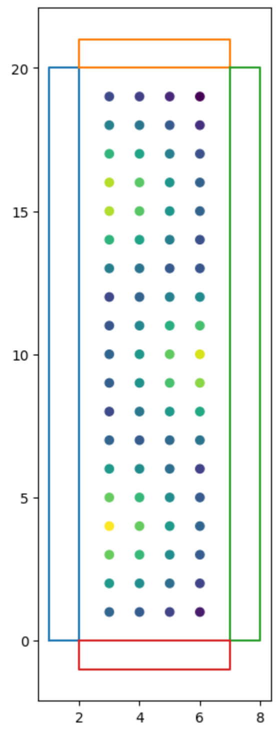

Fig. 15. Intensity of Signal in grid of size 5x20 without objects

Figure 16 shows the positions of sensors when objects were added to this setup and Figure 16 shows intensity.

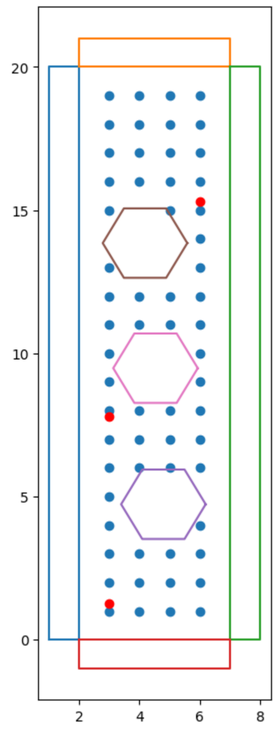

Fig. 16. Positions of sensors in grid of size 5x20 with objects

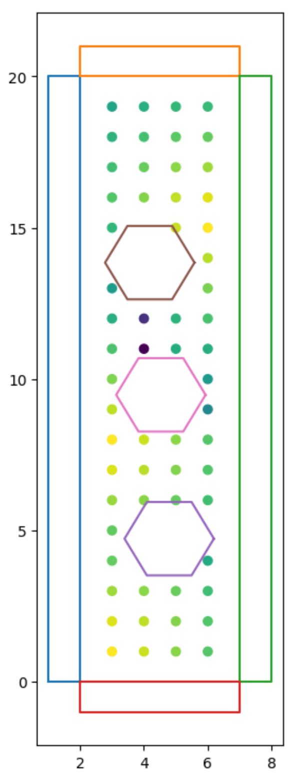

Fig. 17. Intensity of Signal in grid of size 5x20 with objects

VI. Conclusion

In conclusion, we presented a genetic algorithm-based approach for optimizing sensor placement in a room for environmental monitoring. The algorithm was tested on multiple simulated room configurations with varying sensor and obstacle arrangements. The results showed that our approach can efficiently and effectively determine an optimal sensor placement that maximizes the coverage of the monitored area. Additionally, future work can include optimizing the number of sensors used for a given room configuration. Our approach has the potential to be applied to real-world scenarios, such as air quality monitoring in buildings, where optimal sensor placement is crucial for effective monitoring and control.

VII. Future works

Future works in this area could focus on optimizing not only the placement of sensors but also the number of sensors used. The optimal number of sensors for a given room size and configuration could be determined using a similar genetic algorithm-based approach. This would provide a more efficient solution by minimizing the number of sensors needed while still achieving the desired coverage of the monitored area. The inclusion of this additional optimization parameter could further improve the efficiency and effectiveness of the system.

VIII. References

[1] S. Hassani and U. Dackermann, “A systematic review of optimization algorithms for Structural Health Monitoring and optimal sensor placement,” Sensors, vol. 23, no. 6, p. 3293, 2023. doi:10.3390/s23063293 [2] J. H. Holland, “Genetic Algorithms,” Iowa State University, https://www2.econ.iastate.edu/tesfatsi/holland.GAIntro.htm (accessed May 10, 2023). [3] Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Reading: Addison-Wesley [4] W. Ostachowicz, R. Soman, and P. Malinowski, “Optimization of sensor placement for Structural Health Monitoring: A Review,” Structural Health Monitoring, vol. 18, no. 3, pp. 963–988, 2019. doi:10.1177/1475921719825601 [5] S. Spanache, T. Escobat, and L. Travé-Massuyès, ”Sensor Placement Optimisation Using Genetic Algorithms”, (accessed May 10, 2023). [6] A. Khan, S. S. Murtaza, and S. S. Raza, “Optimal illumination of arbitrary rooms using genetic algorithm,” 2019 8th International Conference on Information and Communication Technologies (ICICT), 2019. doi:10.1109/icict47744.2019.9001992