



e-ISSN: 2319-8753 | p-ISSN: 2347-6710

IJRSET

International Journal of Innovative Research in
SCIENCE | ENGINEERING | TECHNOLOGY



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 3, March 2024

ISSN INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA

Impact Factor: 8.423

9940 572 462

6381 907 438

ijrset@gmail.com

www.ijrset.com

Sensor Network Optimization to Maximize Coverage using Modern Algorithm

Dr. Ch. Venkata Suresh, Nikhil V, Nadeemulla Sk, Chetan A

Department of Computer Science and Engineering (IOT), Vasireddy Venkatadri Institute of Technology,
Nambur, Andhra Pradesh, India

C.S Student, Department of Computer Science and Engineering (IOT), Vasireddy Venkatadri Institute of Technology,
Nambur, Andhra Pradesh, India

C.S Student, Department of Computer Science and Engineering (IOT), Vasireddy Venkatadri Institute of Technology,
Nambur, Andhra Pradesh, India

C.S Student, Department of Computer Science and Engineering (IOT), Vasireddy Venkatadri Institute of Technology,
Nambur, Andhra Pradesh, India

ABSTRACT: Wireless sensor networks, or WSNs, are ubiquitous systems made up of dispersed sensor nodes that wirelessly gather and transfer data from the surrounding environment. These networks are becoming essential parts of many applications, such as surveillance, smart cities, healthcare, and environmental monitoring. The proper placement of sensor nodes to provide efficient data gathering and optimal coverage is critical to the functional operation of wireless sensor networks (WSNs). For WSNs to maximize coverage area and minimize resource consumption, including electricity and bandwidth, optimal sensor placement is essential. The spatial range across which sensor nodes can identify and track environmental occurrences is referred to as a WSN's coverage area. In order to collect pertinent data and deliver timely insights into the monitored region, it is imperative to attain complete coverage. Due to its ability to adjust the detecting range of individual sensor nodes, sensor radius tuning is a crucial component in WSN optimization. Network performance may be fine-tuned according to application needs by adjusting sensor radii, which allows for customization of coverage patterns. Attaining an effective and dependable sensor deployment requires optimizing sensor radii by balancing coverage area, energy consumption, and network connectivity. Swarm optimization (PSO) and other contemporary optimization approaches have become effective tools for handling challenging optimization issues in wireless sensor networks (WSNs) in recent years. Particle Swarm Optimization (PSO) methods utilize the laws of social behavior, in which particles, or possible solutions, move about in the search area repeatedly until they find the best answer.

KEYWORDS: Wireless Sensor Networks, meta-heuristic, Voronoi , prohibitively, coverage optimization, Tuning Radius ,Data redundancy ,maximizing resource consumption, PSO algorithm, deployment region

I. INTRODUCTION

Wireless Sensor Networks (WSNs) are a revolutionary technology that are changing a lot of industries, such as transportation, infrastructure management, health-care, and environmental monitoring. Small, inexpensive sensor nodes with sensing, processing, and communication capabilities are the foundation of WSNs. Real-time monitoring, analysis, and decision-making are made possible by these nodes working together to monitor and gather data from the physical environment.

The development of wireless communication technologies, downsizing of sensors, and microelectronics has proliferated WSNs. These advancements have made it possible to install large-scale sensor networks at a reasonable cost, enabling widespread data collecting and monitoring in a variety of settings.

WSNs are primarily designed to collect environmental data and send it to a sink node or central processing unit for further processing, analysis, and interpretation. Depending on the application domain, this information may include motion, sound levels, temperature, humidity, pressure, light intensity, and other environmental characteristics. Flexibility, scalability, and ease of deployment are just a few benefits that WSNs have over conventional wired sensor

systems. Without the requirement for physical connections, WSNs may be quickly implemented in remote or inaccessible regions, in contrast to wired systems, which need for a sophisticated architecture and significant cabling. For applications like environmental monitoring, where building wired infrastructure may be unfeasible or prohibitively expensive, this makes them especially well-suited.

At [2], Optimizing the positioning and setup of sensor nodes to meet specific goals like increasing coverage, lowering energy consumption, and guaranteeing network connectivity is one of the main problems in WSNs. An important factor in defining the network's efficacy and performance is the positioning of the sensor nodes strategically. It is feasible to increase coverage area, boost data accuracy, extend network lifetime, and maximize resource consumption by carefully planning the deployment strategy and placing the sensors.

The use of Wireless Sensor Networks to monitor and engage with our surroundings has completely changed. These networks are made up of many tiny, self-sufficient sensor nodes that are placed in a specific location and tasked with gathering and sending information about the world around them. Ensuring proper coverage, or the degree to which the monitored region is adequately monitored by the sensor nodes, is one of the key issues in WSNs. For WSNs to be as useful and effective as possible in a variety of applications At [10], optimum coverage must be achieved.

WSN sensor coverage optimization is essential for a number of reasons. First off, all-encompassing coverage guarantees that the monitored area is sufficiently watched and tracked throughout. This is especially crucial for applications like environmental monitoring, where incomplete or absent data might result in erroneous judgment calls and judgment calls. We can reduce blind spots and monitoring gaps by maximizing sensor coverage, which will increase the accuracy and dependability of the data that is gathered.

To maximize the effectiveness and efficiency of WSNs, sensor coverage optimization is secondly important. Sensor nodes with low power and communication capabilities are often used in resource-constrained contexts. We can save energy and bandwidth resources while still meeting the required monitoring goals by optimizing sensor coverage. Sensor nodes can be positioned strategically and their sensing ranges adjusted to avoid redundant data collecting, lower energy usage, and increase network longevity.

Furthermore, we can modify WSNs to accommodate changing and dynamic situations thanks to sensor coverage optimization. The efficiency of sensor deployments can be impacted by environmental factors such as barriers, weather patterns, and changes in topography. We may adjust the network design in real-time to account for these changes and guarantee consistent and dependable data collection by continuously monitoring and adjusting sensor coverage.

To achieve effective and efficient data collection from the environment, sensor node location and configuration optimization are critical. The creation of sensor coordinates, adjusting each node's sensing radius, and using sophisticated optimization techniques like Particle Swarm Optimization (PSO) to identify the best solution are essential steps in this optimization process. This introduction explores the significance of each element and how they work together to improve WSN performance.

