SENSOR NETWORK OPTIMIZATION TO MAXIMIZE COVERAGE USING MODERN ALGORITHM

A project report submitted in partial fulfillment of the requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

in

COMPUTER SCIENCE AND ENGINEERING (Internet of Things)

Submitted by

V. NIKHIL	**(20BQ1A4953)**
SK. NADEEMULLA	**(20BQ1A4946)**
A. CHETAN	**(20BQ1A4902)**

under the esteemed guidance of

Dr. Chintalapudi . V Suresh

(Professor)

[Program: Computer Science and Engineering (Internet of Things) – CSO]

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

Approved by AICTE, Permanently Affiliated to JNTUK, NAAC Accredited with ‘A’ Grade, ISO 9001:2015 Certified

Nambur (V), Pedakakani (M), Guntur (Dt.), Andhra Pradesh – 522 508

APRIL - 2024

[Program: Computer Science and Engineering (Internet of Things) – CSO]

VASIREDDY VENKATADRI INSTITUTE OF TECHNOLOGY

(Autonomous)

Approved by AICTE, Permanently Affiliated to JNTUK, NAAC Accredited with 'A' Grade, ISO 9001:2015 Certified

Nambur (V), Pedakakani (M), Guntur (Dt.), Andhra Pradesh – 522 508

CERTIFICATE

This is to certify that the project report entitled "**Sensor Network Optimization To Maximize Coverage using Modern Algorithm** " is being submitted by **V. Nikhil** (20BQ1A4953), **Sk. Nadeemulla** (20BQ1A4946), **A. Chetan** (20BQ1A4902) in partial fulfillment of the requirement for the award of the degree of the **Bachelor of Technology** in **Computer Science and Engineering (Internet of Things)** to the Vasireddy Venkatadri Institute of Technology is a record of bonafide work carried out by them under my guidance and supervision.

The results embodied in this project have not been submitted to any other university or institute for the award of any degree or diploma.

Signature of the Supervisor
Dr. Chintalapudi V Suresh
Professor,
Department of CSO, VVIT.

Head of the Department
Dr. Chintalapudi V Suresh
Professor & HoD,
Department of CSO, VVIT.

DECLARATION

We hereby declare that the work embodied in this project entitled "**Sensor Network Optimization To Maximize Coverage using Modern Algorithm**", which is being submitted by us in requirement for the B. Tech Degree in **Computer Science and Engineering (Internet of Things)**from Vasireddy Venkatadri Institute of Technology, is the result of investigations carried out by us under the supervision of **Dr. Chintalapudi V Suresh, Professor**. The work is original and the results in this thesis have not been submitted elsewhere for the award of any degree or diploma.

Signature of the Candidates

V.Nikhil (Regd.No**: 20BQ1A4953**)

Sk. Nadeemulla (Regd.No**: 20BQ1A4946**)

A. Chetan (Regd.No**: 20BQ1A4902**)

ACKNOWLEDGEMENT

We take this opportunity to express deepest gratitude and appreciation to all those people who made this project work easier with words of encouragement, motivation, discipline, and faith by offering different places to look to expand ourideas and help us towards the successful completion of this project work.

First and foremost, we express deep gratitude to **Sri Vasireddy Vidya Sagar**, Chairman, Vasireddy Venkatadri Institute of Technology for providing necessary facilities throughout the B.Tech programme.

We express sincere thanks to **Dr. Y. Mallikarjuna Reddy,** Principal, Vasireddy Venkatadri Institute of Technology for his constant support and cooperation throughout the B.Tech programme.

We express sincere gratitude to **Dr. Ch. Venkata Suresh**, Professor & HOD, Computer Science & Engineering (Internet of Things), Vasireddy Venkatadri Institute of Technology for his constant encouragement, motivation and faith by offering different places to look to expand our ideas.

We would like to express sincere gratitude to our Guide **Mr. S. Saida Rao**, Assistant Professor, CSO for his insightful advice, motivating suggestions, invaluable guidance, help and support in successful completion of this project.

We would like to take this opportunity to express thanks to the **Teaching and NonTeaching Staff** in the Department of Computer Science & Engineering (Internet of Things), VVIT for their invaluable help and support.

Names of Candidates

V.Nikhil (20BQ1A4953)

Sk. Nadeemulla (20BQ1A4946)

A. Chetan (20BQ1A4902)

Department Vision

To accomplish the aspirations of emerging engineers to attain global intelligence by obtaining computing and design abilities through communication that elevate them to meet the needs of industry, economy, society, environmental and global.

Department Mission

- ➢ To mold the fresh minds into highly competent IoT application developers by enhancing their knowledge and skills in diverse hardware and software design aspects for covering technologies and multi-disciplinary engineering practices.
- ➢ To provide the sate- of- the art facilities to forge the students in industry-ready in IoT system development.
- ➢ To nurture the sense of creativity and innovation to adopt the socio-economic related activities.
- ➢ To promote collaboration with the institutes of national and international repute with a view to have best careers.
- ➢ To enable graduates to emerge as independent entrepreneurs and future leaders.

Program Educational Objectives (PEOs)

PEO-1: To formulate the engineering practitioners to solve industry's technological problems

PEO-2: To engage the engineering professionals in technology development, deployment and engineering system implementation

PEO-3: To instil professional ethics, values, social awareness and responsibility to emerging technology leaders

PEO-4: To facilitate interaction between students and peers in other disciplines of industry and society that contribute to the economic growth**.**

PEO-5: To provide the technocrats the amicable environment for the successful pursuing of engineering and management.

PEO-6: To create right path to pursue their careers in teaching, research and innovation.

Program Outcomes (POs)

PO1: Engineering knowledge: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to the solution of complex engineering problems.

PO2: Problem analysis: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and engineering sciences.

PO3: Design/development of solutions: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental considerations.

PO4: Conduct investigations of complex problems: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to provide valid conclusions.

PO5: Modern tool usage: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.

PO6: The engineer and society: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional engineering practice.

PO7: Environment and sustainability: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable development.

PO8: Ethics: Apply ethical principles and commit to professional ethics and responsibilities and norms of the engineering practice.

PO9: Individual and team work: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.

PO11: Project management and finance: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO12: Life-long learning: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the broadest context of technological change.

Program Specific Outcomes (PSOs)

PSO-1: Proficient and innovative with a strong cognizance in the arenas of sensors, IoT, data science, controllers and signal processing through the application of acquired knowledge and skills.

PSO-2: Apply cutting-edge techniques and tools of sensing and computation to solve multi-disciplinary challenges in industry and society.

PSO-3: Exhibit independent and collaborative research with strategic planning while demonstrating professional and ethical responsibilities of the engineering profession.

Project Outcomes

Students who complete a minor project will:

PW-01. Use the design principles and develop concept for the project

PW-02. Estimate the time frame and cost for the project execution and completion.

PW-03. Analyze the project progress with remedial measures individual in a team.

PW-04. Examine the environmental impact of the project.

PW-05. Demonstrate the project functionality along with report and presentation.

PW-06. Apply the Engineering knowledge in design and economically manufacturing of components to support the society need.

PW-07. Assess health, safety and legal relevant to professional engineering practices.

PW-08. Comply the environmental needs and sustainable development.

PW-09. Justify ethical principles in engineering practices.

PW-010. Perform multidisciplinary task as an individual and / or team member to manage the project/task.

PW-011. Comprehend the Engineering activities with effective presentation and report.

PW-012. Interpret the findings with appropriate technological / research citation.

MAPPING OF PROJECT OUTCOMES TO POs

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
PW-01	3	2	2	2								
PW-02	3	2	2								3	
PW-03	3	3		2	3					3		
PW-04	3					3	3	3				3
PW-05	3	2									3	
PW-06	3	2	2	2	3							
PW-07						3						
PW-08							3					
PW-09								3				
PW-10									3		3	
PW-11										3		
PW-12												3
PW-PO	**3**	**2**	**2**	**2**	**3**	**3**	**3**	**3**	**3**	**3**	**3**	**3**

MAPPING OF PROJECT OUTCOMES TO PSOs

	PSO1	PSO2	PSO3
PW-01	2	2	2
PW-02			2
PW-03			
PW-04	3	3	3
PW-05			
PW-06	2	2	2
PW-07	2	2	2
PW-08	1	1	1
PW-09	2	2	2
PW-10	2	2	2
PW-11	2	2	2
PW-12	1	1	1
PW-PSO	**2**	**2**	**2**

Note: Strong – 3, Moderate – 2, Low – 1

CONTENTS

ABSTRACT

Wireless sensor networks, or WSNs, are ubiquitous systems made up of dispersed sensor nodes that wireless gather and transfer data from the surrounding environment. These networks are becoming essential parts of many applications, such as surveillance, smart cities, health care, 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.

LIST OF FIGURES

LIST OF TABLES

CHAPTER-1
INTRODUCTION

1.1 INTRODUCTION:

Sensor networks have emerged as critical components in various fields, including environmental monitoring, industrial automation, health-care, and smart cities. These networks consist of interconnected sensors capable of collecting and transmitting data from their surrounding environment. However, optimizing sensor networks to ensure maximum coverage while minimizing resource utilization poses a significant challenge. Traditional optimization methods often fall short in addressing the complexities of modern sensor networks. Therefore, the utilization of modern algorithms becomes imperative to enhance network efficiency and performance. The introduction of modern algorithms such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO) has revolutionized the field of sensor network optimization by providing more robust and adaptive solutions. These algorithms leverage techniques inspired by natural phenomena, such as the behavior of swarms, genetic evolution, and ant foraging, to iteratively improve sensor placement and coverage.

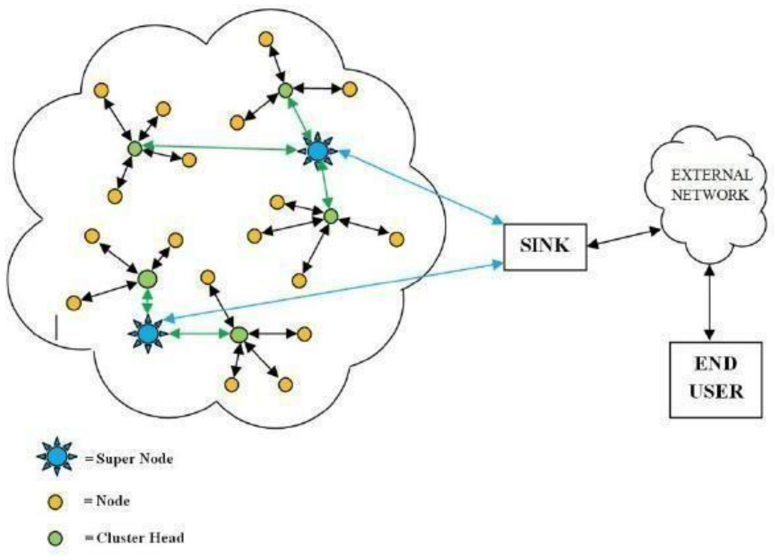


Fig 1: Wireless Sensor Network

BACKGROUND:

The development of sensor networks can be traced back to early applications in military surveillance and industrial automation. Over time, sensor networks have evolved to encompass a wide range of applications, from smart agriculture to disaster management. Traditional optimization methods for sensor networks typically involve heuristic approaches or simple algorithms like greedy algorithms. However, these methods often struggle to handle the dynamic and complex nature of modern sensor networks. In contrast, modern optimization algorithms, such as evolutionary algorithms and swarm intelligence techniques, offer more sophisticated and effective solutions for optimizing sensor networks. The advancement of computing power and simulation tools has also contributed to the development and adoption of modern optimization algorithms, enabling researchers and practitioners to tackle increasingly complex optimization problems in sensor networks.

MOTIVATION:

The motivation behind this project stems from the growing significance of sensor networks in modern society and the persistent challenges associated with optimizing their performance. Sensor networks play a pivotal role in various domains, including environmental monitoring, health-care, transportation, and smart infrastructure. However, the efficient operation of these networks relies heavily on effective optimization techniques to ensure maximum coverage, minimal resource utilization, and adaptability to dynamic environments. Despite significant advancements in sensor technology and network protocols, optimizing sensor networks remains a complex and challenging task due to factors such as limited resources, communication constraints, and environmental dynamics. Therefore, there is a pressing need to explore and develop innovative optimization algorithms capable of addressing these challenges and unlocking the full potential of sensor networks.

1.2 PROBLEM STATEMENT:

The optimization of sensor networks to maximize coverage is a multifaceted problem with several inherent challenges. One of the primary challenges is to determine the optimal placement of sensors within a given area to ensure complete coverage while minimizing overlap and redundancy. Additionally, factors such as limited resources, communication constraints, and environmental conditions further complicate the optimization process.Moreover, the emergence of new applications, such as Internet of Things (IoT) and smart cities, has heightened the importance of optimizing sensor networks to support diverse and evolving requirements.

One of the key challenges is to generate random coordinates within the deployment area and tune the sensing radii of sensors to ensure comprehensive coverage while avoiding redundancy and unnecessary overlap. This involves developing algorithms and techniques that can efficiently explore the solution space and identify optimal sensor placements. Additionally, there is a need to consider factors such as the terrain, environmental conditions, and communication range constraints when designing the sensor network deployment.

Furthermore, the problem statement includes the optimization of sensor placements using the Particle Swarm Optimization (PSO) algorithm. PSO is a metaheuristic optimization technique inspired by the social behavior of birds flocking or fish schooling. It iteratively updates the positions of particles (representing potential sensor locations) based on their own experiences and the collective knowledge of the swarm, converging towards optimal solutions. Integrating PSO with random coordinate generation and radius tuning can lead to more efficient and effective sensor network deployments.

Overall, the problem statement focuses on developing methodologies and algorithms to address the challenges associated with sensor network optimization, including random coordinate generation, radius tuning, and utilization of the PSO algorithm for maximizing coverage and resource utilization while minimizing overlap in sensor deployments.

1.3 OBJECTIVE:

The primary objective of this project is to develop and implement algorithms for optimizing sensor networks to maximize coverage. Specifically, the project aims to investigate and evaluate modern optimization algorithms, such as Particle Swarm Optimization (PSO) and Genetic Algorithms (GA), to achieve this objective. Additionally, the project aims to compare the performance of these algorithms against traditional optimization methods and identify the most suitable approach for different applications and scenarios.

The objective of this project is to utilize the Particle Swarm Optimization (PSO) algorithm for optimizing sensor network deployments. Specifically, the project aims to achieve the following objectives:

- ➢ Random Coordinate Generation: Implement a method for generating random coordinates within a specified area to represent potential sensor locations.
- ➢ Radius Tuning: Develop a mechanism for tuning the sensing radius of individual sensors based on their placement, aiming to maximize coverage while minimizing resource consumption and overlap.
- ➢ Maximizing Coverage: Utilize the PSO algorithm to iteratively adjust sensor positions and coverage radii, with the goal of maximizing the coverage area of the sensor network.
- ➢ Proper Resource Utilization: Ensure efficient utilization of resources by optimizing sensor placements to avoid redundancy and unnecessary overlap.
- ➢ Minimizing Overlapping: Mitigate the occurrence of overlapping coverage areas by dynamically adjusting sensor positions and radii during the optimization process.

SCOPE:

The scope of this project encompasses the optimization of sensor networks for various applications, including environmental monitoring, surveillance, and industrial automation. The project will consider both static and dynamic deployment scenarios and evaluate the scalability and adaptability of the proposed algorithms. However, the project will not delve into hardware design or implementation details but will focus solely on algorithm development and simulation-based evaluations.

1.4 WIRELESS SENSOR NETWORK (WSN):

A Wireless Sensor Network (WSN) is a network of autonomous sensors distributed throughout a geographical area to collect and transmit data about the surrounding environment wirelessly. Each sensor node is equipped with one or more sensors to monitor specific physical or environmental parameters such as temperature, humidity, light, or motion. These nodes communicate with each other and with a central base station using wireless communication protocols like Zigbee or Bluetooth. WSNs are characterized by their ability to self-organize, allowing nodes to dynamically form and reconfigure network topological as needed. Energy efficiency is a critical consideration in WSN design, as sensor nodes are typically battery-powered and deployed in remote or inaccessible locations. To prolong network lifetime, energy-efficient protocols and algorithms are employed to minimize energy consumption. WSNs have diverse applications including environmental monitoring, industrial automation, healthcare systems, and military operations, providing a cost-effective solution for collecting data over large areas or in challenging environments where traditional wired solutions are impractical.

In addition to their primary applications, Wireless Sensor Networks (WSNs) continue to evolve and find new uses in emerging fields. In agriculture, precision farming techniques leverage WSNs to optimize crop yields, conserve water resources, and minimize the use of fertilizers and pesticides through targeted application based on real-time environmental data. Smart grid systems employ WSNs for monitoring energy consumption, detecting power outages, and optimizing energy distribution, contributing to more efficient and reliable electricity networks. In transportation, WSNs enable intelligent traffic management, vehicle tracking, and congestion avoidance, leading to safer and more efficient transportation systems. In urban planning, WSNs facilitate smart city initiatives by providing data on infrastructure usage, waste management, and environmental conditions, supporting sustainable development and improving quality of life for residents. Furthermore, research in fields such as wildlife conservation, space exploration, and underwater exploration continues to push the boundaries of WSN technology, expanding its potential applications and capabilities.

Why Wireless Sensor Network (WSN):

Wireless Sensor Networks (WSNs) are essential for remote monitoring in challenging environments where wired solutions are impractical or costly. WSNs offer real-time data collection, energy efficiency, scalability, and adaptability, making them ideal for applications such as environmental monitoring, industrial automation, and disaster response. With WSNs, organizations can make timely decisions, reduce costs, and efficiently manage resources.