Co-Ordinates Generation:

The exact position of every sensor node inside the deployment region must be ascertained in order to provide sensor coordinates. The coverage area, connection, and general performance of the WSN are all directly impacted by this task, which is crucial. At [3], Properly positioned and precise sensor nodes eliminate redundancy, maximize resource efficiency, and guarantee thorough coverage. At [9], Based on the particular needs of the application and the limitations of the environment, a variety of techniques, including as random placement, grid-based placement, and deployment algorithms, are used to calculate sensor coordinates.

Tuning Radius:

The following step is to adjust each node's sensing radius after sensor locations have been found. The region that a sensor node can identify events or phenomena in the environment is known as the sensing radius[3]. In order to maximize network lifetime, minimize energy consumption, and achieve the target coverage area, the sensing radius must be optimized. Coverage, energy efficiency, and network connection may all be balanced by varying the sensing radius according to node density, communication range, and application requirements.

PSO Enhancement:

At [4], PSO and other meta-heuristic optimization algorithms have been more well-known in recent years due to their

effectiveness in handling challenging optimization situations. PSO is modeled after the social behaviors of fish schools and bird flocks, in which particles, or possible solutions, move around in the search space until they locate the best answer. PSO may be used in the context of WSN optimization to determine where sensor nodes should be placed and how best to adjust sensing radii in order to optimize coverage, reduce energy consumption, or accomplish other particular goals. PSO is capable of effectively exploring the solution space and converging to close to optimum solutions by iteratively updating particle placements based on their individual experience (personal best) and the combined experience of the swarm (global best).

Connectivity and Significance:

WSN optimization may be approached holistically by combining coordinate generation, radius adjustment, and PSO optimization. Sensor networks that provide extensive coverage, effective resource use, and optimal performance for various applications may be designed and implemented by integrating these components. The importance of each component and how they work together to achieve effective WSN installations are examined in this article, which will eventually improve data gathering, analysis, and decision-making across a range of industries.

II. RELATED WORK

In Wireless Sensor Networks (WSNs), heuristic techniques such as grid-based placement, random placement, and Voronoi diagrams provide simple and efficient ways to arrange sensor nodes. Although there are no assurances about optimality, these techniques are highly valued due to their ease of use and effectiveness in real-world scenarios. By dividing the deployment area into areas according to how close sensor nodes are, At [11], [12] Voronoi diagrams maximize coverage and reduce overlaps in coverage. Sensor nodes are arranged in an organized grid pattern using grid-based placement, which makes network administration easier and allows for consistent coverage. A rapid and simple deployment method is provided by random placement, which disperses sensor nodes throughout the deployment region at random. Even though these heuristic techniques might not always produce the best results, they are useful because they frequently match coverage needs and resource limits in real-world circumstances.

$$V(\pi) = \{q \in \mathbb{R}^n \mid \|q - \pi_i\| \leq \|q - \pi_j\|, \forall j \neq i\}$$

At [13], Convex optimization techniques are essential for defining and addressing issues related to sensor placement and coverage optimization. Through the representation of the optimization issue as a convex program, these methods provide a strong foundation for quickly identifying optimum solutions. Convex optimization algorithms are particularly made to handle convex optimization problems; they offer strong theoretical guarantees and convergence to global optima. Sensor network designers may efficiently optimize sensor locations to increase coverage, reduce energy consumption, and guarantee network connectivity by utilizing convex optimization. This improves the overall performance and dependability of WSN installations.

Genetic algorithms (GAs) At [1], [6], [7] mimic the workings of natural selection, providing a potent method for addressing optimization issues. GAs are a powerful tool for repeatedly developing optimum sensor locations in sensor network optimization. GAs iteratively search the solution space to produce new and improved solutions by expressing prospective solutions as chromosomes and using genetic operators including selection, crossover, and mutation. GAs are an important tool in the design and optimization of Wireless Sensor Networks (WSNs) because of their iterative approach, which makes it possible for them to quickly find the best sensor combinations that optimize coverage, reduce resource consumption, and improve network performance.

K-Clustering At [14], sometimes referred to as K-means clustering, is a flexible technique for sensor network optimization that groups sensor nodes into clusters based on their physical closeness. The network can perform a variety of optimization tasks thanks to this segmentation, including as load balancing, data aggregation, and route optimization. K-means clustering minimizes energy consumption, improves network efficiency, and accelerates data processing and transmission by organizing sensor nodes into clusters. K-means clustering is a fundamental approach in sensor network optimization that greatly enhances overall network performance and reliability by facilitating the smooth operation and resource management of wireless sensor networks.

L-

One notable protocol in sensor network optimization is At [15], LEACH (Low Energy Adaptive Clustering Hierarchy), which is well-known for its capacity to increase network lifetime and efficiently use energy. LEACH dynamically controls energy allocation in Wireless Sensor Networks (WSNs) by arranging sensor nodes into clusters



and switching cluster-head responsibilities on a regular basis. This approach not only increases energy efficiency but also extends the lifespan of the network, which is important for applications that need distant sensing or long-term monitoring. LEACH's extensive implementation is a key component in WSN optimization, meeting the urgent demand for energy reduction while preserving network dependability and performance. The Particle Swarm Optimization (PSO)[2] technique may be used to overcome a number of drawbacks that might occur when utilizing classic optimization algorithms for sensor network optimization. The first problem with premature convergence is that it can cause classical algorithms like At [16] , simulated annealing (SA) and genetic algorithms (GA) to converge to poor solutions before fully exploring the solution space.

III. PROPOSED METHODOLOGY

The optimal placement of sensor nodes is crucial for maximizing network performance and efficiency in Wireless Sensor Networks (WSNs). A tool that generates random coordinates within user-specified dimensions has been created to address this crucial issue and make it easier to optimize node placements in sensor networks. This application offers a flexible and configurable method to achieve optimal coverage and resource usage, marking a significant improvement in WSN deployment tactics.

At [8] , Placing sensor nodes optimally is critical to optimizing coverage, reducing energy consumption, and guaranteeing stable network connectivity. We can efficiently monitor and gather environmental data by placing sensor nodes strategically across the deployment area. This allows for timely insights and well-informed decision-making in a variety of fields. Ability to define the deployment area's size, enabling flexibility in response to a range of environmental factors and application needs. The tool allows users to explore different node placement configurations and find optimum places that meet coverage targets by producing random coordinates within the given length and width.