- **Remote Monitoring:**WSNs enable remote monitoring in challenging environments where wired solutions are impractical or costly.
- **Real-Time Data Collection:**WSNs provide real-time data collection capabilities, facilitating timely decision-making in various applications.
- **Energy Efficiency:**Energy-efficient protocols and algorithms in WSNs prolong network lifetime, crucial for battery-powered sensor nodes.
- **Cost-Effectiveness:**WSNs offer a cost-effective alternative to wired infrastructure, reducing installation and maintenance expenses.
- **Scalability:**WSNs can scale to accommodate large numbers of sensor nodes, covering vast geographical areas or densely populated regions.
- **Flexibility and Adaptability:**WSNs can be rapidly deployed and reconfigured to suit changing monitoring requirements or environmental conditions.
- **Fault Tolerance:**WSNs are fault-tolerant due to their distributed nature, enabling network reconfiguration in case of node failure or communication disruptions.

Applications and use cases of WSN:

Wireless Sensor Networks (WSNs) are deployed in environmental monitoring, industrial automation, healthcare, agriculture, infrastructure monitoring, smart cities, and wildlife tracking. They collect data on temperature, machinery performance, patient health, soil moisture and wildlife behavior.

- **Environmental Monitoring:**WSNs are deployed in forests, oceans, and urban areas to monitor parameters like temperature, humidity, air quality, and pollution levels, aiding in environmental conservation and management.
- **Industrial Automation:**In industrial settings, WSNs monitor machinery performance, track inventory, optimize production processes, and ensure workplace safety by detecting factors like temperature, pressure, vibration, and chemical levels.

- **Healthcare Monitoring:**WSNs enable remote monitoring of patients' vital signs, medication adherence, and movement patterns, improving healthcare delivery and allowing for early detection of health issues.
- **Smart Agriculture:**In precision agriculture, WSNs monitor soil moisture, temperature, crop health, and weather conditions to optimize irrigation, fertilization, and pest control, thereby enhancing crop yields and conserving resources.
- **Smart Cities:**WSNs in smart cities monitor traffic flow, parking availability, air quality, noise levels, and waste management, facilitating urban planning and resource allocation
- **Disaster Management:**During natural disasters, WSNs provide real-time data on environmental conditions, infrastructure damage, and emergency response efforts, aiding in rescue operations, decision-making, and disaster recovery.
- **Wildlife Tracking:**WSNs are used to track wildlife movements, behavior patterns, and habitat conditions, assisting in conservation efforts and wildlife management strategies.

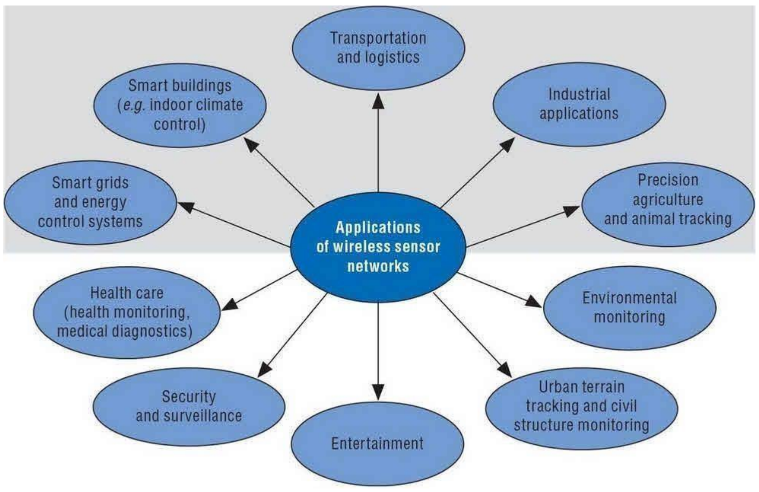

Applications of wireless sensor networks
Transportation and logistics
Industrial applications
Precision agriculture and animal tracking
Environmental monitoring
Urban terrain tracking and civil structure monitoring
Entertainment
Security and surveillance
Health care (health monitoring, medical diagnostics)
Smart grids and energy control systems
Smart buildings (e.g. indoor climate control)

Fig 2: Applications Of WSN

1.5 COVERAGE TECHNIQUES IN WSN:

Coverage techniques in Wireless Sensor Networks (WSNs) are essential for ensuring that the monitored area is adequately covered by sensor nodes. Techniques such as random deployment, deterministic deployment, grid-based deployment, Voronoi tessellation, optimization-based deployment, mobile sensor deployment, collaborative coverage, and barrier coverage are commonly employed. These techniques aim to achieve uniform coverage, minimize coverage gaps, and optimize resource utilization. By strategically deploying sensor nodes or dynamically adjusting their positions, WSNs can effectively monitor the environment, detect events, and gather valuable data for various applications such as environmental monitoring, surveillance, and infrastructure management.

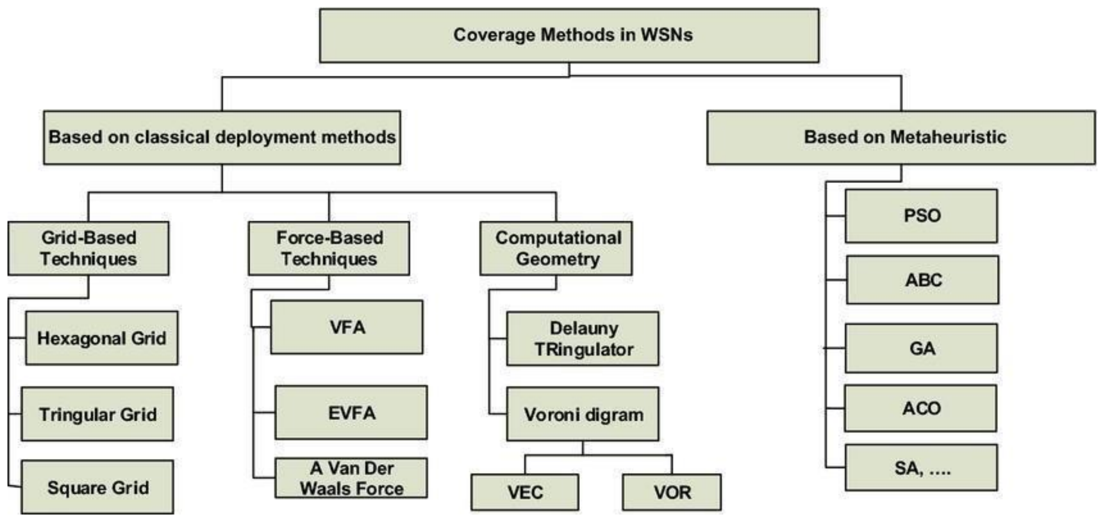


Fig 3: Coverage Methods in WSNs

In the context of WSNs, achieving optimal coverage and connectivity is essential for ensuring the effectiveness and reliability of the network. Various factors, such as the deployment strategy, node mobility, and environmental conditions, influence coverage and connectivity. Classical deployment techniques, including grid-based deployment and Voronoi tessellation, aim to distribute nodes evenly to maximize coverage. However, these methods may not be suitable for irregularly shaped areas or dynamic environments. Meta-heuristic techniques, such as genetic algorithms and particle swarm optimization, offer more flexible and adaptive approaches to node placement, optimizing coverage while considering factors like energy consumption and network lifetime. Additionally, self-scheduling techniques allow nodes to dynamically adjust their positions based on real-time data and network conditions, further enhancing coverage and connectivity. By leveraging a combination of these techniques and adapting them to specific deployment scenarios, WSNs can achieve robust coverage and connectivity tailored to the application's requirements.

Particle Swarm Optimization (PSO):

Particle Swarm Optimization (PSO) is a powerful meta-heuristic optimization algorithm and inspired by swarm behavior observed in nature such as fish and bird schooling. PSO is a Simulation of a simplified social system. The original intent of PSO algorithm was to graphically simulate the graceful but unpredictable choreography of a bird flock.In nature, any of the bird’s observable vicinity is limited to some range. However, having more than one birds allows all the birds in a swarm to be aware of the larger surface of a fitness function.

PSO operates by simulating the movement of particles within a multidimensional search space, with each particle representing a potential solution to the optimization problem. Initially, particles are randomly distributed throughout the search space. As the algorithm progresses, particles adjust their positions based on their own experience (personal best) and the best positions found by neighboring particles (global best). This iterative process continues until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory solution.

The movement of particles in PSO is governed by two main parameters: velocity and position. The velocity of each particle determines its direction and magnitude of movement within the search space, while the position represents the current solution represented by the particle. By updating velocity and position based on the personal best and global best solutions, particles are guided towards regions of the search space that are likely to contain optimal solutions.

PSO has been successfully applied to a wide range of optimization problems in various domains, including engineering, finance, and computer science. Its simplicity, efficiency, and ability to handle complex search spaces make it a popular choice for optimization tasks where traditional methods may be impractical or ineffective.

In the context of Wireless Sensor Networks (WSNs), PSO can be employed to optimize sensor placement, coverage, routing, and energy consumption. By leveraging PSO, WSN designers can find optimal configurations that enhance network performance, reliability, and efficiency, ultimately improving the effectiveness of the entire network.

Mathematical model :

- Each particle in particle swarm optimization has an associated position, velocity, fitness value.
- Each particle keeps track of the particle_bestFitness_value particle_bestFitness_position.
- A record of global_bestFitness_position and global_bestFitness_value is maintained.

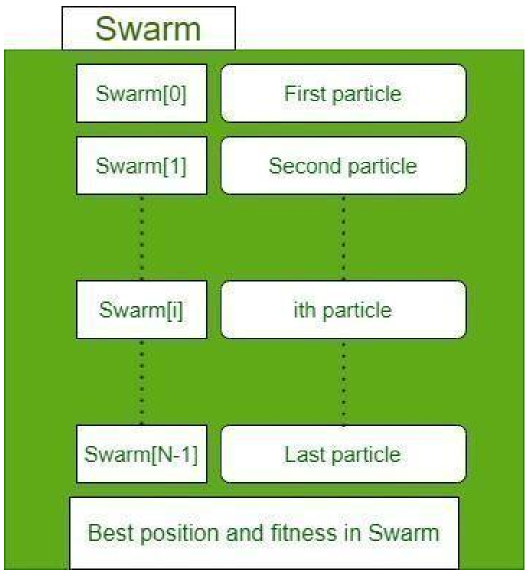


Fig 4: Data structure to store Swarm population

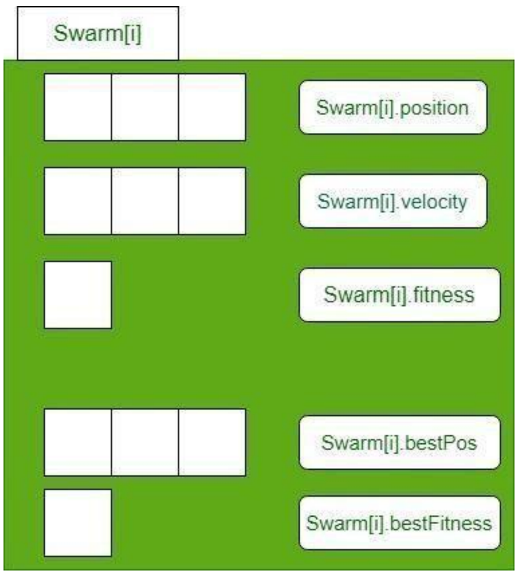


Fig 5: Data structure to store ith particle of Swarm

Algorithm:

```
Step1: Randomly initialize Swarm population of N particles Xi ( i=1, 2, …, n)
Step2: Select hyperparameter values
          w, c1 and c2
Step 3: For Iter in range(max_iter): # loop max_iter times
           For i in range(N): # for each particle:
              a. Compute new velocity of ith particle
                  swarm[i].velocity =
                      w*swarm[i].velocity +
                      r1*c1*(swarm[i].bestPos - swarm[i].position) +
                      r2*c2*( best_pos_swarm - swarm[i].position)
              b. Compute new position of ith particle using its new velocity
                  swarm[i].position += swarm[i].velocity
              c. If position is not in range [minx, maxx] then clip it
                  if swarm[i].position < minx:
                     swarm[i].position = minx
                  elif swarm[i].position > maxx:
                     swarm[i].position = maxx
              d. Update new best of this particle and new best of Swarm
                   if swaInsensitive to scaling of design variables.rm[i].fitness < swarm[i].bestFitness:
                     swarm[i].bestFitness = swarm[i].fitness
                     swarm[i].bestPos = swarm[i].position

                   if swarm[i].fitness < best_fitness_swarm
                     best_fitness_swarm = swarm[i].fitness
                     best_pos_swarm = swarm[i].position
            End-for
        End -for
Step 4: Return best particle of Swarm
```

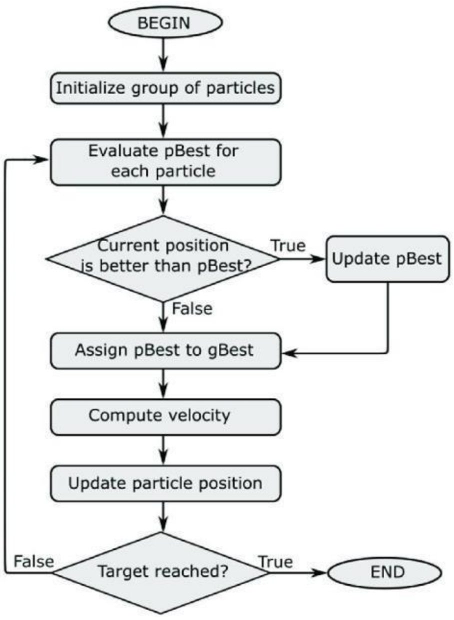


Fig 6: PSO Flow Chart

Role of PSO In Sensor Optimization:

Particle Swarm Optimization (PSO) plays a pivotal role in sensor optimization within Wireless Sensor Networks (WSNs), offering an efficient solution for addressing complex optimization problems. One of its primary roles is optimizing sensor placement within the monitored area to maximize coverage while minimizing resource consumption. By iteratively adjusting sensor positions based on their performance and the performance of neighboring nodes, PSO can find optimal configurations that enhance coverage and reliability. Additionally, PSO can optimize sensor operation to minimize energy consumption and extend the lifespan of battery-powered nodes by adjusting parameters such as transmission power and duty cycle. Furthermore, PSO can optimize routing protocols to minimize communication overhead and latency, ensuring efficient data transmission and reliable communication within the network. Overall, PSO plays a crucial role in enhancing the performance, reliability, and efficiency of sensor networks in various applications and environments by efficiently exploring solution spaces and finding near-optimal configurations.

Particle Swarm Optimization (PSO) plays a pivotal role in sensor optimization within Wireless Sensor Networks (WSNs), offering an efficient solution for addressing complex optimization problems. One of its primary roles is optimizing sensor placement within the monitored area to maximize coverage while minimizing resource consumption. By iteratively adjusting sensor positions based on their performance and the performance of neighboring nodes, PSO can find optimal configurations that enhance coverage and reliability. Additionally, PSO can optimize sensor operation to minimize energy consumption and extend the lifespan of battery-powered nodes by adjusting parameters such as transmission power and duty cycle. Furthermore, PSO can optimize routing protocols to minimize communication overhead and latency, ensuring efficient data transmission and reliable communication within the network. Moreover, PSO's ability to adapt to changing environmental conditions and network dynamics makes it particularly well-suited for dynamic and unpredictable scenarios, such as those encountered in outdoor environments or disaster response situations.

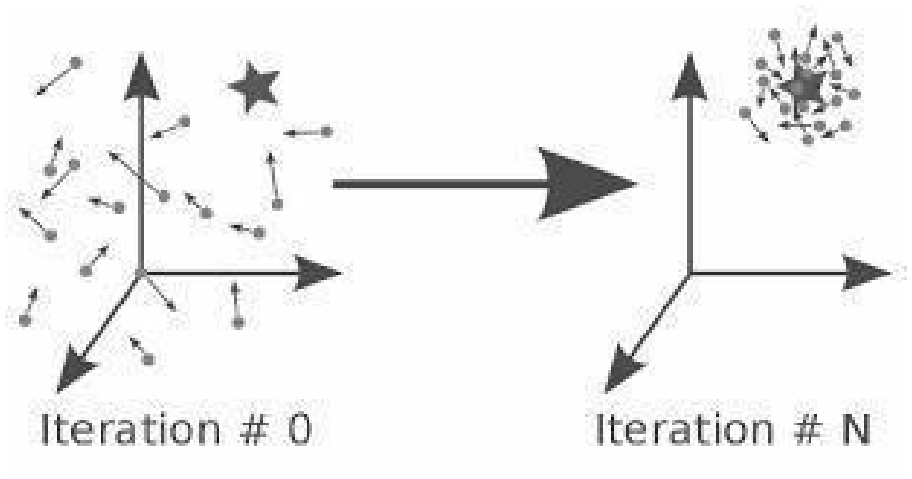


Fig 7: PSO Iterations

CHAPTER-2
EXISTING MODELS

2.1 EXISTING TECHNIQUES:

Heuristic techniques such as GA, PSO, ACO, and FFA serve as practical approaches for optimizing Wireless Sensor Networks (WSNs). These meta heuristic algorithms efficiently solve complex optimization problems in WSNs, facilitating tasks such as sensor placement optimization or energy-efficient routing configurations. On the other hand, clustering techniques like k-clustering, grid-based clustering, density-based clustering, and hierarchical clustering play a crucial role in data aggregation and processing within WSNs by partitioning sensor nodes into clusters. This approach enables local data aggregation, reducing energy consumption and communication overhead. Meanwhile, clustering protocols like LEACH, HEED, PEGASIS, TEEN, DEEC, and LEACH-C dynamically organize sensor nodes into clusters, distributing energy consumption evenly across the network. These protocols employ various strategies such as adaptive clustering, centralized clustering, or hierarchical communication to optimize energy efficiency and network scalability. Together, these heuristic techniques, clustering techniques, and clustering protocols contribute to optimizing WSNs for improved performance, energy efficiency, and scalability across various applications and environments.