In order to optimize resource use and improve network performance, sensor radius adjustment is essential for maximizing coverage and decreasing overlapping within a sensor network. Sensor radius tuning provides a more even and balanced distribution of coverage throughout the deployment area by modifying each sensor's radius according to its location and coverage needs. Sensors may effectively expand their reach to cover regions that would otherwise be underserved by using sensor radius tuning, which allows sensors to adjust their coverage radius based on their location. This optimization technique improves the network's capacity to precisely gather pertinent data, reduces blind spots, and achieves complete coverage. In order to reduce sensor overlaps inside the network, sensor radius tweaking is essential.

Table 1 : Difference between using same radius and different range of radii

Feature	Same Radius for All Sensors	Radius Tuning According to Sensor Placement
Optimization	Limited optimization potential as the same radius is applied to all sensors, ignoring variations in coverage requirements.	Enhanced optimization potential as the radius is tuned based on sensor placement, maximizing coverage and minimizing redundancy.
Area Coverage	May result in uneven coverage, with some areas having excessive overlap and others under covered.	Provides more uniform coverage across the deployment area by adjusting the radius based on the distribution of sensors.
Overlap	Higher likelihood of sensor overlap due to uniform radius, leading to redundant data collection and wasted resources.	Reduced overlap between sensors as radius tuning ensures optimal placement, minimizing redundancy and improving resource utilization.



Random Generation	Random coordinate generation remains the same for all sensors, resulting in static placement patterns.	Random coordinate generation may vary for each sensor, resulting in dynamic placement patterns tailored to individual coverage requirements.
Optimization Effort	Requires less effort in terms of parameter tuning and optimization strategy due to uniform radius application.	Requires additional effort in parameter tuning and optimization strategy to ensure effective radius tuning based on sensor placement.
Coverage Flexibility	Less flexible in adapting to diverse coverage requirements or varying deployment scenarios.	More flexible in adapting to diverse coverage requirements or varying deployment scenarios, leading to customizable coverage patterns.
Computational Complexity	Lower computational complexity due to uniform radius application.	Higher computational complexity due to the need for radius tuning based on sensor placement.

With Uniform Radius :

At [5] , Optimizing sensor placement is difficult due to the uniform radius limitation, particularly in locations where a fine fit is required to cover limited residual spaces. When this happens, employing high-range sensors to cover these regions might lead to wasteful spending and resource allocation since the extra coverage goes beyond the necessary bounds. On the other hand, if low-range sensors are used, there can be coverage gaps and some areas won't be tracked. The inflexibility of sensor range distribution hinders the effective use of resources and might result in less than ideal coverage results.

Case 1 :

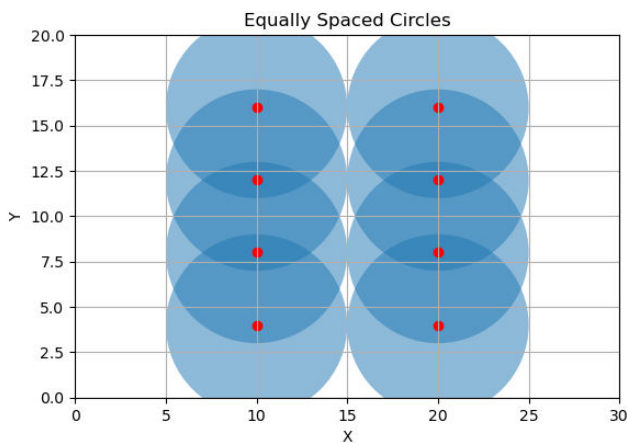


Fig 1: Example coverage of sensors with equal radius

Radius :10m

Coordinates of sensors:

Sensor 1: (10.0, 4.0)

Sensor 2: (10.0, 8.0)

Sensor 3: (10.0, 12.0)

Sensor 4: (10.0, 16.0)

Sensor 5: (20.0, 4.0)

Sensor 6: (20.0, 8.0)

Sensor 7: (20.0, 12.0)

Sensor 8: (20.0, 16.0)

Total area: 600m²

Total area covered by circles:

628.3185307179587m²

Total area uncovered by circles:

-28.31853071795865m²

Sensors with the same radius produce overlapping coverage, which leads to duplicate data gathering and wasteful use of resources. Data redundancy arises from several sensors monitoring the same region, wasting bandwidth and using more energy. Furthermore, the quality and dependability of the monitoring system may be impacted by the duplicated data, which might add noise and inconsistencies into the gathered data.



Case 2 :

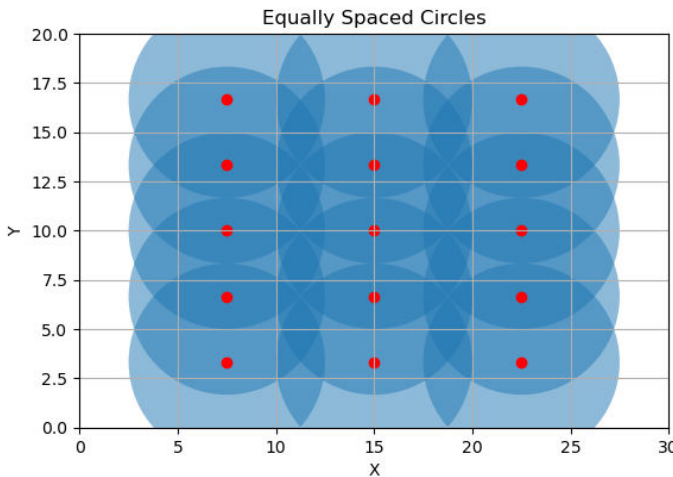


Fig 2: Example coverage of sensors with equal radius

Total area: $600m^2$
 Total area covered by circles:
 $1178.0972450961724m^2$
 Total area uncovered by circles: -
 $578.0972450961724m^2$

Radius :10m

No.of Sensors:15

Coordinates of sensors:

Sensor 1: (7.5, 3.3333333333333335)

Sensor 2: (7.5, 6.666666666666667)

Sensor 3: (7.5, 10.0)

Sensor 4: (7.5, 13.333333333333334)

Sensor 5: (7.5, 16.666666666666668)

Sensor 6: (15.0, 3.3333333333333335)

Sensor 7: (15.0, 6.666666666666667)

Sensor 8: (15.0, 10.0)

Sensor 9: (15.0, 13.333333333333334)

Sensor 10: (15.0,

16.666666666666668)

Sensor 11: (22.5,

3.3333333333333335)

Sensor 12: (22.5, 6.666666666666667)

Sensor 13: (22.5, 10.0)

Sensor 14: (22.5,

13.333333333333334)

Sensor 15: (22.5,

16.666666666666668)

Random Coordinates and Radius Generation :

Furthermore, the application constantly modifies the number of sensors produced in accordance with the user-specified estimated area, with smaller areas producing fewer sensors and bigger areas producing more. By ensuring that the density of sensor nodes is proportionate to the size of the deployment region, this adaptive function maximizes coverage efficiency and resource use. The program offers a customized method to sensor deployment that enhances the efficacy of the sensor network across a variety of locations and application scenarios by automatically adjusting the number of sensors to meet the spatial dimensions supplied by the user.

Users may visually evaluate the density and dispersion of sensor nodes in the deployment area thanks to the graphical representation of node placements. Potential coverage gaps may be found, node placement can be optimized, and the efficacy of the deployed network can be assessed with the help of this visual depiction.

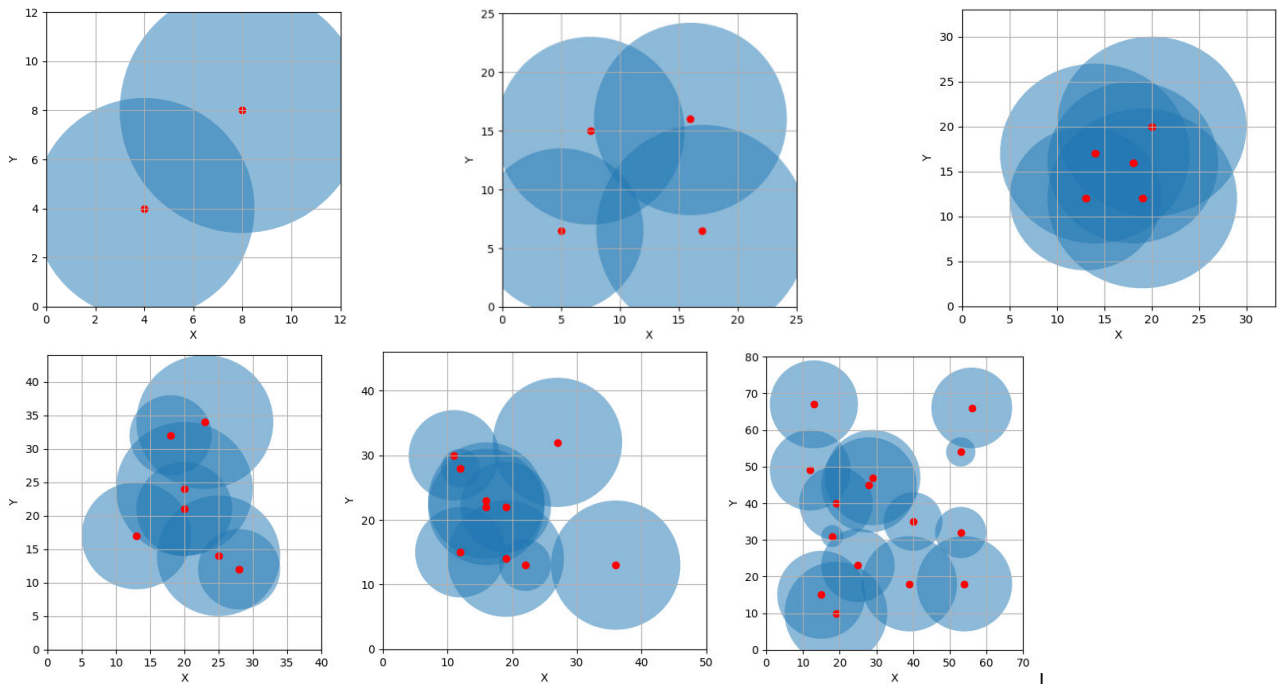


Fig 3: Random Coordinates and radius generated by Manual Software

Using the most recent Particle Swarm Optimization (PSO) technique, the software is further enhanced by automatically optimizing the sensor placements in addition to creating random coordinates and dynamically modifying the number of sensors based on the designated region. Inspired by the social nature of fish schools or flocks of birds, the PSO algorithm is a potent optimization tool that finds the best solution by having particles, or possible solutions, repeatedly move to different locations in the search space.

3.1 Initialization: A swarm of particles, each representing a potential solution in the search space, is initialized by the PSO method. Each particle is associated with a set of coordinates that indicate the locations of sensor nodes inside the deployment region in the context of sensor placement optimization.

pop: The total number of swarm particles.

itermax: The PSO algorithm's maximum number of iterations.

R: The interval or range that is utilized to compute velocities.

Wmax: The greatest weight factor due to inertia.

Wmin: The weight factor of inertia at minimum.

npar: Each particle's total number of parameters. In this instance, it comprises each sensor's radius, x-coordinate, and y-coordinate.

3.2 Objective Function: The PSO method establishes an objective function that assesses the degree of solution quality for every particle. To evaluate the efficacy of the sensor placements in the context of sensor placement optimization, the objective function takes into account variables like coverage area, connection, energy consumption, and other pertinent metrics.