Heuristic techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Firefly Algorithm (FFA) play a vital role in optimizing Wireless Sensor Networks (WSNs) for various applications. These metaheuristic algorithms offer practical approaches for solving complex optimization problems in WSNs, where traditional methods may be impractical due to large solution spaces or non-linear objective functions. For instance, GA can be utilized to optimize sensor placement or routing configurations, while PSO can efficiently explore the search space to find near-optimal solutions. Similarly, ACO and FFA can be employed to improve energy efficiency or enhance network performance through adaptive algorithms inspired by natural phenomena.

In addition to k-clustering, other clustering techniques are also utilized in Wireless Sensor Networks (WSNs) to optimize data aggregation and processing. Grid-based clustering divides the sensor field into a grid and assigns each grid cell to a cluster, enabling spatially distributed data aggregation. Density-based clustering identifies dense regions of sensor nodes and forms clusters based on node density, adapting to varying node distributions and environmental conditions. Hierarchical clustering organizes sensor nodes into a hierarchical structure, allowing for multi-level data aggregation and scalable communication. These clustering techniques complement k-clustering by providing alternative approaches to partitioning sensor nodes into clusters based on different criteria such as spatial proximity, node density, or network topology. Together, these clustering techniques contribute to efficient data aggregation and processing in WSNs, reducing energy consumption and communication overhead while prolonging the network lifetime.

Clustering protocols like LEACH organize sensor nodes into clusters, evenly distributing energy consumption in WSNs, optimizing energy efficiency and scalability. HEED and PEGASIS offer innovative approaches, combining centralized and distributed mechanisms or chain-based communication to further minimize energy consumption during data transmission. Additionally, protocols like TEEN and DEEC dynamically adjust clustering parameters based on network conditions, enhancing energy efficiency and reliability. These protocols collectively optimize energy consumption and network scalability in WSNs.

2.2 HEURISTIC TECHNIQUES:

Heuristic techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and Firefly Algorithm (FFA) are instrumental in optimizing Wireless Sensor Networks (WSNs) across various applications. These metaheuristic algorithms offer practical solutions for tackling complex optimization problems encountered in WSNs, where traditional methods may falter due to large solution spaces or non-linear objective functions. For example, GA can efficiently optimize sensor placement or routing configurations by iteratively evolving candidate solutions based on natural selection principles. PSO excels in exploring the search space to find near-optimal solutions, making it valuable for optimizing network parameters such as coverage and energy efficiency. Similarly, ACO and FFA leverage adaptive algorithms inspired by natural phenomena like ant foraging behavior and firefly flashing patterns to improve energy efficiency and enhance network performance. By harnessing the collective intelligence of these heuristic techniques, WSNs can achieve optimized configurations that meet the specific requirements of diverse applications while overcoming the inherent challenges of dynamic and resource-constrained environments.Heuristic techniques like GA, PSO, ACO, and FFA offer versatile approaches for optimizing WSNs across various domains. Their adaptability and efficiency make them well-suited for addressing complex optimization tasks such as sensor placement, coverage optimization, routing, and energy management. By leveraging principles inspired by natural phenomena, these algorithms effectively navigate the intricate solution spaces of WSNs, providing robust and scalable solutions. Their widespread use underscores their significance in enhancing the performance, reliability, and efficiency of WSNs in real-world applications.

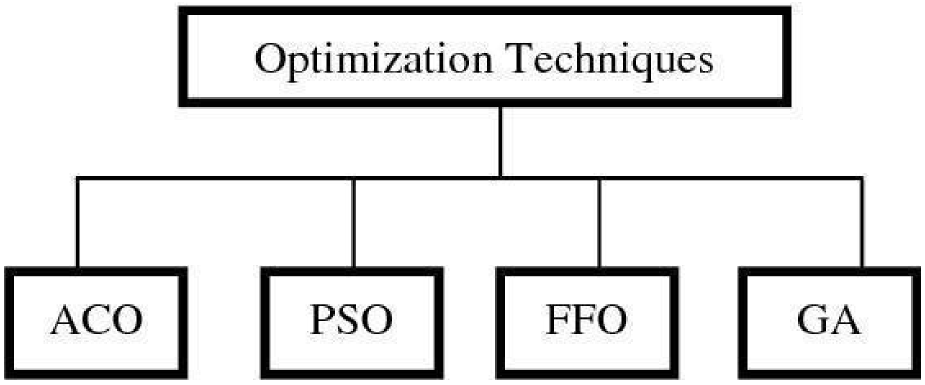


Fig 8: Optimization Techniques

1. **Particle Swarm Optimization (PSO):**

Particle Swarm Optimization (PSO) is a robust meta-heuristic optimization algorithm inspired by swarm behavior observed in nature, such as fish and bird schooling. It simulates a simplified social system with the original intention of graphically emulating the graceful yet unpredictable choreography of a bird flock. In PSO, particles represent potential solutions to the optimization problem and move within a multidimensional search space. Initially distributed randomly, particles adjust their positions iteratively based on personal best and global best solutions found by neighboring particles. This movement continues until a stopping criterion is met, such as reaching a maximum number of iterations or achieving a satisfactory solution. Governed by velocity and position parameters, particles navigate towards promising regions of the search space, driven by the desire to find optimal solutions. In nature, the perceptual range of individual birds within a flock is limited, but the collective behavior of the entire swarm allows for a broader awareness of the fitness landscape. Similarly, in PSO, each particle's exploration is guided not only by its own experience but also by the collective knowledge shared among neighboring particles. This collaboration enables PSO to efficiently navigate complex and high-dimensional search spaces, effectively balancing exploration and exploitation to find optimal solutions. The success of PSO lies in its adaptability and versatility, allowing it to tackle a wide range of optimization problems across diverse domains. In engineering, PSO has been applied to tasks such as structural optimization, parameter tuning in machine learning algorithms, and optimal design of electrical circuits. In finance, PSO has been utilized for portfolio optimization, option pricing, and risk management strategies. Furthermore, in computer science, PSO has been employed for feature selection, image processing, and neural network training. One of the key advantages of PSO is its simplicity and ease of implementation compared to other optimization algorithms.

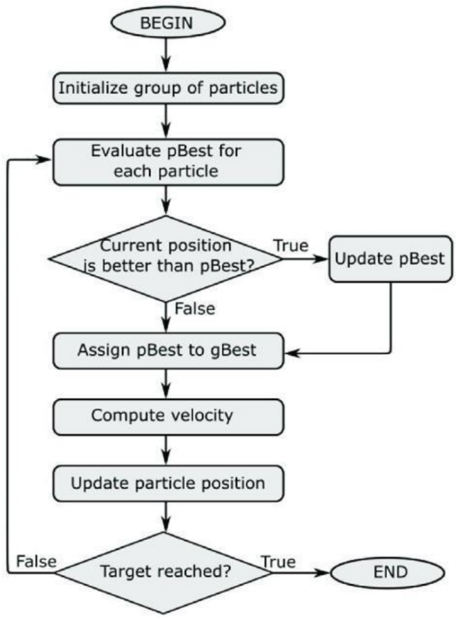


Fig 9: PSO Flow Chart

2. Genetic Algorithm (GA):

Genetic Algorithm (GA) is a powerful optimization technique inspired by the principles of natural selection and genetics. Developed by John Holland in the 1970s, GA mimics the process of evolution to search for optimal solutions to complex problems. At its core, GA operates by maintaining a population of candidate solutions represented as individuals or chromosomes. These chromosomes encode potential solutions to the optimization problem and undergo genetic operations such as selection, crossover, and mutation to generate new offspring. Through iterative generations, the algorithm evolves the population, favoring individuals with higher fitness values that better satisfy the objective function. Selection mechanisms, such as tournament selection or roulette wheel selection, determine which individuals are chosen for reproduction based on their fitness.

Crossover and mutation operators recombine genetic information from selected parents to produce offspring with potentially improved characteristics. The iterative process continues until a termination condition is met, such as reaching a maximum number of generations or achieving a satisfactory solution. GA has demonstrated remarkable success in solving a wide range of optimization problems, including function optimization, scheduling, and machine learning. Its ability to efficiently explore large search spaces and discover near-optimal solutions makes it a popular choice for optimization tasks where traditional methods may be impractical or ineffective.

Ongoing research in GA focuses on improving performance, scalability, and convergence properties, as well as extending its applicability to new problem domains and real-world applications.Furthermore, Genetic Algorithm (GA) offers several key advantages that contribute to its widespread adoption and effectiveness in solving optimization problems. One notable advantage is its ability to handle complex, nonlinear, and multimodal objective functions, making it suitable for a wide range of real-world applications across various domains.

Additionally, GA's inherent parallelism allows for efficient exploration of the search space, enabling faster convergence and scalability, particularly in high-dimensional optimization problems.

Moreover, GA's population-based approach ensures diversity within the solution space, reducing the risk of premature convergence to suboptimal solutions.

This diversity promotes exploration of different regions of the search space, increasing the likelihood of discovering globally optimal solutions. Furthermore, GA's flexible representation of solutions and customizable genetic operators make it adaptable to different problem domains and solution representation.

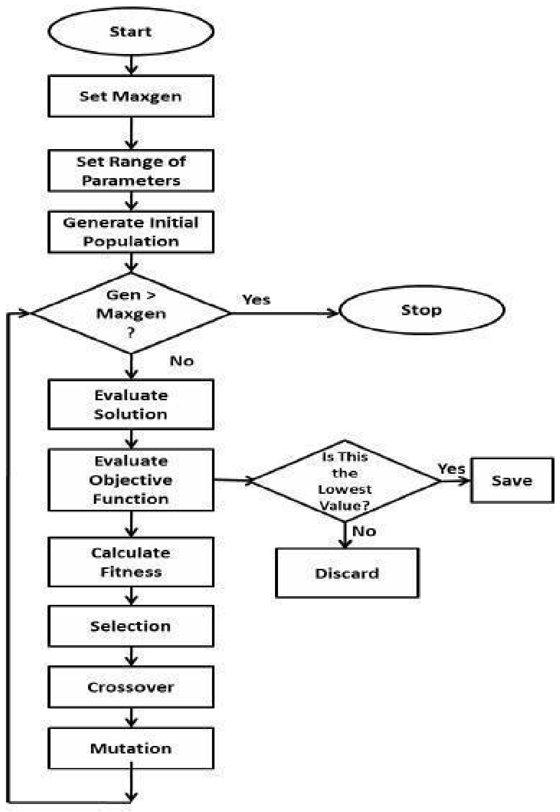


Fig 10: GA Flow Chart

2.3 CLUSTERING TECHNIQUES:

Clustering techniques play a crucial role in organizing data into meaningful groups or clusters based on similarities among data points. Among the commonly used clustering methods are K-clustering, where the data is partitioned into K clusters by assigning each data point to the cluster with the nearest centroid, updated iteratively until convergence. Grid-based clustering divides the data space into a grid structure and assigns points to grid cells, forming clusters based on data density within cells. Density-based clustering algorithms, such as DBSCAN, identify clusters based on regions of high data density, distinguishing core points from noise or border points.

Meanwhile, hierarchical clustering constructs a hierarchy of clusters by recursively merging or splitting clusters based on their similarity, resulting in a dendrogram that can be cut at different levels to obtain clusters at various granularity levels. These clustering techniques offer versatile approaches to data partitioning, each with its strengths and weaknesses, enabling researchers and practitioners to tailor clustering analyses to the characteristics of the data and the objectives of the study.Clustering techniques are widely used in various fields such as data mining, pattern recognition, and machine learning for tasks like customer segmentation, anomaly detection, and image segmentation.

Each clustering method has its own set of parameters and assumptions, and the choice of technique depends on factors such as the nature of the data, the desired number of clusters, and the computational resources available. Researchers continue to explore and develop new clustering algorithms to address challenges such as handling high-dimensional data, scalability to large datasets, and robustness to noise and outliers.

Clustering Techniques:

- K-clustering
- Grid-based clustering
- Density-based clustering
- Hierarchical clustering

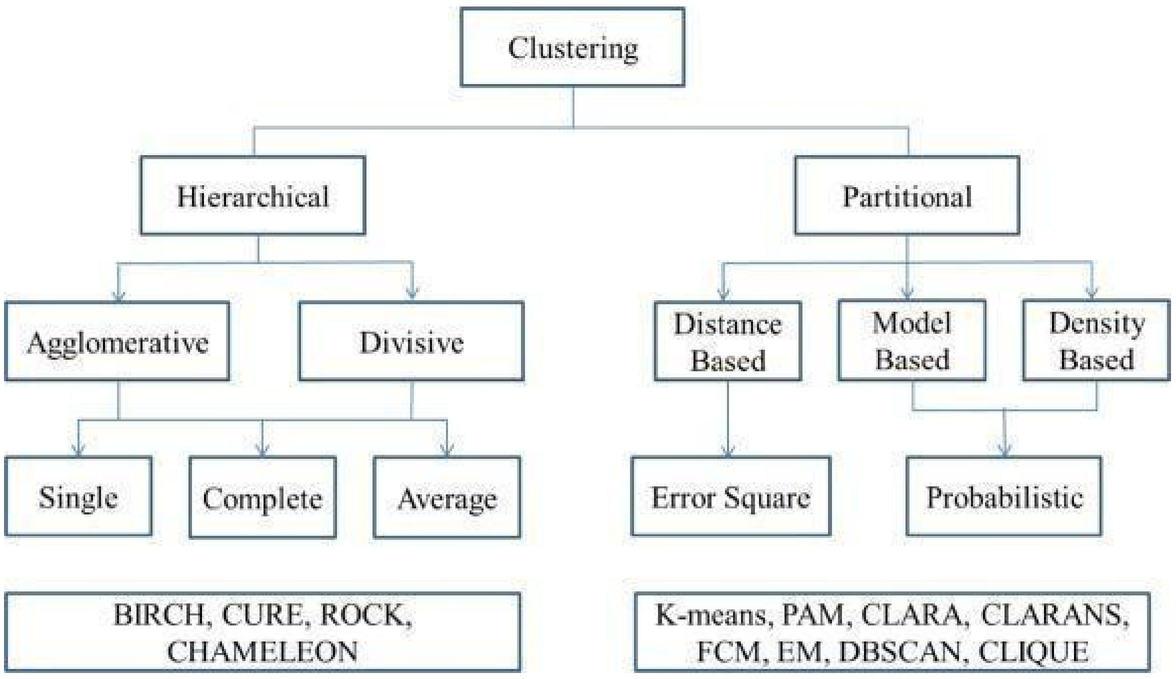


Fig 11: Clustering Techniques

1. Grid-based clustering:

Grid-based clustering is a data partitioning technique that divides the data space into a grid structure and assigns data points to grid cells based on their coordinates. This approach simplifies the clustering process by organizing the data into a grid, making it easier to identify clusters based on the density of data points within grid cells. Grid-based clustering algorithms typically involve two main steps: grid construction and cluster formation. In the grid construction step, the data space is divided into a grid of equal-sized cells. The size of the grid cells can be predefined or adjusted dynamically based on the distribution of data points. Once the grid is constructed, data points are mapped to the corresponding grid cells based on their spatial coordinates. In the cluster formation step, clusters are formed based on the density of data points within grid cells. Cells with a high density of data points are considered as core cells, while cells with lower densities are classified as noise or border cells. Clusters are then formed by grouping neighboring core cells and their associated data points. Grid-based clustering offers several advantages, including simplicity, scalability, and efficiency. It is particularly well-suited for datasets with a uniform distribution of data points or when the underlying data structure can be approximated by a grid. However, grid-based clustering may struggle with datasets that have irregular shapes or varying densities, as it may not accurately capture the underlying cluster structure in such cases. Overall, grid-based clustering is a useful technique for partitioning data into clusters based on spatial proximity and density, making it applicable to a wide range of data analysis tasks such as spatial data mining, image processing, and geographic information systems.Grid-based clustering has found applications in various fields such as geographic information systems (GIS), where it is used for spatial data analysis, land cover classification, and urban planning. Additionally, in image processing, grid-based clustering techniques are employed for segmentation tasks to partition images into regions of interest based on pixel intensities or textures. One of the key advantages of grid-based clustering is its simplicity and computational efficiency, making it suitable for large-scale datasets and real-time applications. However, its performance may degrade when dealing with high-dimensional data or datasets with non-uniform distributions.

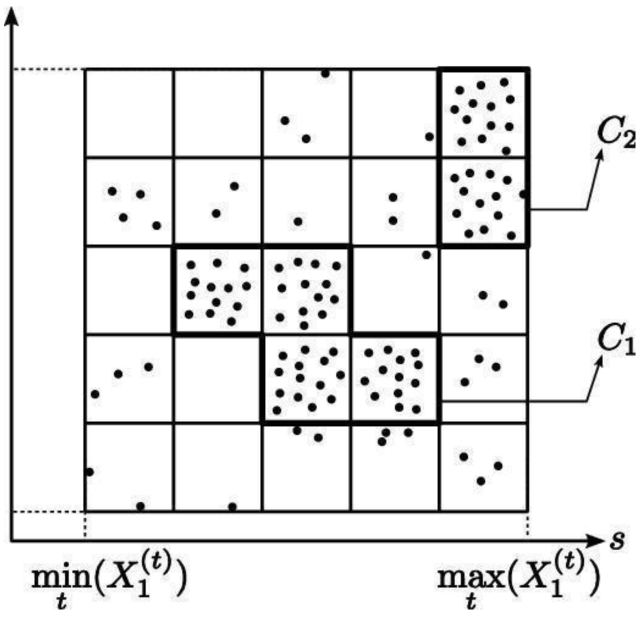


Fig 12: Grid-Based Clustering

2. K-clustering:

K-clustering, also known as K-means clustering, is a popular unsupervised learning algorithm used for partitioning data into K clusters. The algorithm works by iteratively assigning data points to the nearest cluster centroid and updating the centroids based on the mean of the data points assigned to each cluster. This process continues until the centroids converge or a specified number of iterations is reached. K-clustering aims to minimize the within-cluster variance, ensuring that data points within the same cluster are similar to each other while being different from data points in other clusters. The value of K, representing the number of clusters, needs to be specified beforehand, and the algorithm's performance may vary depending on the initial placement of centroids. Despite its simplicity and efficiency, K-clustering may struggle with non-linear or non-spherical clusters and is sensitive to outliers. Various techniques, such as the use of multiple initializations and silhouette analysis, can be employed to improve the robustness and effectiveness of K-clustering in practice.K-clustering finds applications in a wide range of domains, including data mining, pattern recognition, image segmentation, and customer segmentation. In data mining, it is used for exploratory analysis to uncover hidden patterns and structures within datasets. In pattern recognition, K-clustering is employed for clustering similar patterns or objects based on their features. Image segmentation tasks involve partitioning images into regions or objects of interest, with K-clustering being a common technique used for this purpose. Despite its limitations, K-clustering remains one of the most widely used clustering algorithms due to its simplicity, scalability, and effectiveness in many practical scenarios.

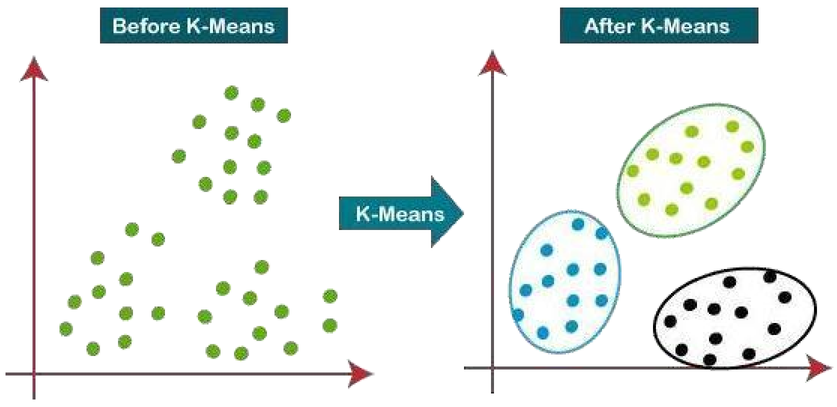


Fig 13: K-Means Clustering

K-Means Algorithm:

Step-1: Select the number K to decide the number of clusters.

Step-2: Select random K points or centroids. (It can be other from the input dataset).

Step-3: Assign each data point to their closest centroid, which will form the predefined K clusters.

Step-4: Calculate the variance and place a new centroid of each cluster.

Step-5: Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.

Step-6: If any reassignment occurs, then go to step-4 else go to FINISH.

2.4 CLUSTERING PROTOCOLS:

Clustering protocols play a crucial role in Wireless Sensor Networks (WSNs) by organizing sensor nodes into clusters to efficiently manage energy consumption and communication overhead. LEACH (Low-Energy Adaptive Clustering Hierarchy) is a pioneering clustering protocol that dynamically selects cluster heads based on their residual energy, aiming to evenly distribute energy usage across the network. HEED (Hybrid Energy-Efficient Distributed clustering) combines centralized and distributed approaches to optimize cluster formation, considering both energy efficiency and network connectivity. PEGASIS (Power-Efficient GAthering in Sensor Information Systems) introduces a chain-based topology where sensor nodes transmit data in a sequential manner, reducing energy consumption through data aggregation and cooperative communication. TEEN (Threshold Sensitive Energy Efficient sensor Network protocol) employs threshold-sensitive techniques to adaptively adjust sensor node activity levels based on environmental conditions, optimizing energy usage while maintaining network coverage. DEEC (Distr ibuted Energy-Efficient Clustering) focuses on distributed cluster formation to prolong network lifetime by balancing energy consumption among sensor nodes. LEACH-C (Centralized LEACH) enhances the LEACH protocol by introducing a centralized control mechanism for cluster formation and management, improving scalability and energy efficiency in large-scale WSNs. These clustering protocols contribute to the energy-efficient operation of WSNs, enabling prolonged network lifetime and reliable data transmission for various applications, including environmental monitoring, industrial automation, and smart infrastructure.

Clustering Protocols:

- LEACH (Low-Energy Adaptive Clustering Hierarchy)
- HEED (Hybrid Energy-Efficient Distributed clustering)
- PEGASIS (Power-Efficient GAthering in Sensor Information Systems)
- TEEN (Threshold Sensitive Energy Efficient sensor Network protocol)
- DEEC (Distributed Energy-Efficient Clustering)
- LEACH-C (Centralized LEACH)

LEACH (Low-Energy Adaptive Clustering Hierarchy):

LEACH (Low-Energy Adaptive Clustering Hierarchy) is a pioneering clustering protocol designed for Wireless Sensor Networks (WSNs) to extend network lifetime and conserve energy. It operates by dynamically organizing sensor nodes into clusters and selecting cluster heads based on their residual energy levels. LEACH employs a distributed approach, allowing sensor nodes to autonomously form clusters and elect cluster heads in each round of operation. Cluster heads are responsible for aggregating data from cluster members and transmitting it to a central base station, reducing the energy consumption of individual sensor nodes. One of LEACH's key features is its adaptability, as it adjusts cluster head selection probabilities to evenly distribute energy usage across the network over time. By rotating cluster head roles among sensor nodes in a probabilistic manner, LEACH mitigates energy imbalances and prolongs network lifetime. This protocol is particularly well-suited for resource-constrained sensor nodes and dynamic environments where energy efficiency is critical.

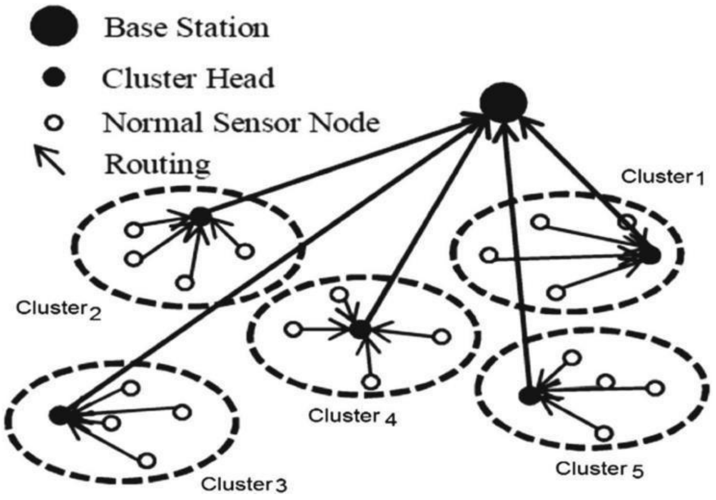


Fig 14: LEACH

CHAPTER-3

RANDOMIZED SENSOR PLACEMENT AND COVERAGE OPTIMIZATION IN A DEFINED AREA

The "Randomized Sensor Placement and Coverage Optimization in a Defined Area" program is a sophisticated computational tool designed to tackle the intricate challenge of strategically deploying sensors within a specified geographical region to maximize coverage and effectiveness. Leveraging a blend of randomized algorithms and optimization techniques, the program navigates the complex landscape of sensor placement, aiming to ensure comprehensive monitoring of the area while minimizing resource allocation. At its core, the program employs randomized algorithms to initially select sensor locations within the defined area, leveraging randomness to explore a wide range of possibilities and avoid getting stuck in local optima. Subsequently, optimization algorithms come into play, iteratively adjusting sensor positions to fine-tune coverage and eliminate redundancy. These algorithms take into account various factors such as geographical features, obstacles, and terrain, ensuring that sensor placement is optimized to suit the unique characteristics of the area under surveillance.

Algorithm:

1. It takes user input for the length and breadth of the area to be covered.
2. It calculates the total area to be covered.
3. It initializes lists to store sensor areas and sensor coordinates.
4. It generates random sensor coordinates and radii based on the size of the area. The radii are randomly chosen within certain ranges.
5. It calculates the coverage area for each sensor.
6. It repeats the process for a certain number of iterations (100 in this case).
7. It identifies the combination of sensors that covers the maximum area.
8. It plots circles representing the coverage of each sensor and marks their centers on a matplotlib plot.

In addition to its utility in environmental monitoring, security surveillance, and infrastructure management, the program's versatility extends to a myriad of other applications across various domains. In the realm of disaster management, it plays a crucial role in early warning systems by ensuring comprehensive coverage of disaster-prone areas with sensors that can detect anomalies such as seismic activity, temperature fluctuations, or changes in air quality. Similarly, in agricultural settings, the program facilitates precision agriculture by strategically deploying sensors to monitor soil moisture levels, temperature variations, and crop health, thereby optimizing irrigation schedules and maximizing yields. Furthermore, in urban planning and smart city initiatives, the program aids in enhancing public safety, traffic management, and resource allocation by optimizing the placement of surveillance cameras, traffic sensors, and IoT devices to monitor urban infrastructure .

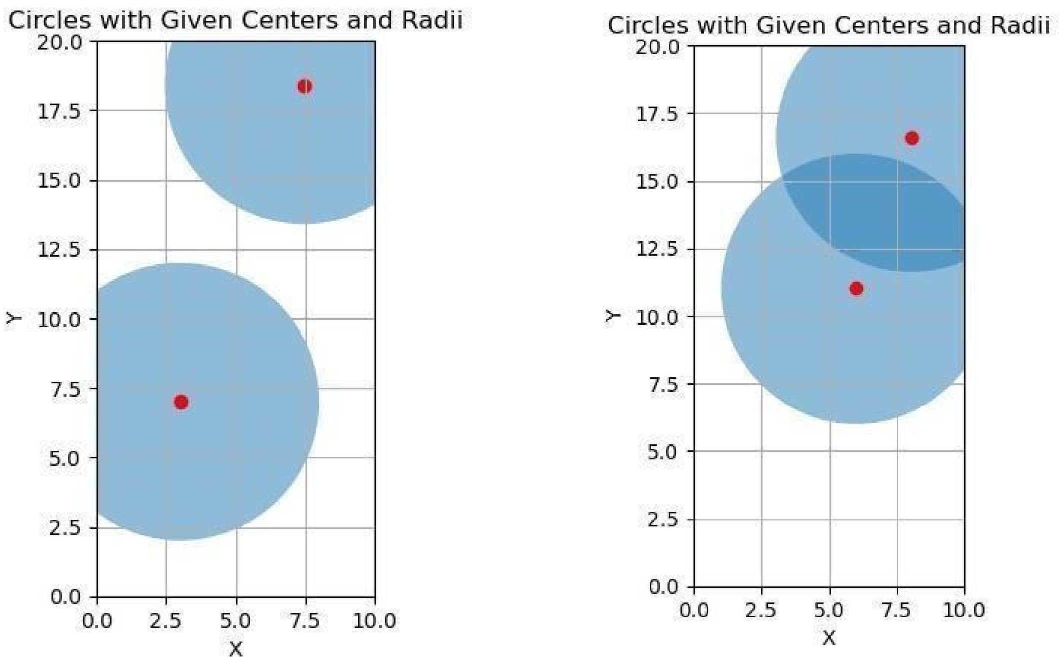


Fig 15: Coverage Areas of Sensors

3.1 MANUAL PLAECEMENT OF SENSORS WITH UNIFORM RADIUS:

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 :

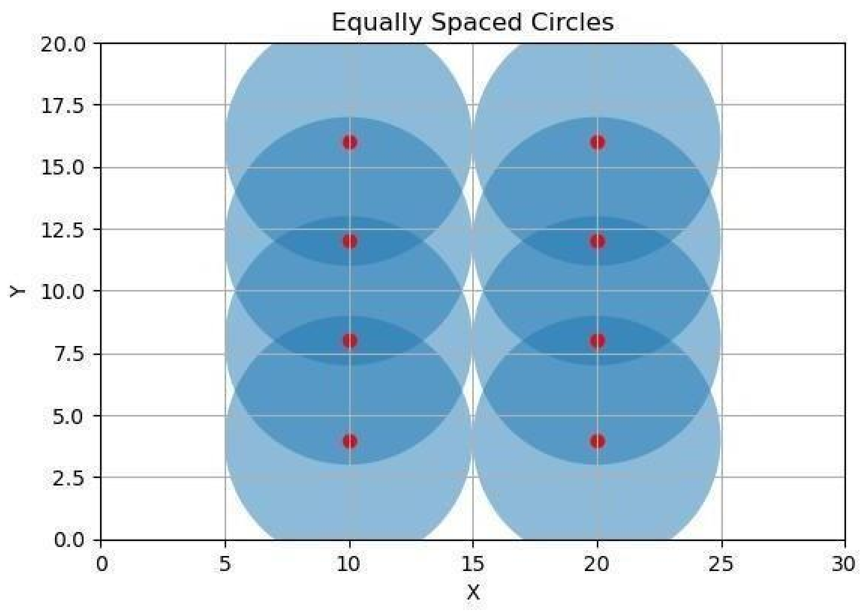


Fig 16: Equally Spaces Circles with 8 sensors

```
No.of Sensors: 8
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: 600
Total area covered by circles: 628.3185307179587
Total area uncovered by circles: -28.31853071795865
```

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.Additionally, redundant sensor coverage not only results in data duplication but also imposes additional computational overhead for data processing and analysis. The presence of overlapping sensor regions complicates data fusion algorithms, potentially leading to inaccuracies in the synthesized information.

Case-2 :

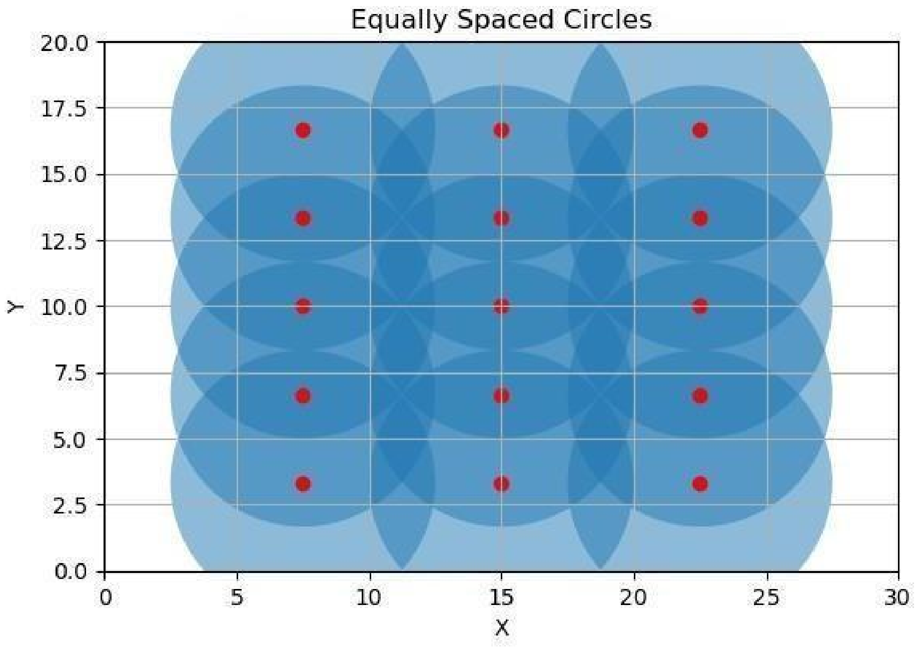


Fig 17: Equally spaced Circles with 15 sensors

```
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)
Total area: 600
Total area covered by circles: 1178.0972450961724
Total area uncovered by circles: -578.0972450961724
```

Case-3 :