Initialization of cost and fitness

```
cost = np.zeros((itermax, pop))
```

```
fit = np.zeros(pop)
```

```
Pbest = posg.copy()
```

```
localcost = cost.copy()
```

```
localfit = fit.copy()
```

Initialization of global best-known position and cost

```
globalcost = np.inf
```

```
Gbest = Pbest[np.argmin(localcost), :]
```


$$\text{globalfit} = 1 / (1 + \text{globalcost})$$

3.3 Particle Movement: Particles modify their locations in each iteration according to their individual experiences (personal best) and the swarm's collective experiences (global best). Velocity vectors, which specify the speed and direction of each particle's travel inside the search space, serve as the guides for this motion.
 $\text{velupdate} = W * \text{vel} + (c1 * r1 * (\text{Pbest} - \text{pos})) + (c2 * r2 * (\text{Gbest} - \text{pos}))$

3.4 Convergence: Until a termination condition is satisfied, such as reaching a maximum number of iterations or obtaining a suitable solution quality, the PSO algorithm iteratively updates particle locations and velocities. Particles converge on the best answer throughout the optimization process, fine-tuning sensor node locations to enhance network efficiency.

$$\text{cost}[\text{iter}-1, \text{pp}] = (1 * b) - (\text{np.pi} * x1 * x1 + \text{np.pi} * x2 * x2)$$

$$\text{fit}[\text{pp}] = 1 / (1 + \text{cost}[\text{iter}-1, \text{pp}])$$

3.5 Optimal Sensor Positions: The PSO algorithm determines the best locations for sensor nodes inside the deployment area at the conclusion of the optimization phase. The placement of these optimum sites results in a more reliable and efficient sensor network deployment by maximizing coverage, reducing energy consumption, and improving network efficiency.

$$\text{Gbest} = \text{Pbest}[\text{pp}, :]$$

$$\text{globalcost} = \text{localcost}[\text{pp}]$$

$$\text{globalfit} = \text{localfit}[\text{pp}]$$

Users may automatically optimize sensor placements without the need for manual intervention by adding the PSO algorithm into the application. This ensures that the sensor network reaches its maximum potential in terms of coverage, connection, and resource use. The program's functionality is improved by this sophisticated optimization method, which also lets users install dependable and highly effective sensor networks in a range of settings and applications.

IV. RESULTS AND ANALYSIS

The Particle Swarm Optimization (PSO) technique is used to randomly generate sensor coordinates with the goal of determining the best locations for sensors in a given space. In this approach, the positions of particles, which stand in for possible sensor sites, are updated iteratively depending on both individual particle experiences and the collective experiences of the swarm.

Initially, inside the designated boundaries of the region where sensors are to be installed, a population of particles is randomly initialized. The coordinates (x, y) of each particle indicate a possible sensor position inside the region. Then, using a predetermined objective function, the PSO algorithm iterates across a number of generations, assessing the fitness of each particle's location.

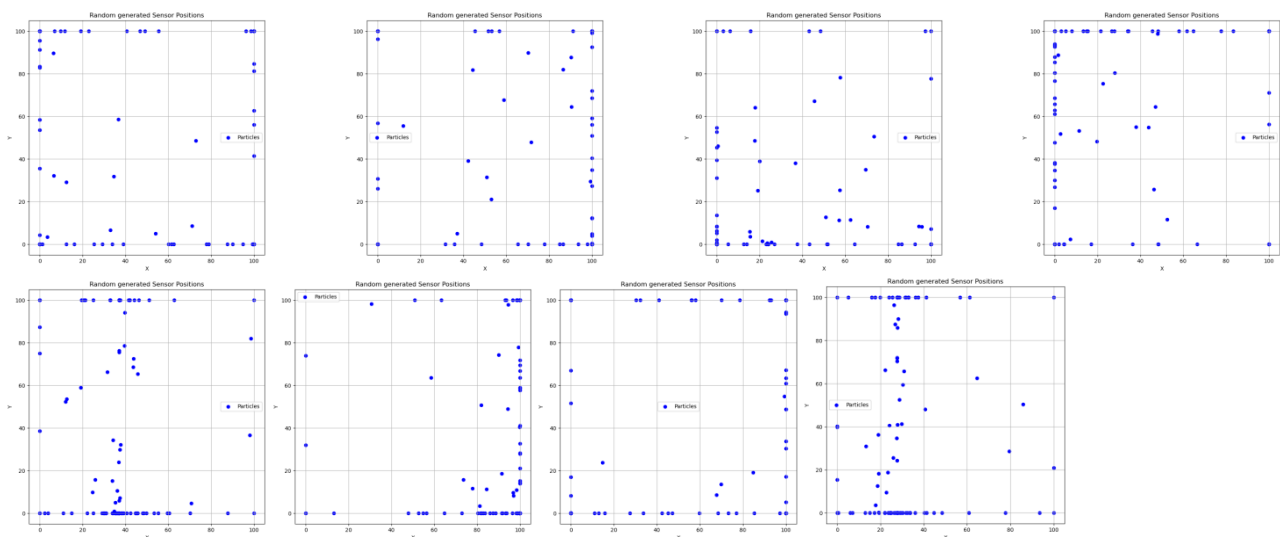


Fig 4 : Random Co-ordinates generation over every iteration by PSO Algorithm

A particle's ability to attain the targeted coverage or performance measure inside the sensor network is usually what determines its fitness. In this instance, the coordinates of the sensors and their corresponding coverage regions are used to assess the coverage area or efficacy of the sensor network.

Based on each particle's best-known location (personal best) and the best-known position of the whole swarm (global best), the PSO algorithm adjusts each particle's velocity and position throughout each iteration. The inertia weight factor, cognitive parameter, and social parameter—all of which control the swarm's capacity for exploration and exploitation—have an impact on this updating process.

Particles gravitate toward better solutions when the PSO algorithm runs through iterations, gradually adjusting the sensor placements to optimize coverage or accomplish other optimization goals. The algorithm eventually finds a solution where the sensor coordinates produce the best performance metric or the largest coverage area.

The particles iteratively update their locations as the optimization goes along, taking into account both their present velocity and the swarm's collective experiences. The particles eventually converge towards optimal places within the sensor network that increase coverage and decrease overlap through velocity tweaking.

In order to direct the particles toward better solutions, their velocities are changed at each cycle. Both the collective knowledge of the swarm (social influence) and the experiences of individual particles (cognition) have an impact on this adjustment process. The PSO algorithm efficiently traverses the search space and converges towards ideal sensor placements for increased coverage by iteratively updating velocities and locations.

The PSO algorithm's combination of random coordinate generation and velocity adjustment [3] allows the sensor network to effectively traverse the optimization landscape, leading to improved performance and optimal coverage in the end.

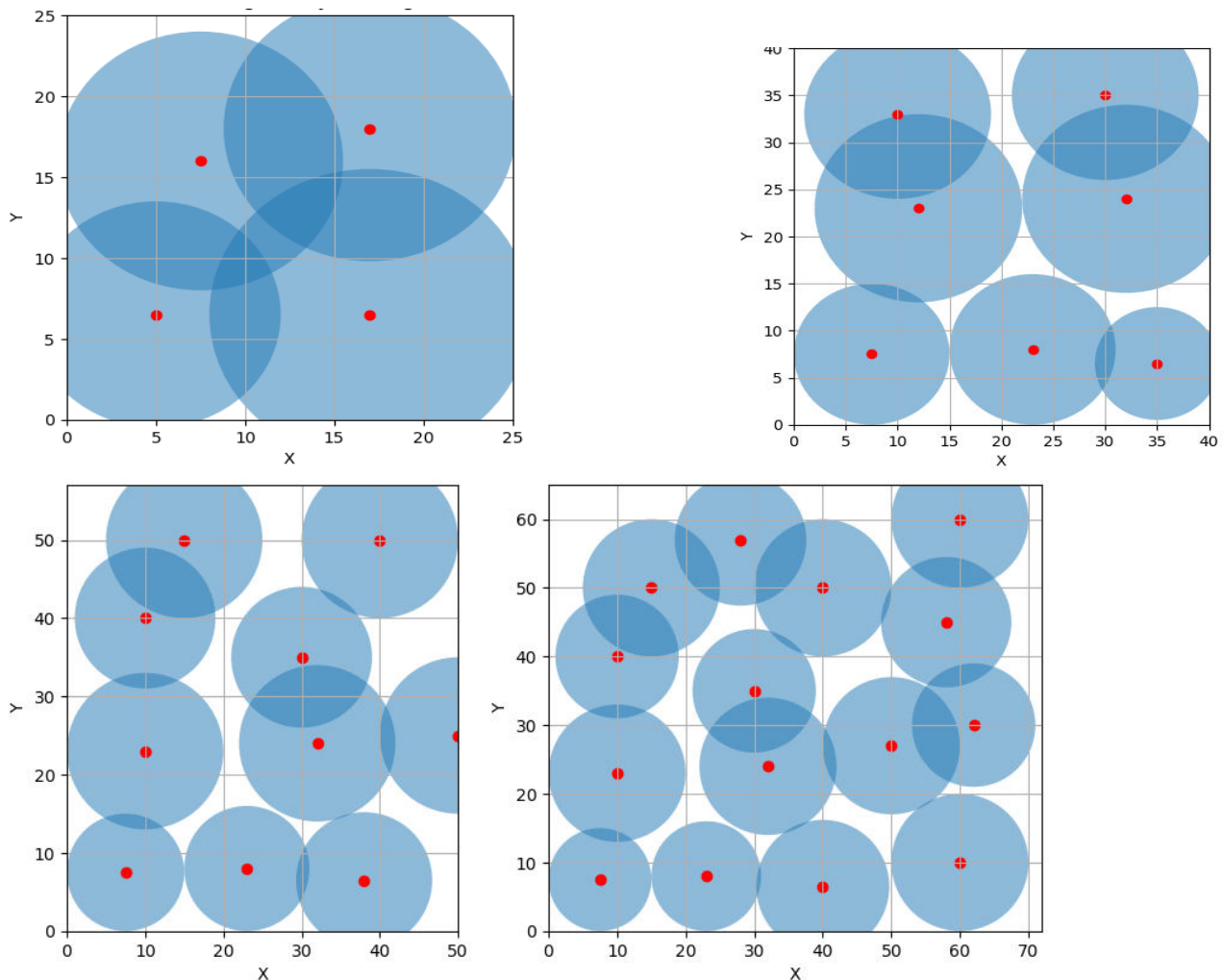


Fig 5: Generation and plotting of random Co-ordinates and radius by PSO Algorithm

The iterative optimization process of the Particle Swarm Optimization (PSO) algorithm is illustrated by the graph showing the consistent direct proportional rise of the optimal location across iterations. The number of iterations is usually represented by the x-axis in this graph, while the quality or fitness of the algorithm's best-known solution is represented by the y-axis.

Case 1 :

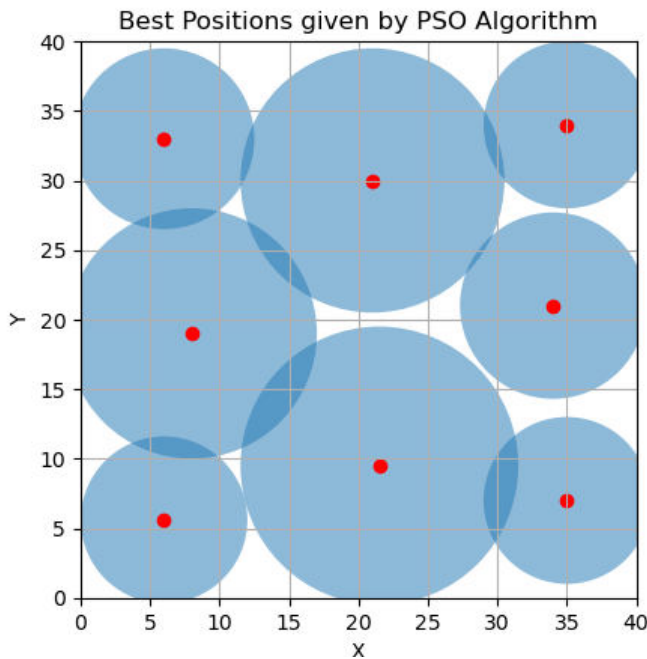


Fig 6: PSO generated Optimized positions and radius

No of Sensors :8
 Coordinates of sensors:
 Sensor 1: (6, 5.6) ,6m
 Sensor 2: (21.5, 9.5) ,10m
 Sensor 3: (35, 7) ,6m
 Sensor 4: (21, 30) ,9.5m
 Sensor 5: (6, 33) ,6.5m
 Sensor 6: (35, 34) ,6m
 Sensor 7: (8, 19) ,9m
 Sensor 8: (34, 21) ,6.7m
 Total area of the region: 1600 m²
 Total area covered by circles: 1465.2073977077434 m²
 Total area uncovered by circles: 134.79260229225656 m²