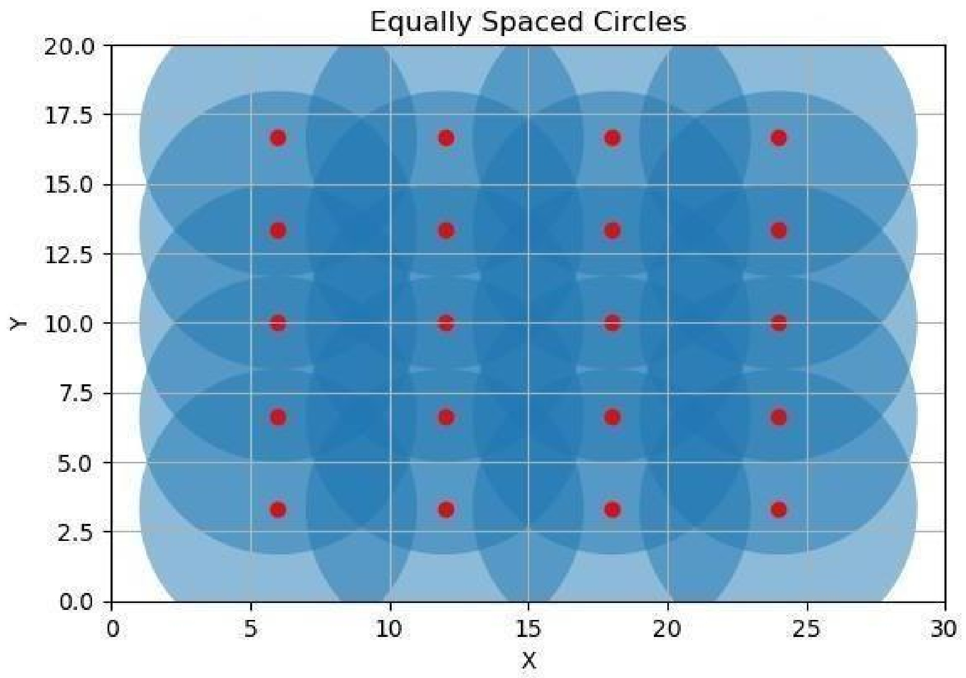


Fig 18: Equally spaced Circles with 20 sensors

```
No.of Sensors: 20
Coordinates of sensors:
Sensor 1: (6.0, 3.3333333333333335)
Sensor 2: (6.0, 6.666666666666667)
Sensor 3: (6.0, 10.0)
Sensor 4: (6.0, 13.333333333333334)
Sensor 5: (6.0, 16.666666666666668)
Sensor 6: (12.0, 3.3333333333333335)
Sensor 7: (12.0, 6.666666666666667)
Sensor 8: (12.0, 10.0)
Sensor 9: (12.0, 13.333333333333334)
Sensor 10: (12.0, 16.666666666666668)
Sensor 11: (18.0, 3.3333333333333335)
Sensor 12: (18.0, 6.666666666666667)
Sensor 13: (18.0, 10.0)
Sensor 14: (18.0, 13.333333333333334)
Sensor 15: (18.0, 16.666666666666668)
Sensor 16: (24.0, 3.3333333333333335)
Sensor 17: (24.0, 6.666666666666667)
Sensor 18: (24.0, 10.0)
Sensor 19: (24.0, 13.333333333333334)
Sensor 20: (24.0, 16.666666666666668)

Total area: 600
Total area covered by circles: 1570.796326794897
Total area uncovered by circles: -970.7963267948969
```

Case-4 :

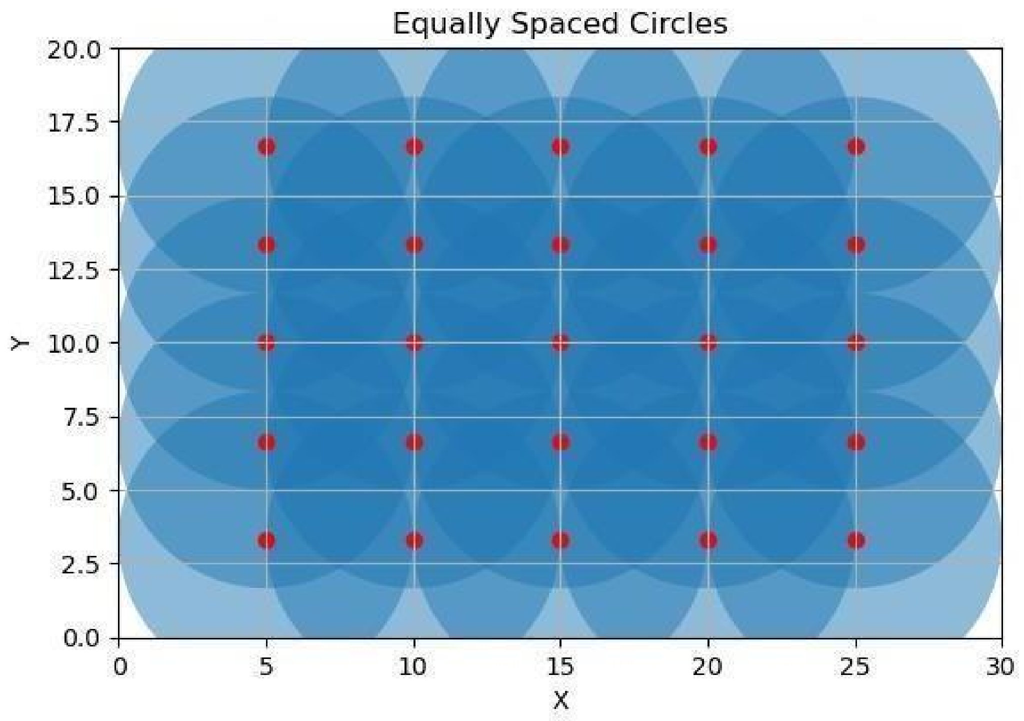


Fig 19: Equally spaced Circles with 25 sensors

```
No.of Sensors: 25
Coordinates of sensors:
Sensor 1: (5.0, 3.3333333333333335)
Sensor 2: (5.0, 6.666666666666667)
Sensor 3: (5.0, 10.0)
Sensor 4: (5.0, 13.333333333333334)
Sensor 5: (5.0, 16.666666666666668)
Sensor 6: (10.0, 3.3333333333333335)
Sensor 7: (10.0, 6.666666666666667)
Sensor 8: (10.0, 10.0)
Sensor 9: (10.0, 13.333333333333334)
Sensor 10: (10.0, 16.666666666666668)
Sensor 11: (15.0, 3.3333333333333335)
Sensor 12: (15.0, 6.666666666666667)
Sensor 13: (15.0, 10.0)
Sensor 14: (15.0, 13.333333333333334)
Sensor 15: (15.0, 16.666666666666668)
Sensor 16: (20.0, 3.3333333333333335)
Sensor 17: (20.0, 6.666666666666667)
Sensor 18: (20.0, 10.0)
Sensor 19: (20.0, 13.333333333333334)
Sensor 20: (20.0, 16.666666666666668)
Sensor 21: (25.0, 3.3333333333333335)
Sensor 22: (25.0, 6.666666666666667)
Sensor 23: (25.0, 10.0)
Sensor 24: (25.0, 13.333333333333334)
Sensor 25: (25.0, 16.666666666666668)
Total area: 600
Total area covered by circles: 1963.4954084936214
Total area uncovered by circles: -1363.4954084936214
```

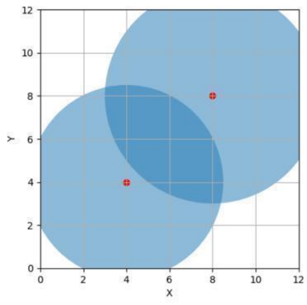
Table 1: Manual placement of Sensors with uniform Radius

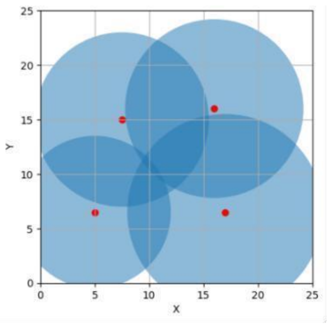
No.of Sensors	Total area of a region	Total area covered by sensors	Total area uncovered by sensors
8	600	628.3185307179587	-28.318530717958
15	600	1178.0972450961724	-578.09724509617
20	600	1570.796326794897	-970.79632679489
25	600	1963.4954084936214	-1363.4954084936

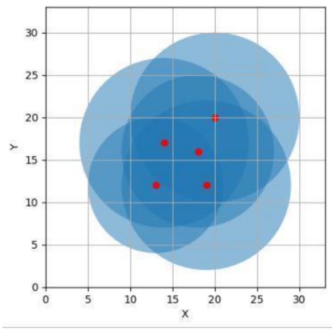
3.2 RANDOM COORDINATES GENERATION :

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.One approach to random coordinates generation is to use a uniform distribution.This ensures that the generated coordinates are spread evenly across the deployment region, preventing clustering or bias towards certain areas.Another consideration in random coordinates .







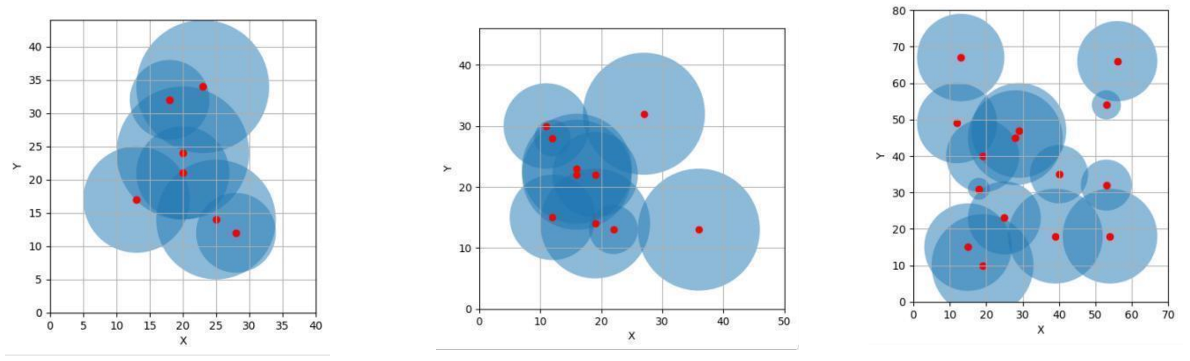


Fig 20: Random Coordinates Generation

Table 2 : Difference between using same radius and different range of radii

Feature	Same Radius for All Sensors	Radius Tuning According to Sensor Placeme nt
Optimization	Limited optimization potential as the same radius is applied to all se nsors, ignoring variations in cover age requirements.	Enhanced optimization potential as the radius is tuned based on sensor placement, maximizing c overage 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 basedon 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 Generatio n	Random coordinate generation remains the same for all sensors, resulting in static placement patterns.	Random coordinate generation may varyforeach sensor, resulting in dynamic placementpatterns tailored to individual coverage requirements.
Optimizati onEffort	Requires less effort in terms of parameter tuning and optimizationstrategy due to uniform radius application.	Requires additional effort in parameter tuning and optimization strategy to ensure effective radius tuning based on sensor placement.
Coverag e Flexibilit y	Less flexible in adapting to diversecoverage requirements or varying deployment scenarios.	More flexible in adapting to diverse coveragerequirements or varying deployment scenarios,leading to customizable coverage patterns.
Computation alComplexity	Lower computational complexity due to uniform radius application.	Higher computational complexity due to theneed for radius tuning based on sensorplacement.

The PSO algorithm leverages the collective intelligence and adaptive behavior observed in natural phenomena like fish schools or flocks of birds to efficiently explore and exploit the search space for optimal sensor placements. By mimicking the collaborative behaviors of these natural systems, the algorithm enables particles, representing potential sensor locations, to communicate and share information, fostering a cooperative search for the best solution. This collective exploration allows the algorithm to navigate complex solution landscapes effectively, dynamically adjusting sensor placements to adapt to changing environmental conditions or objectives. Moreover, the ability of the PSO algorithm to dynamically optimize sensor placements based on real-time feedback enhances its versatility and applicability across various scenarios and problem domains.

3.3 OPTIMIZING CODE EFFICIENCY AND PERFORMANCE IN PSO APPROACH :

To optimize the given code, you can employ several techniques aimed at improving its efficiency and performance. One approach is to focus on algorithmic optimizations, such as reducing redundant computations, minimizing memory usage, and improving data structures. This involves carefully analyzing the code to identify potential bottlenecks and areas for improvement. Additionally, you can consider parallelizing computationally intensive tasks to leverage the power of multicore processors and accelerate processing.

Another optimization strategy involves optimizing the code for specific hardware architectures and compiler optimizations. This may include using specialized libraries and functions optimized for the target platform, enabling compiler optimizations flags, and leveraging hardware acceleration features like SIMD (Single Instruction, Multiple Data) instructions.

To achieve optimization through the Particle Swarm Optimization (PSO) algorithm, you can apply it to fine-tune the parameters of the code or to optimize its overall performance. PSO is a metaheuristic optimization technique inspired by the social behavior of bird flocking or fish schooling. It iteratively adjusts a population of candidate solutions (particles) based on their individual and collective performance, aiming to converge towards optimal solutions.In the context of code optimization, PSO can be used to optimize parameters such as algorithmic constants, numerical thresholds, and control parameters affecting the code's behavior. By treating these parameters as dimensions in the search space, PSO iteratively explores and adjusts them to minimize a fitness function representing the code's performance metrics, such as execution time or memory usage.

To apply PSO for code optimization, you would typically define a fitness function that evaluates the performance of the code based on its parameters. This function could measure aspects such as execution time, resource utilization, or output quality, depending on the optimization goals. Then, you would initialize a population of particles representing different parameter configurations, update their positions and velocities iteratively based on their performance, and iteratively refine the population until convergence to optimal or near-optimal solutions.

Algorithm :

```
Initialize a population of particles randomly within the search space
For each particle:
  Initialize particle's position and velocity randomly
  Evaluate the fitness of the particle's current position
  Update the personal best position of the particle

While termination condition is not met:
  For each particle:
    Update the velocity of the particle using formula:
      velocity = inertia * velocity
            + c1 * random(0,1) * (personal_best_position - current_position)
            + c2 * random(0,1) * (global_best_position - current_position)
    Update the position of the particle using formula:
      position = position + velocity
    Evaluate the fitness of the new position
    Update the personal best position if necessary
    Update the global best position if necessary

Return the best solution found
```

3.4 ENHANCING SENSOR NETWORK EFFICIENCY THROUGH PSO-BASED OPTIMIZATION:

Particle Swarm Optimization (PSO) is a powerful technique for optimizing sensor networks, offering a robust and efficient approach to enhancing sensor coverage and resource allocation. Inspired by the collective behavior of organisms like bird flocks and fish schools, PSO enables sensors to dynamically adjust their positions within a given area to maximize coverage, minimize redundancy, and conserve energy. Operating iteratively, PSO updates a swarm of candidate solutions, or particles, representing potential sensor locations, guided by local and global information exchange.

This iterative refinement process gradually converges towards an optimal configuration that balances exploration and exploitation, effectively exploring the solution space while responding to changes in the environment or task requirements. PSO's versatility lies in its ability to handle complex, nonlinear optimization problems with large solution spaces, accommodating diverse objectives such as maximizing coverage area, minimizing energy consumption, or optimizing network connectivity. Overall, PSO serves as a valuable optimization tool for sensor networks, enhancing performance across various application domains, including environmental monitoring, surveillance, industrial automation, and smart cities.Overall, optimizing sensor networks through PSO involves several key steps.

First, defining the problem parameters such as sensor range, coverage area, and network constraints is essential for formulating the optimization task. Next, setting up the PSO parameters, including the number of particles, maximum iterations, and inertia weight factors, influences the convergence behavior and exploration-exploitation trade-off of the algorithm. Initialization of particle positions and velocities within predefined bounds is critical for starting the optimization process. During each iteration, PSO evaluates the fitness of particles based on the coverage achieved by sensor configurations, updating the local best-known positions (Pbest) and the global best-known position (Gbest) accordingly. and the global best-known position (Gbest) accordingly. The cognitive and social parameters regulate the balance between particle movement towards individual and collective best solutions, facilitating effective information .sharing and decision-making within the swarm. Additionally, the inertia weight factor controls the trade-off between exploration (high inertia) and exploitation (low inertia) throughout the optimization process, adapting to

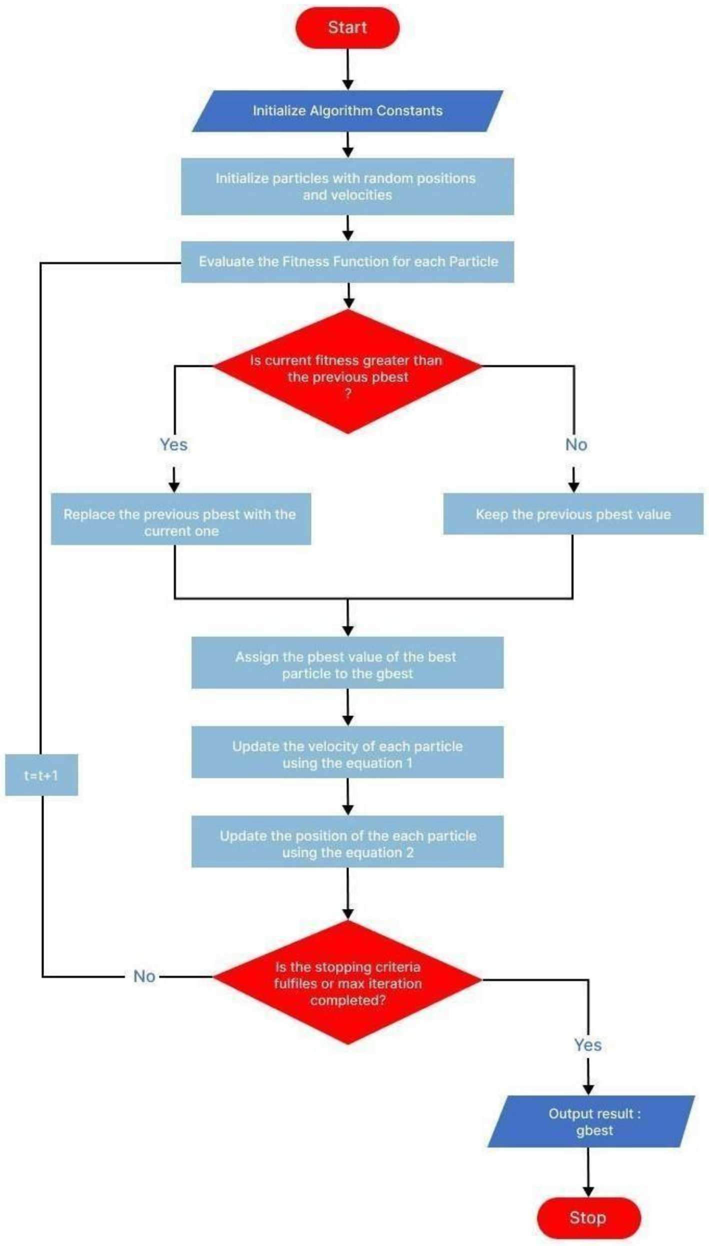

Start
Initialize Algorithm Constants
Initialize particles with random positions and velocities
Evaluate the Fitness Function for each Particle
Is current fitness greater than the previous pbest ?
Yes
No
Replace the previous pbest with the current one
Keep the previous pbest value
Assign the pbest value of the best particle to the gbest
Update the velocity of each particle using the equation 1
Update the position of the each particle using the equation 2
t=t+1
Is the stopping criteria fulfiles or max iteration completed?
No
Yes
Output result : gbest
Stop

Fig 21: PSO Algorithm Flow Chart

the problem's characteristics and convergence dynamics. Finally, evaluating the performance of the optimized sensor network in terms of coverage efficiency, energy consumption, communication overhead, or other relevant metrics provides insights into the effectiveness of the PSO-based optimization approach. Through its iterative, population-based optimization framework. Finally, evaluating the performance of the optimized sensor network in terms of coverage efficiency, energy consumption, communication overhead, or other relevant metrics provides insights into the effectiveness of the PSO-based optimization approach.