Case 2 :

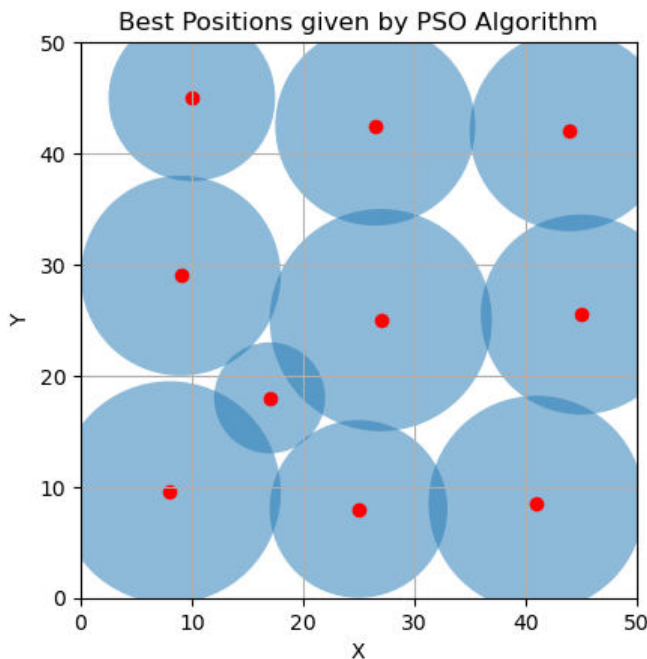


Fig 7: PSO generated Optimized positions and radius

No of Sensors :10
 Coordinates of sensors:
 Sensor 1: (8, 9.5) ,10m
 Sensor 2: (25, 8) ,8m
 Sensor 3: (41, 8.5) ,9.7m
 Sensor 4: (27, 25) ,10m
 Sensor 5: (17, 18) ,5m
 Sensor 6: (26.5, 42.5) ,9m
 Sensor 7: (9, 29) ,9m
 Sensor 8: (10, 45) ,7.5m
 Sensor 9: (44, 42) ,9m
 Sensor 10: (45, 25.5) ,9m
 Total area of the region: 2500 m²
 Total area covered by circles: 2398.1033361912328 m²
 Total area uncovered by circles: 101.89666380876724 m²

Case 3 :



Fig 8: PSO generated Optimized positions and radius

No.of Sensors:15

Coordinates of sensors:

Sensor 1: (7.5, 7) ,7.5m

Sensor 2: (23, 8) ,8m

Sensor 3: (40, 6.5) ,9.7m

Sensor 4: (30, 25) ,10 m

Sensor 5: (10, 23) ,10m

Sensor 6: (28, 40) ,9m

Sensor 7: (10, 40) ,9m

Sensor 8: (10, 55) ,10m

Sensor 9: (47, 55) ,9m

Sensor 10: (48, 24) ,9m

Sensor 11: (60, 10) ,10m

Sensor 12: (45, 38) ,9.5m

Sensor 13: (63, 30) ,10m

Sensor 14: (63, 50) ,10m

Sensor 15: (28, 57) ,9.6m

Total area of the region: 4680 m²

Total area covered by circles: 4403.7654 m²

Total area uncovered by circles: 276.2346 m²

Population Vs Iterations :

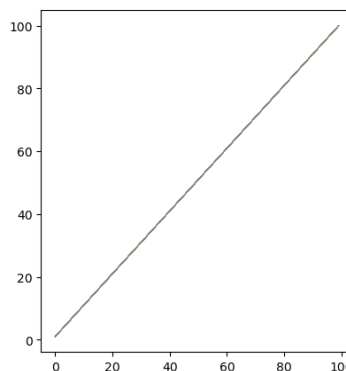


Fig 9: Graph between Population and iteration in PSO

The cost function progressively approaches a constant value as the number of iterations rises. When the particles search the search space and try to find better solutions, the cost first varies greatly in the early rounds of the PSO algorithm. The cost values fluctuate a lot throughout this fluctuation, which suggests that the particles are still searching for favorable areas of the solution space.

As the PSO algorithm iterates, the observed behavior in the cost function graph shows how the optimization process becomes better and better. Higher coverage areas or improved performance metrics for the sensor network are the results of the algorithm's successful optimization of the sensor placements, which is shown by a progressive decline in the cost function. Insufficient exploration or convergence problems, on the other hand, may be indicated by inconsistent or variable cost values, emphasizing the significance of sufficient iterations for reaching optimal solutions.

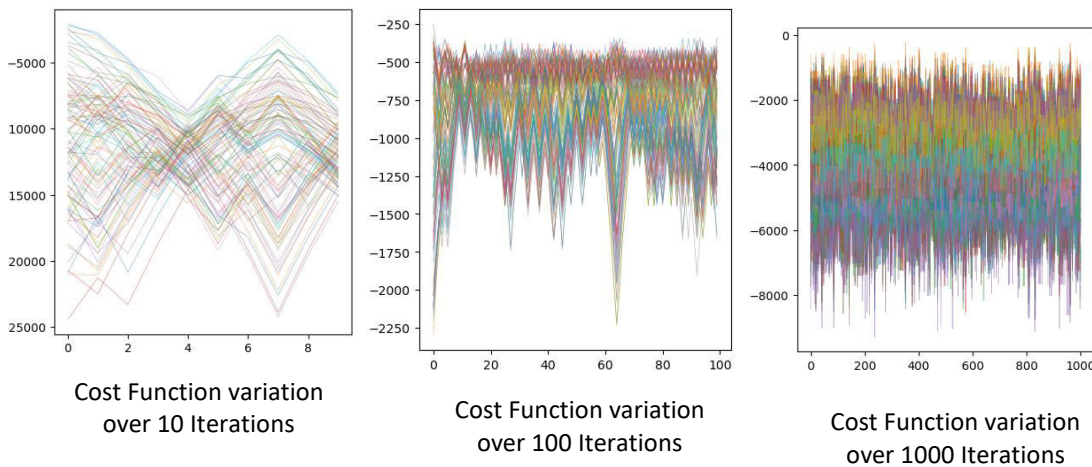


Fig 10: Cost Variation in PSO Algorithm over multiple Iterations

V. CONCLUSION AND FUTURE SCOPE

In conclusion, there is much promise for increasing the network's coverage through the use of the Particle Swarm Optimization (PSO) method in conjunction with random coordinate generation and sensor tuning for sensor network optimization. The PSO algorithm effectively searches the solution space and finds ideal configurations that improve the efficacy and performance of the sensor network through the iterative refining of sensor placements and radius adjustment.