CHAPTER-4
RESULTS AND ANALYSIS

4.1 PSO-BASED OPTIMIZATION FOR SENSOR COVERAGE:

This Python program implements a Particle Swarm Optimization (PSO) algorithm to find optimal sensor positions within a specified area, aiming to maximize coverage. It begins by setting problem parameters such as the dimensions of the area, population size, maximum number of iterations, and number of dimensions. PSO-specific parameters like inertia weight factor, cognition, and social parameters are also defined. Bounds for the coordinates are initialized, along with placeholders for the best-known position and cost.

Particle velocities are initialized randomly, and initial positions are generated within the specified bounds. The PSO loop iterates over the defined number of iterations, evaluating the fitness of each particle based on its coverage area. The global best position and cost are updated if a particle's cost surpasses the current global best. Velocities are updated using the PSO equations, adjusting based on the inertia weight, cognitive, and social parameters, as well as random factors. Positions are then updated accordingly and clipped to stay within bounds.

After the loop, the optimal coordinates and maximum coverage area are outputted. Finally, the resultant sensor positions are plotted using Matplotlib, visualizing the distribution of particles within the area. This program demonstrates the application of PSO in optimizing sensor placement for maximizing coverage, offering insights into how metaheuristic algorithms can be utilized in solving optimization problems.The PSO algorithm iteratively refines the positions of particles by simulating social behavior and individual cognition, gradually converging towards an optimal solution. This approach is particularly useful for optimization problems where the search space is complex or high-dimensional, as it efficiently explores the solution space without requiring gradient information. In this specific context, the program aims to find the best locations for sensors within the defined area to maximize coverage, which is crucial in various applications such as surveillance, environmental monitoring, or network coverage optimization.

By leveraging PSO, the program effectively balances exploration and exploitation, enabling it to escape local optima and converge towards better solutions. Additionally, the program demonstrates the flexibility and scalability of PSO, as it can be easily adapted to different problem domains by adjusting parameters or objective functions. Overall, this program showcases the power and versatility of metaheuristic optimization techniques like PSO in solving real-world optimization challenges, providing a valuable tool for researchers and practitioners in diverse fields.

The PSO loop iterates over the specified number of iterations, evaluating the fitness for each particle based on its coverage area. It updates the global best position and cost if a better solution is found. Velocities are updated using the inertia weight factor and random values, adjusting the particles' movement towards the global best position while considering both cognition and social influence.

Positions are then updated based on the calculated velocities, and a clipping mechanism ensures that the particles stay within the defined bounds. After the loop completes, the program outputs the optimal coordinates and the maximum coverage area achieved.Overall, this program showcases the power and versatility of metaheuristic optimization techniques like PSO in solving real-world optimization challenges, providing a valuable tool for researchers and practitioners in diverse fields. Finally, the program visualizes the resultant coordinates using matplotlib, plotting the particles' positions and highlighting the optimal position if desired. This visualization provides insight into the distribution of sensors within the area and facilitates further analysis or decision-making. Overall, the program seamlessly integrates PSO optimization with visualization, offering a comprehensive solution for sensor placement optimization tasks.

Particle Swarm Optimization Algorithm :

Algorithm: Particle Swarm Optimization (PSO)

Inputs:

- Problem parameters (length, breadth, pop, itermax, npar)
- PSO parameters (Wmax, Wmin, c1, c2)
- Bounds for coordinates (xmin, xmax)

Outputs:

- Optimal coordinates (global_best_position)
- Maximum coverage area (global_best_cost)

```
1. Initialize Parameters:
  - Set length, breadth, pop, itermax, npar
  - Set Wmax, Wmin, c1, c2
  - Define bounds (xmin, xmax)

2. Initialize Variables:
  - Initialize global_best_position = [0, 0]
  - Initialize global_best_cost = -infinity
  - Initialize vel[pop][npar] = 0
  - Generate random initial positions within bounds: positions[pop][npar]

3. PSO Loop:
  for iteration = 1 to itermax do
    a. Evaluate Fitness:
      - Compute cost for each particle: costs[pop]
        cost = length * breadth - product of positions
    b. Update Global Best:
      - Find index of particle with maximum cost: best_index = argmax(costs)
      - If cost[best_index] > global_best_cost:
          global_best_position = positions[best_index]
          global_best_cost = costs[best_index]
    c. Update Velocities:
      - Calculate inertia weight: W = Wmax - (Wmax - Wmin) * iteration / itermax
      - Generate random numbers r1, r2 [pop][npar]
      - Update velocities: velupdate = W * vel + c1 * r1 * (local_best_position -
        positions) + c2 * r2 * (global_best_position - positions)
      - Update vel = velupdate
    d. Update Positions:
      - Update positions: positions += vel
      - Clip positions to stay within bounds: positions = clip(positions, xmin, xmax)
  end for
```

4. Output Results:

- Print optimal coordinates: global_best_position
- Print maximum coverage area: global_best_cost

5. Visualization (Optional):

- Plot the resultant sensor positions
- Scatter plot of positions
- Optional: Highlight optimal position

4.2 COMPREHENSIVE OVERVIEW OF PARTICLE SWARM OPTIMIZATION ALGORITHM:

◆ **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.

◆ **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)
# Initialization of best-known positions and costs
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), :]
globalfit = 1 / (1 + globalcost)
```

- **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.

```
velupdate = W * vel + (c1 * r1 * (Pbest - pos)) + (c2 * r2 * (Gbest - pos))
```

- **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.

```
cost[iter-1, pp] = (l * b) - (np.pi * x1 * x1 + np.pi * x2 * x2)
fit[pp] = 1 / (1 + cost[iter-1, pp])
```

- **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.

```
Gbest = Pbest[pp, :]
    globalcost = localcost[pp]
    globalfit = localfit[pp]
```

4.3 RESULTS AND ANALYSIS OF PSO :

The Particle Swarm Optimization (PSO) technique serves as a powerful method for generating sensor coordinates aimed at identifying optimal locations within a given spatial domain. This iterative approach relies on the movement and interaction of particles, representing potential sensor sites, to converge towards the most favorable solutions. Initially, a population of particles is randomly distributed within the predefined boundaries of the sensor deployment area. Each particle is assigned coordinates (x, y) representing a potential sensor position within this region. As the algorithm progresses through successive iterations, the positions of these particles are dynamically updated based on both individual particle experiences and the collective behavior of the swarm. Throughout this process, a predefined objective function evaluates the fitness of each particle's location, guiding the optimization process towards configurations that maximize desired criteria such as coverage or efficiency. Through iterative refinement, the PSO algorithm gradually converges towards optimal sensor placements, leveraging the collective intelligence of the particle swarm to navigate the solution space and identify high-quality solutions for sensor deployment.

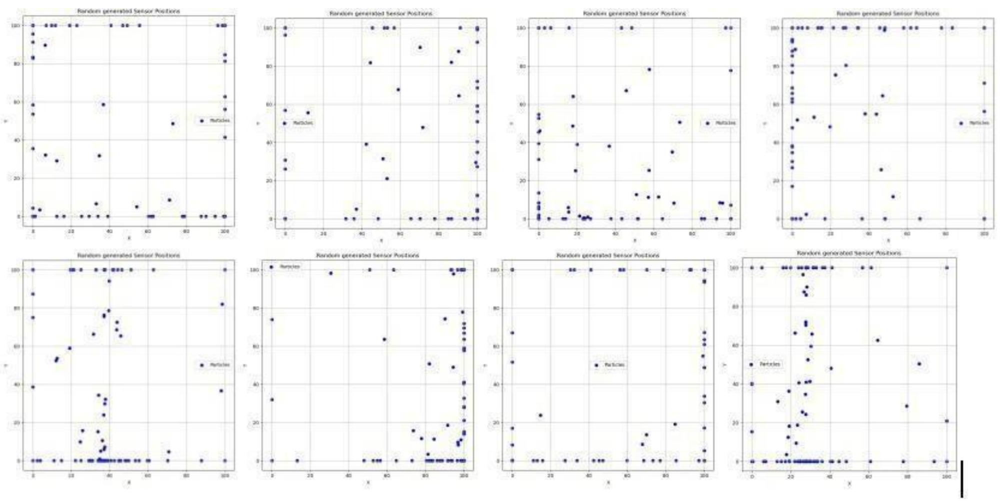


Fig 22: Generating Random Sensor Coordinates Through PSO

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 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.

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.

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.Furthermore, the graphical representation of node placements facilitates intuitive assessment of the deployed network's efficacy. This visual aid enables users to identify potential coverage gaps and optimize sensor positions accordingly, ensuring comprehensive monitoring across the designated area. By leveraging this visual feedback loop, users can fine-tune sensor placements to achieve optimal coverage and maximize the efficiency of the sensor network deployment.

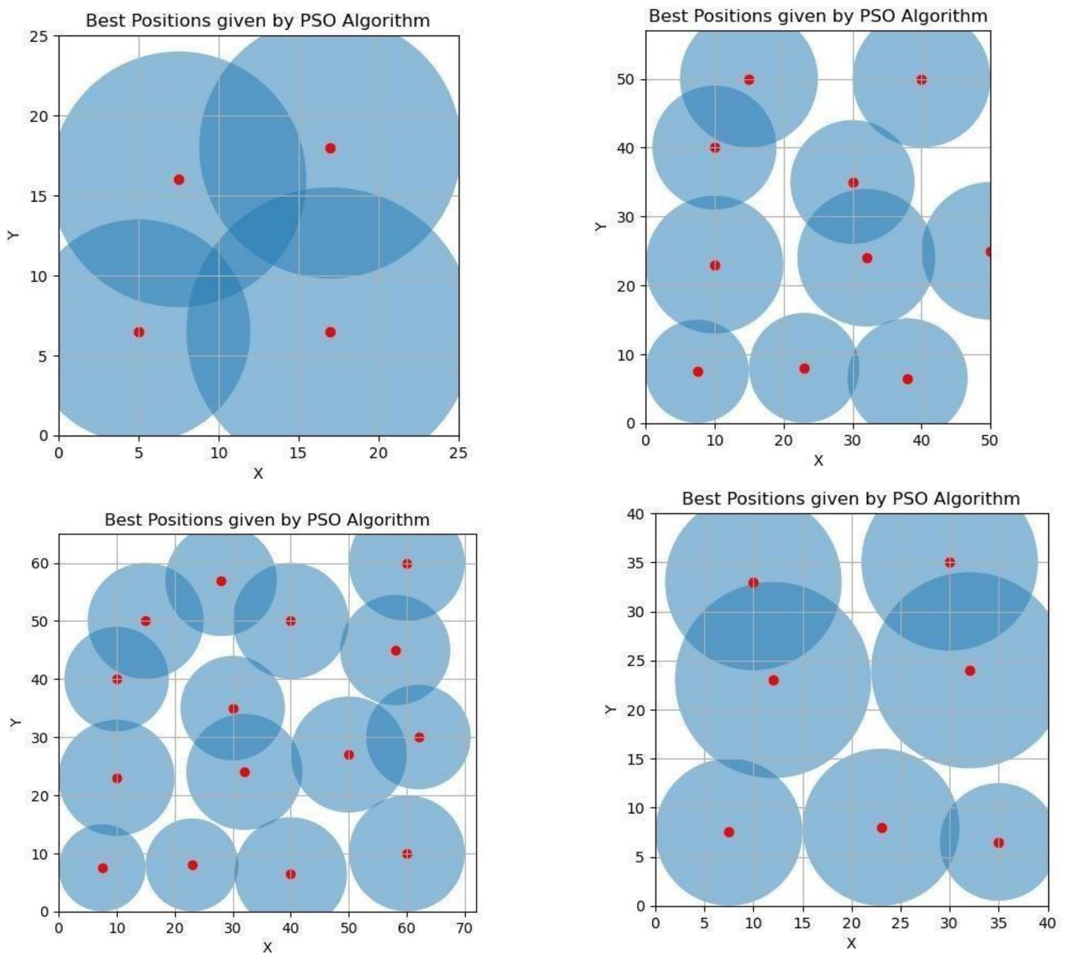


Fig 23: Best Sensor Positions Given by PSO

Case-1 :

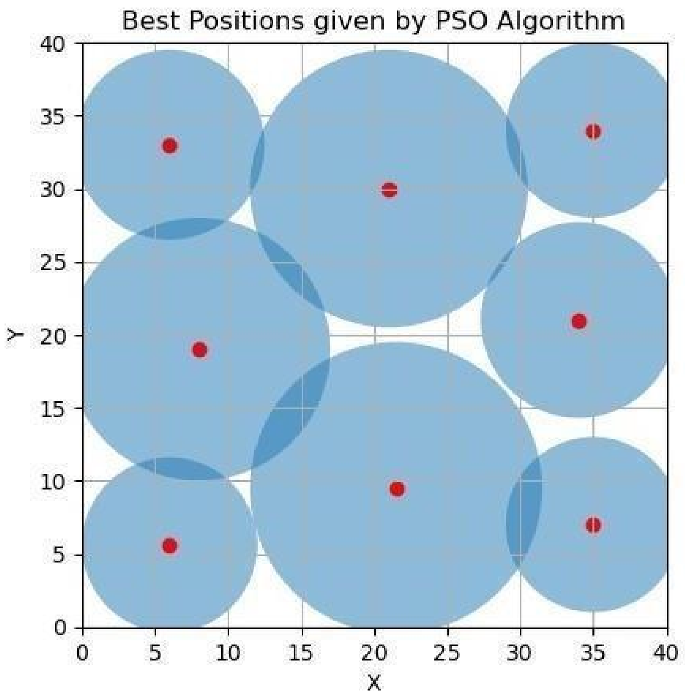


Fig 24: Best Sensor Positions Given by PSO for 8 Sensors

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

Case-2 :

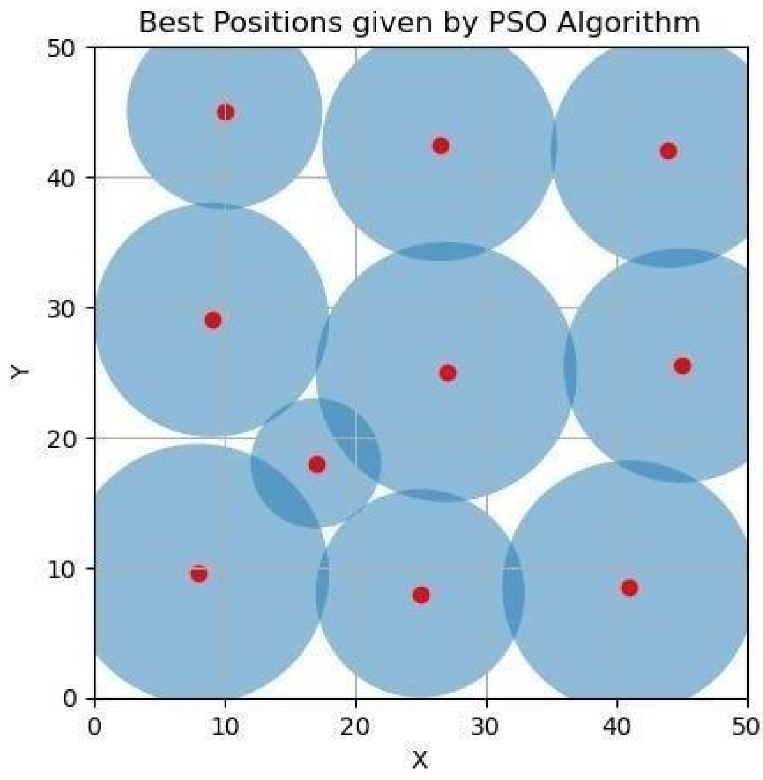


Fig 25: Best Sensor Positions Given by PSO for 10 Sensors

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

Case-3 :

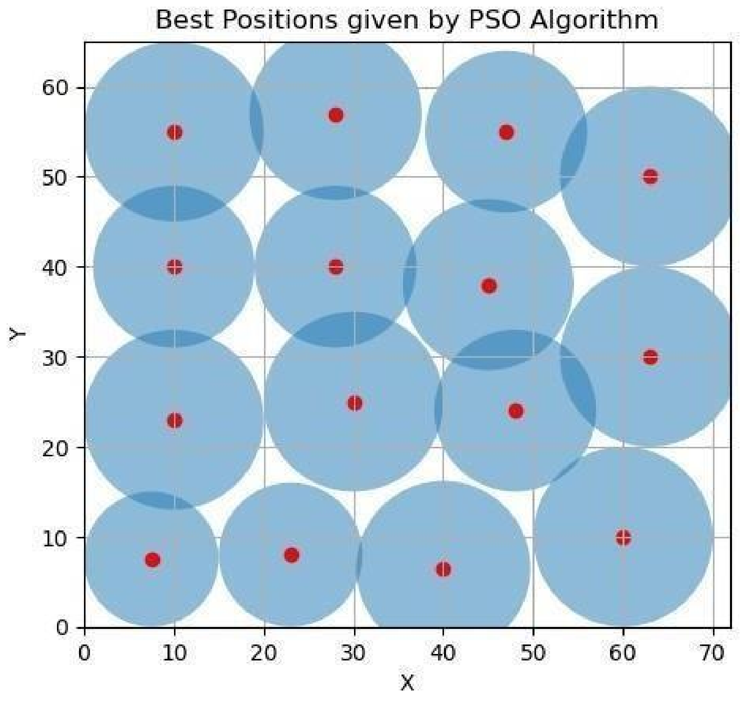


Fig 26: Best Sensor Positions Given by PSO for 15 Sensors

No.of Sensors:15
Coordinates of sensors:
Sensor 1: (7.5, 7) ,7.5
Sensor 2: (23, 8) ,8
Sensor 3: (40, 6.5) ,9.7
Sensor 4: (30, 25) ,10
Sensor 5: (10, 23) ,10
Sensor 6: (28, 40) ,9
Sensor 7: (10, 40) ,9
Sensor 8: (10, 55) ,10
Sensor 9: (47, 55) ,9
Sensor 10: (48, 24) ,9
Sensor 11: (60, 10) ,10
Sensor 12: (45, 38) ,9.5
Sensor 13: (63, 30) ,10
Sensor 14: (63, 50) ,10
Sensor 15: (28, 57) ,9.6
Total area of the region: 4680
Total area covered by circles: 4403.7654
Total area uncovered by circles: 276.2346

Table3: Best sensor Positions Given by PSO

No.of Sensors	**Total area of the region**	**Total area covered**	**Total area uncovered**
8	1600	1465.2073977077434	134.79260229225
10	2500	2398.103361912328	101.89666380876
15	4680	4403.7654	276.2346

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. By simulating the behavior of these particles, PSO effectively explores the solution space, enabling the identification of optimal sensor positions for maximum coverage. This iterative process allows the algorithm to adapt and refine its search over time, gradually converging towards the most suitable solution. Additionally, the program's integration of PSO not only streamlines the optimization process but also enhances scalability and adaptability, making it well-suited for various scenarios and environments.

CHAPTER -5
CONCLUSIONS AND FUTURE SCOPE

5.1 CONCLUSION:

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.

5.2 FUTURE SCOPE:

The future scope of this project encompasses several avenues for further research, development, and application in the field of sensor network optimization. One potential direction involves exploring hybrid optimization approaches that combine the strengths of multiple algorithms, such as Particle Swarm Optimization (PSO), Genetic Algorithms (GA), and Ant Colony Optimization (ACO), to achieve superior performance in terms of coverage, energy efficiency, and scalability.

Additionally, integrating machine learning techniques, such as deep learning and reinforcement learning, into sensor network optimization frameworks holds promise for enhancing adaptability and self-optimization capabilities in dynamic environments. Another area for future exploration is the development of decentralized optimization algorithms that enable autonomous decision-making and coordination among sensor nodes, leading to more robust and resilient network architectures.

Furthermore, extending the scope of optimization beyond traditional sensor networks to include emerging technologies such as Internet of Things (IoT), Cyber-Physical Systems (CPS), and Edge Computing opens up new possibilities for enhancing real-time data analytics, predictive maintenance, and situational awareness in diverse application domains. Lastly, there is a need for comprehensive validation and benchmarking frameworks to evaluate the performance of optimization algorithms under various scenarios and conditions, facilitating the comparison of different approaches and guiding the selection of appropriate techniques for specific use cases.

REFERNCES

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.l

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.

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

IJIRSET

International Journal of Innovative Research in SCIENCE | ENGINEERING | TECHNOLOGY

INTERNATIONAL JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

Volume 13, Issue 3, March 2024

INTERNATIONAL STANDARD SERIAL NUMBER INDIA

Impact Factor: 8.423

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

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 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

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

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

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

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.

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

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

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

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 sensor s, ignoring variations in coverage requ irements.	Enhanced optimization potential as the radius is tune d based on sensor placement, maximizing coverage a nd minimizing redundancy.
Area Coverage	May result in uneven coverage, with s ome areas having excessive overlap and others under covered.	Provides more uniform coverage across the deployment area by adjusting the radius based on th e 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 :

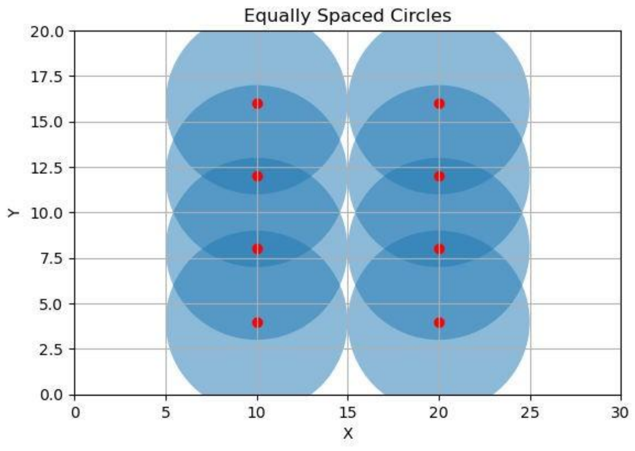


Fig 1: Example coverage of sensors with equal radius

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: 600m^2
Total area covered by circles: 628.3185307179587m^2
Total area uncovered by circles: -28.31853071795865m^2

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.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

Case 2 :

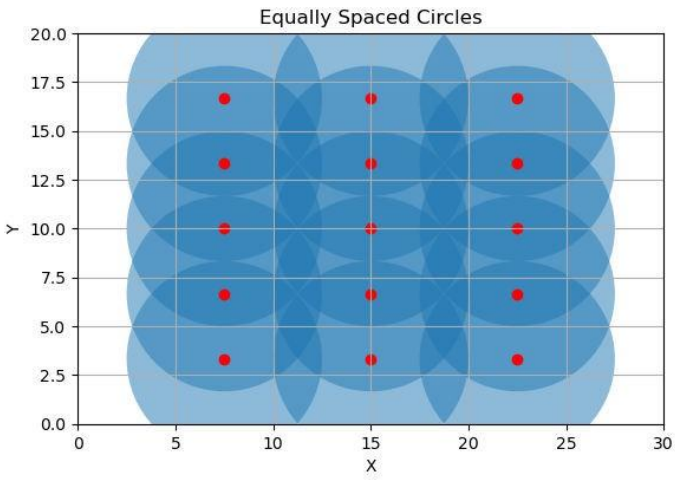


Fig 2: Example coverage of sensors with equal radius

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.

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

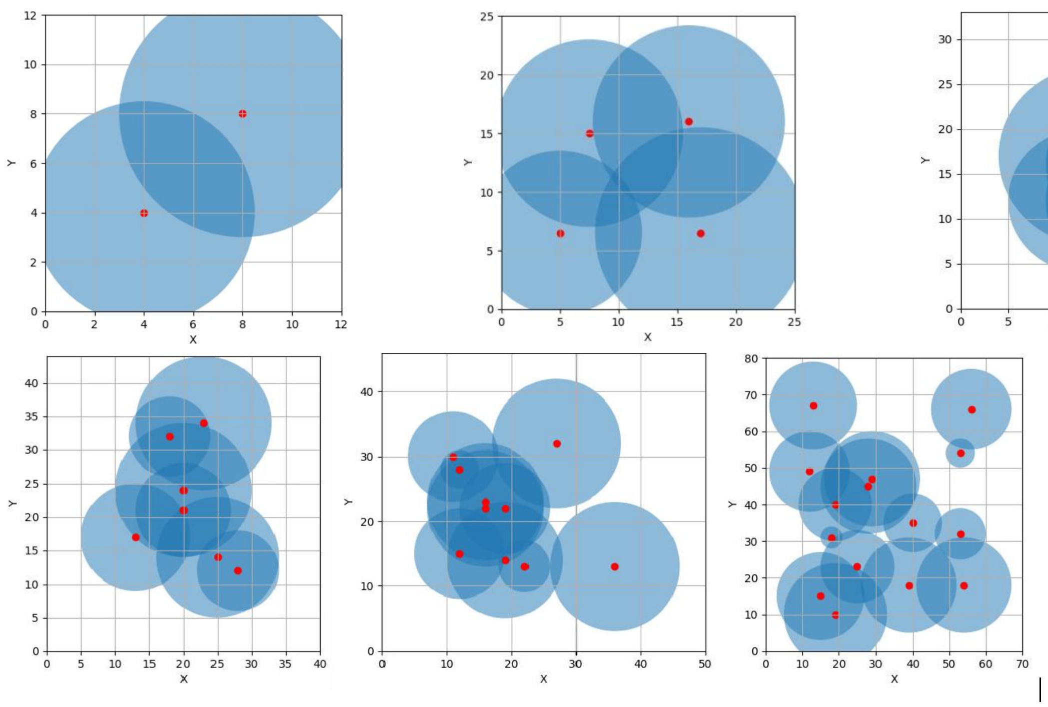


Fig 3: Random Coordinates and radius generated by Manual Software

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), :]
```