In the deployment region, a variety of possible sensor positions may be explored thanks to the use of random coordinate generation. The PSO algorithm constantly modifies sensor placements to enhance coverage by utilizing the collective wisdom of the swarm. This ensures that vital regions are sufficiently monitored while minimizing redundancy and maximizing resource consumption.

Sensor radius tuning provides fine-grained control over each sensor's coverage area, enabling accurate modifications depending on the application's unique needs. This adaptive tuning makes sure that the sensor network strikes the best possible balance between resource efficiency and coverage, which improves the network's overall efficacy in identifying and tracking environmental occurrences and phenomena.

A reliable and adaptable method for optimizing sensor networks is provided by the combination of the PSO algorithm with random coordinate generation and sensor tweaking. This approach advances a number of applications, such as surveillance, environmental monitoring, and infrastructure management, by optimizing coverage while consuming the fewest resources possible. The efficiency and scalability of the optimization process may be further improved by investigating further improvements and refinements through study and experimentation.

REFERENCES

1. Jin, S., Zhou, M., & Wu, A. S. (Year). Sensor Network Optimization Using a Genetic Algorithm. School of Electrical Engineering and Computer Science, University of Central Florida, Orlando, FL 32816.1
2. Vimalarani, C., Subramanian, R., & Sivanandam, S. N. (2015). An Enhanced PSO-Based Clustering Energy Optimization Algorithm for Wireless Sensor Network. **International Journal of Distributed Sensor Networks**, 11(12), 1-13. doi:10.1155/2015/439532
3. Ling, H., Zhu, T., He, W., Luo, H., Wang, Q., & Jiang, Y. (2020). Coverage Optimization of Sensors under Multiple Constraints Using the Improved PSO Algorithm. **Computational Intelligence and Neuroscience, 2020**. <https://doi.org/10.1155/2020/5416483>
4. Zhao, Q.; Li, C.; Zhu, D.; Xie, C. Coverage Optimization of Wireless Sensor Networks Using Combinations of PSO and Chaos Optimization. *Electronics* 2022, 11, 853. <https://doi.org/10.3390/electronics11060853>. Academic Editor: Jaime Lloret. Received: 27 January 2022; Accepted: 5 March 2022; Published: 9 March 2022.
5. Zhao, W., & Fan, Z. (June 2011). Network Coverage Optimization Strategy in Wireless Sensor Networks Based on Particle Swarm Optimization.
6. IJEAST. (2021). Genetic Algorithm for Wireless Sensor Networks. *International Journal of Engineering and Applied Sciences Technology*, 6(8), 97-103. ISSN No. 2455-2143. Retrieved from <http://www.ijeast.com>

7. Norouzi, A., & Zaim, A. H. (2014). Genetic Algorithm Application in Optimization of Wireless Sensor Networks. *International Journal of Distributed Sensor Networks*, 10(2), 608625. <https://doi.org/10.1155/2014/608625>
8. Zhang, Y., & Liu, M. (2020). Node Placement Optimization of Wireless Sensor Networks Using Multi-Objective Adaptive Degressive Array Number Encoded Genetic Algorithm. *Algorithms*, 13(8), 189. <https://doi.org/10.3390/a13080189>
9. M. Farsi, M. A. Elhosseini, M. Badawy, H. Arafat Ali and H. Zain Eldin, "Deployment Techniques in Wireless Sensor Networks, Coverage and Connectivity: A Survey," in *IEEE Access*, vol. 7, pp. 28940-28954, 2019, doi: 10.1109/ACCESS.2019.2902072.
10. Ahuja, R. K., Magnanti, T. L., Orlin, J. B., & Reddy, M. R. (1995). Applications of Network Optimization. In M. O. Ball et al. (Eds.), *Handbooks in OR & MS*, Vol. 7. Elsevier Science B.V.
11. Tao, R., Liu, J., Song, Y., Peng, R., Zhang, D., & Qiao, J. (2021). Detection and Optimization of Traffic Networks Based on Voronoi Diagram. Retrieved https://www.researchgate.net/publication/358257513_Detection_and_Optimization_of_Traffic_Networks_Based_on_Voronoi_Diagram
12. T. Shu, K. B. Dsouza, V. Bhargava and C. d. Silva, "Using geometric centroid of Voronoi Diagram for coverage and lifetime optimization in mobile wireless sensor networks," *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, Edmonton, AB, Canada, 2019, pp. 1-5, doi: 10.1109/CCECE.2019.8861820.
13. T. Chen, Q. Ling and G. B. Giannakis, "Online convex optimization for dynamic network resource allocation," *2017 25th European Signal Processing Conference (EUSIPCO)*, Kos, Greece, 2017, pp. 136-140, doi: 10.23919/EUSIPCO.2017.8081184.
14. Jin, S., Zhou, M., & Wu, A. S. (2022). K-Means clustering of optimized wireless network sensor using genetic algorithm. *Periodicals of Engineering and Natural Sciences (PEN)*, 10(3), 276-285. DOI: 10.21533/pen.v10i3.3059
15. Hindawi. (August 2022). Optimization of Leach Protocol in Wireless Sensor Network Using Machine Learning. *Computational Intelligence and Neuroscience*, 2022(3), 1-8. DOI: 10.1155/2022/5393251. License: CC BY 4.0.
16. N. A. B. A. Aziz, A. W. Mohemmed and M. Y. Alias, "A wireless sensor network coverage optimization algorithm based on particle swarm optimization and Voronoi diagram," *2009 International Conference on Networking, Sensing and Control*, Okayama, Japan, 2009, pp. 602-607, doi: 10.1109/ICNSC.2009.4919346.



INTERNATIONAL
STANDARD
SERIAL
NUMBER
INDIA



INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

 9940 572 462  6381 907 438  ijirset@gmail.com



www.ijirset.com

Scan to save the contact details