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

globalfit = 1 / (1 + 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.
velupdate = W * vel + (c1 * r1 * (Pbest - pos)) + (c2 * r2 * (Gbest - 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.

```
cost[iter-1, pp] = (l * b) - (np.pi * x1 * x1 + np.pi * x2 * x2)
            fit[pp] = 1 / (1 + cost[iter-1, 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.

```
  Gbest = Pbest[pp, :]
                        globalcost = localcost[pp]
                        globalfit = localfit[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.

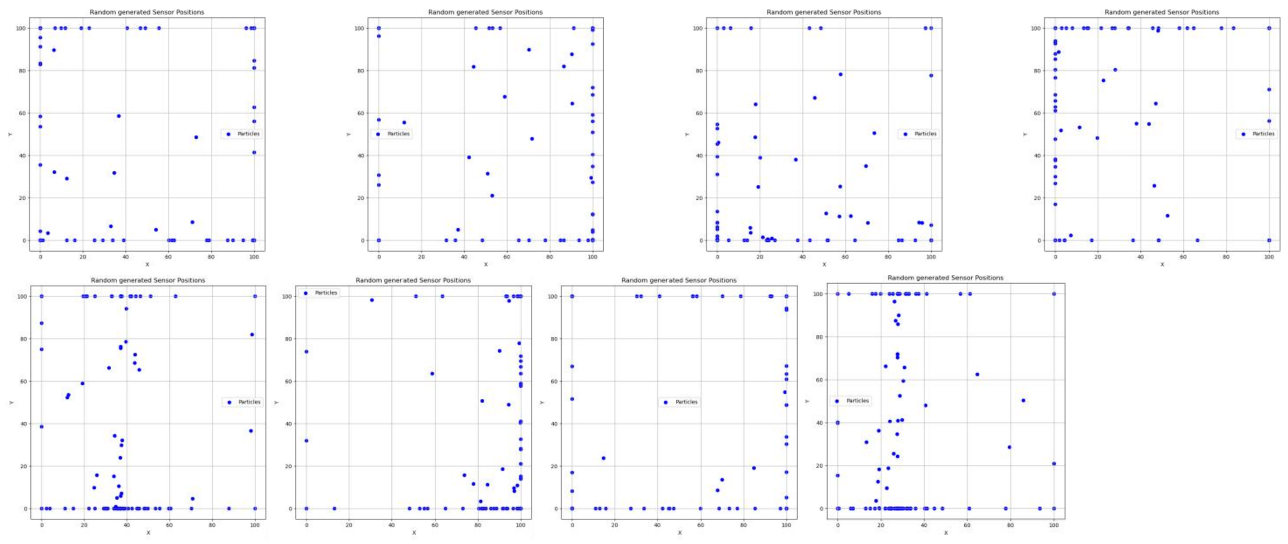


Fig 4 : Random Co-ordinates generation over every iteration by PSO Algorithm

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.

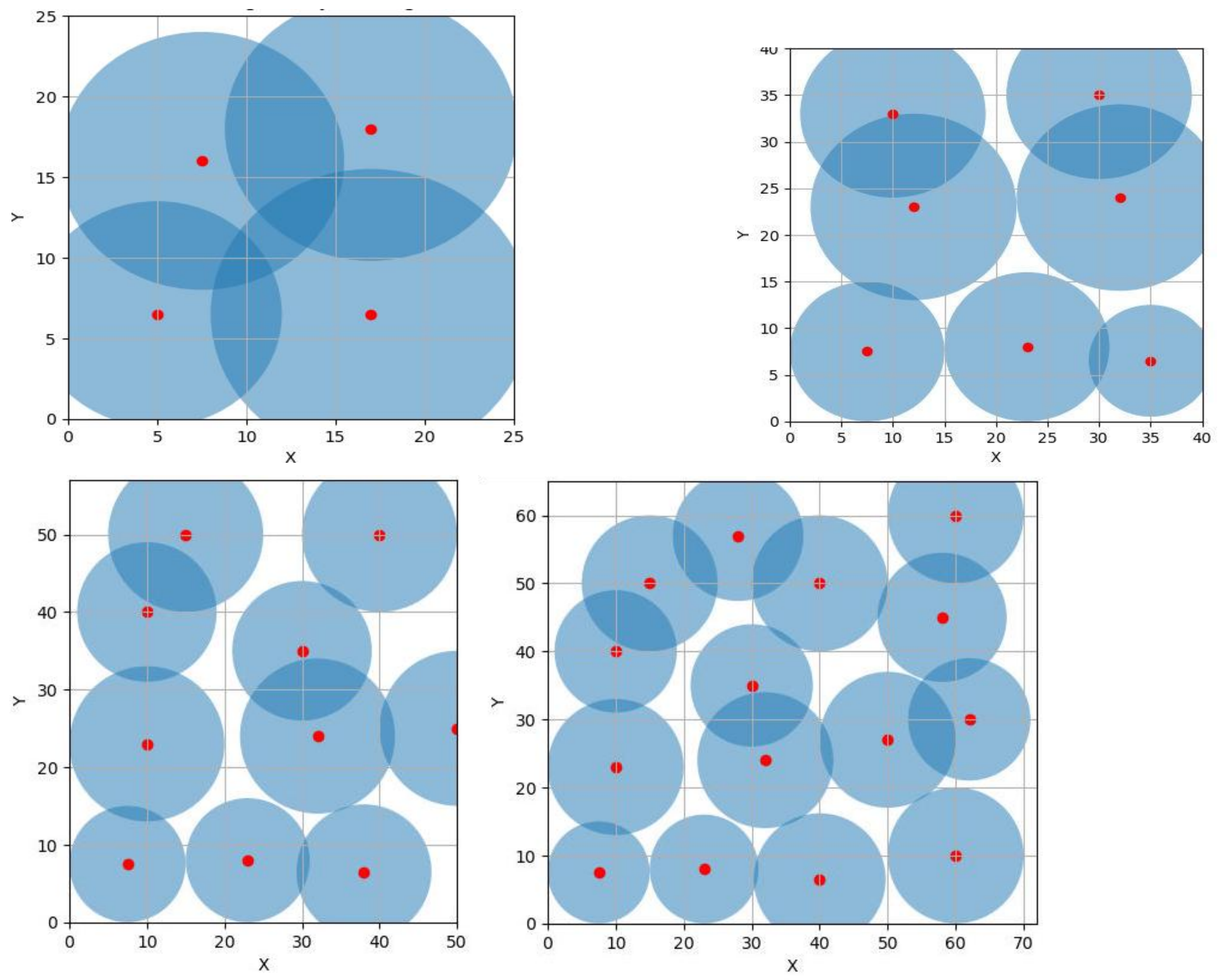


Fig 5: Generation and plotting of random Co-ordinates and radius by PSO Algorithm

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

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 :

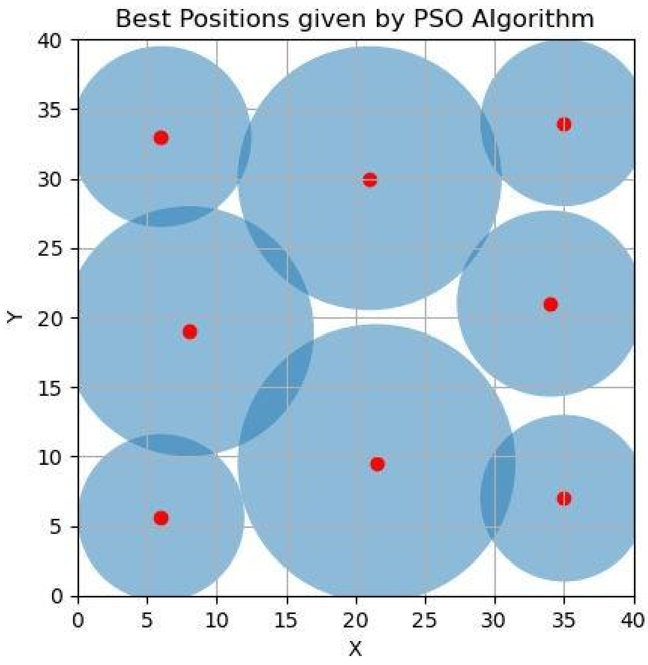


Fig 6: PSO generated Optimized positions and radius

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 m^2

Total area covered by circles: 1465.2073977077434 m^2

Total area uncovered by circles: 134.79260229225656 m^2

Case 2 :

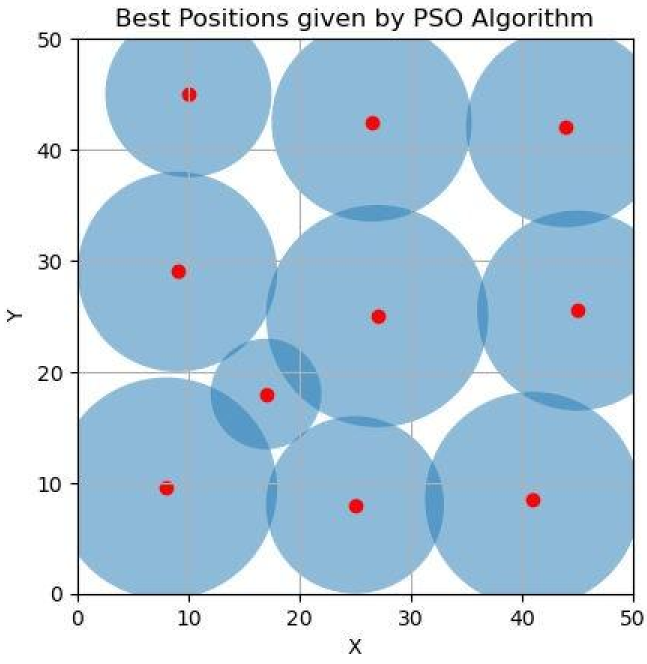


Fig 7: PSO generated Optimized positions and radius

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 m^2

Total area covered by circles: 2398.1033361912328 m^2

Total area uncovered by circles: 101.89666380876724 m^2

International Journal of Innovative Research in Science, Engineering and Technology (IJIRSET)

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

Case 3 :

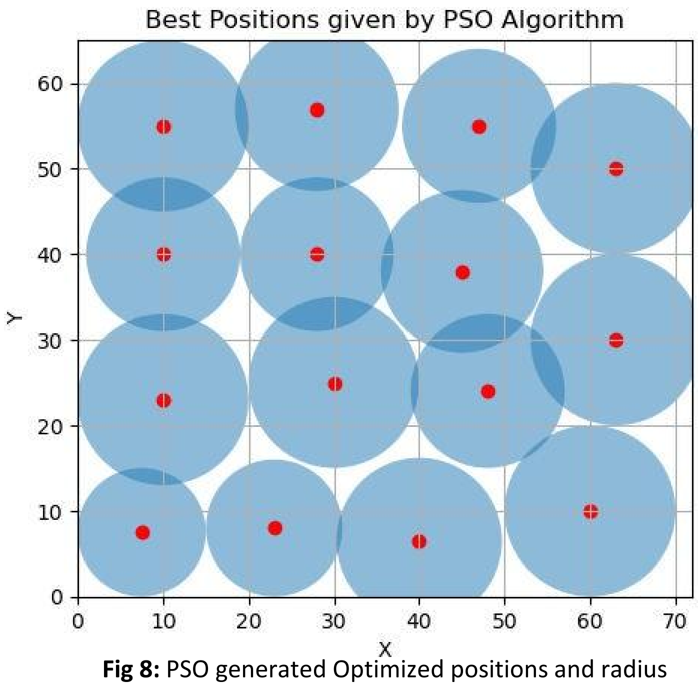


Fig 8: PSO generated Optimized positions and radius

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 m^2

Total area covered by circles: 4403.7654 m^2

Total area uncovered by circles: 276.2346 m^2

Population Vs Iterations :

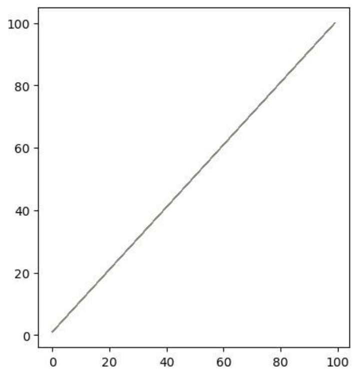
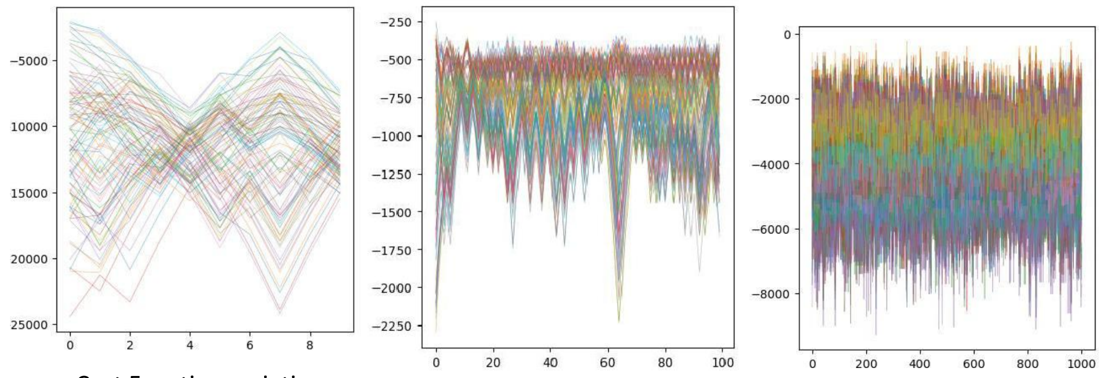


Fig 9: Graph between Population and iteration in PSO

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.



Cost Function variation over 10 Iterations

Cost Function variation over 100 Iterations

Cost Function variation over 1000 Iterations

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.l
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

|e-ISSN: 2319-8753, p-ISSN: 2347-6710| www.ijirset.com | Impact Factor: 8.423| A Monthly Peer Reviewed & Referred Journal |

|| Volume 13, Issue 3, March 2024 ||

| DOI:10.15680/IJIRSET.2024.1303172|

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 JOURNAL OF INNOVATIVE RESEARCH

IN SCIENCE | ENGINEERING | TECHNOLOGY

p1

ORIGINALITY REPORT

26%	20%	19%	16%
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.nitte.edu.in Internet Source	2%
2	www.geeksforgeeks.org Internet Source	2%
3	www.coursehero.com Internet Source	1%
4	kharinee25.medium.com Internet Source	1%
5	Submitted to Liverpool Hope Student Paper	1%
6	www.mdpi.com Internet Source	1%
7	Ton Duc Thang University Publication	<1%
8	link.springer.com Internet Source	<1%
9	www.bitswgl.ac.in Internet Source	<1%

10	mdpi-res.com Internet Source	<1%
11	Submitted to National School of Business Management NSBM, Sri Lanka Student Paper	<1%
12	Ojonukpe S. Egwuche, Abhilash Singh, Absalom E. Ezugwu, Japie Greeff, Micheal O. Olusanya, Laith Abualigah. "Machine learning for coverage optimization in wireless sensor networks: a comprehensive review", Annals of Operations Research, 2023 Publication	<1%
13	Submitted to University of Teesside Student Paper	<1%
14	Submitted to Kingston University Student Paper	<1%
15	Submitted to University of Bolton Student Paper	<1%
16	Thi-Kien Dao, Trong-The Nguyen, Truong-Giang Ngo, Trinh-Dong Nguyen. "An optimal wsn coverage based on adapted transit search algorithm", International Journal of Software Engineering and Knowledge Engineering, 2023 Publication	<1%

17	Naufal Ammarfaizal, Aji Gautama Putrada, Maman Abdurohman. "A Cluster Head Selection Method Comparison of DCHSM, DEEC, and LEACH on Wireless Sensor Network Using Voronoi Diagram", 2021 International Conference Advancement in Data Science, E-learning and Information Systems (ICADEIS), 2021 Publication	<1%
18	ir.kluniversity.in Internet Source	<1%
19	Kübra Tümay Ateş. "Estimation of Short-Term Power of Wind Turbines Using Artificial Neural Network (ANN) and Swarm Intelligence", Sustainability, 2023 Publication	<1%
20	ir.juit.ac.in:8080 Internet Source	<1%
	aece.ro Internet Source	<1%
	rt.ok-em.com Internet Source	<1%
	eprints.utem.edu.my Internet Source	<1%
	ictactjournals.in Internet Source	<1%

25	vvitguntur.com Internet Source	<1%
26	cse.anits.edu.in Internet Source	<1%
27	Bheema Shanker Neyigapula. "Designing a Robust and Efficient Routing Protocol for Wireless Sensor Networks", Research Square Platform LLC, 2023 Publication	<1%
28	Submitted to University of Northumbria at Newcastle Student Paper	<1%
29	shodhganga.inflibnet.ac.in Internet Source	<1%
30	www.researchgate.net Internet Source	<1%
31	Zhenbao Wang, Shuyue Liu, Yuchen Zhang, Xin Gong, Shihao Li, Dong Liu, Ning Chen. "Exploring the Relative Importance and Interactive Impacts of Explanatory Variables of the Built Environment on Ride-Hailing Ridership by Using the Optimal Parameter-Based Geographical Detector (OPGD) Model", Applied Sciences, 2023 Publication	<1%
32	Submitted to Asian Institute of Technology Student Paper	

		<1%
33	thesai.org Internet Source	<1%
34	www.ajol.info Internet Source	<1%
35	Submitted to University of Greenwich Student Paper	<1%
36	Xueqin Liu, Mingzhe Jin. "Classification analysis of Kouji Uno’s novels using topic model", Behaviormetrika, 2019 Publication	<1%
37	ouci.dntb.gov.ua Internet Source	<1%
38	Submitted to RMIT University Student Paper	<1%
39	arxiv.org Internet Source	<1%
40	worldwidescience.org Internet Source	<1%
41	www.hindawi.com Internet Source	<1%
42	cheatography.com Internet Source	<1%

43	Akram Sheikhi, Maryam Bazgir, Mohammad Bagher Dowlatshahi. "Chapter 54-1 Optimization and Machine Learning Algorithms for Intelligent Microwave Sensing: A Review", Springer Science and Business Media LLC, 2024 Publication	<1 %
44	Submitted to ESC Rennes Student Paper	<1 %
45	Jyoti A. Dhanke, T. Jayapratha, K. Kannan, Sampada Gulavani, A. Karthikeyan, P. Anitha Christy Angelin. "Improved K-Means Clustering Algorithm with Wireless Networks to Increase the Production in the Automobile Industry", 2022 IEEE North Karnataka Subsection Flagship International Conference (NKCon), 2022 Publication	<1 %
46	erepo.uef.fi Internet Source	<1 %
47	sciendo.com Internet Source	<1 %
48	Dakun Yu, Zhongwei Xu, Meng Mei. "Multi-objective Task Scheduling Optimization Based on Improved Bat Algorithm in Cloud Computing Environment", International	<1 %

	Journal of Advanced Computer Science and Applications, 2023 Publication	
49	vitap.ac.in Internet Source	<1%
50	Submitted to Indian Institute of Technology, Madras Student Paper	<1%
51	Submitted to University of Westminster Student Paper	<1%
52	coek.info Internet Source	<1%
53	d197for5662m48.cloudfront.net Internet Source	<1%
54	oa.upm.es Internet Source	<1%
55	Anand Sivasubramaniam. "PENS", Proceedings of the 1st international conference on Scalable information systems - InfoScale 06 InfoScale 06, 2006 Publication	<1%
56	Lecture Notes in Computer Science, 2012. Publication	<1%
57	Submitted to University of Hong Kong Student Paper	<1%

58	Submitted to University of Sheffield Student Paper	<1%
59	Submitted to University of Strathclyde Student Paper	<1%
60	tudr.thapar.edu:8080 Internet Source	<1%
61	www.slideshare.net Internet Source	<1%
62	Submitted to Universidad EAFIT Student Paper	<1%
63	assets.researchsquare.com Internet Source	<1%
64	pdfcookie.com Internet Source	<1%
65	vdoc.pub Internet Source	<1%
66	123dok.net Internet Source	<1%
67	Submitted to Institute of Aeronautical Engineering (IARE) Student Paper	<1%
68	M. Supriya, Amit Kumar Tyagi, Shrikant Tiwari, Richa. "chapter 8 Sensor-Based	<1%

Intelligent Recommender Systems for Agricultural Activities", IGI Global, 2024
Publication

69	Mohammad Riyaz Belgaum, Shahrulniza Musa, Fuead Ali, Muhammad Mansoor Alam et al. "Self-Socio Adaptive Reliable Particle Swarm Optimization Load Balancing in Software-Defined Networking", IEEE Access, 2023 Publication	<1%
70	sic.ici.ro Internet Source	<1%
71	www.al-kindipublisher.com Internet Source	<1%
72	Abbas Sheykhfard, Farshidreza Haghighi, Subasish Das, Grigorios Fountas. "Evasive actions to prevent pedestrian collisions in varying space/time contexts in diverse urban and non-urban areas", Accident Analysis & Prevention, 2023 Publication	<1%
73	Submitted to IPS Academy, Institute Of Engineering & Science ,Indore Student Paper	<1%
74	Jianwen Zhou, Wang Xinyu, Lei Chang, Adham E. Ragab. "Absorbed energy capacity, and dynamics of improved perovskite solar cells:	<1%

Introducing SVM-PSO-GA algorithm to predict vibrational information", Aerospace Science and Technology, 2024
Publication

---

75 S. Regilan, L.K. Hema. "Optimizing environmental monitoring in IoT: integrating DBSCAN with genetic algorithms for enhanced clustering", International Journal of Computers and Applications, 2023
Publication
<1%

---

76 Submitted to Webster University
Student Paper
<1%

---

77 Yogita Yashveer Raghav, Vaibhav Vyas. "Chapter 3 A Comparative Analysis Report of Nature-Inspired Algorithms for Load Balancing in Cloud Environment", Springer Science and Business Media LLC, 2024
Publication
<1%

---

78 Zhang Lin, Huan-Chao Keh, Ruikun Wu, Diptendu Sinha Roy. "Joint Data Collection and Fusion Using Mobile Sink in Heterogeneous Wireless Sensor Networks", IEEE Sensors Journal, 2020
Publication
<1%

---

79 iaeme.com
Internet Source
<1%

---

80 www.ijmsbr.com
Internet Source

<1%

81 Submitted to Guru Nanak Dev Engineering College
Student Paper
<1%

82 Qing Cao. "A grid-based clustering method for mining frequent trips from large-scale, event-based telematics datasets", 2009 IEEE International Conference on Systems Man and Cybernetics, 10/2009
Publication
<1%

83 Submitted to University of the Philippines Los Banos
Student Paper
<1%

84 Submitted to Xiamen University
Student Paper
<1%

85 ebin.pub
Internet Source
<1%

ijiemr.org
Internet Source
<1%

www.irjmets.com
Internet Source
<1%

88 Submitted to City University of Seattle
Student Paper
<1%

89	Submitted to Georgia Institute of Technology Main Campus Student Paper	<1 %
90	Suraj Kumar Prusty, Soumya P. Dash, V. K. Surya, Nijwm Wary. "Differential Evolution-Based Adaptation Algorithm for Multistage Continuous-Time Linear Equalizer", IEEE Transactions on Components, Packaging and Manufacturing Technology, 2023 Publication	<1 %
91	ijesc.org Internet Source	<1 %
92	livrepository.liverpool.ac.uk Internet Source	<1 %
93	psecommunity.org Internet Source	<1 %
94	www.igi-global.com Internet Source	<1 %
95	www2.mdpi.com Internet Source	<1 %
96	Submitted to Curtin University of Technology Student Paper	<1 %
97	Submitted to King's College Student Paper	<1 %

98	Submitted to Liverpool John Moores University Student Paper	<1%
99	Submitted to University of Birmingham Student Paper	<1%
100	Submitted to University of Salford Student Paper	<1%
101	ri.cmu.edu Internet Source	<1%
102	Submitted to College of Engineering & Technology Bhubaneswar Student Paper	<1%
103	Submitted to George Bush High School Student Paper	<1%
104	Submitted to Universiti Teknologi Petronas Student Paper	<1%
105	online-journals.org Internet Source	<1%
106	ses.library.usyd.edu.au Internet Source	<1%
107	www.econstor.eu Internet Source	<1%
108	Submitted to Jawaharlal Nehru Technological University Kakinada Student Paper	<1%

109	Shraddha Dattatray Jape, Kaveri Vitthal Mungase, Vaishnavi Bhausaheb Thite, Devyani Jadhav. "A Comprehensive Analysis on 5G, IoT and Its Impact on Agriculture and Healthcare", 2023 Second International Conference on Augmented Intelligence and Sustainable Systems (ICAISS), 2023 Publication	<1%
110	Submitted to University of Nottingham Student Paper	<1%
111	c.coek.info Internet Source	<1%
112	khazna.ku.ac.ae Internet Source	<1%
113	linknovate.com Internet Source	<1%
114	scholarcommons.scu.edu Internet Source	<1%
115	www.researchsquare.com Internet Source	<1%
116	www.univ-usto.dz Internet Source	<1%
117	Advances in Intelligent Systems and Computing, 2014. Publication	<1%

118 G. Venkataraman, S. Emmanuel, S. Thambipillai. "DASCA: A Degree and Size based Clustering Approach for Wireless Sensor Networks", 2005 2nd International Symposium on Wireless Communication Systems, 2005 <1%
Publication

119 Ghaihab Hassan Adday, Shamala K. Subramaniam, Zuriati Ahmad Zukarnain, Normalia Samian. "Investigating and Analyzing Simulation Tools of Wireless Sensor Networks: A Comprehensive Survey", IEEE Access, 2024 <1%
Publication

120 Jian-hua Huang, Zi-ming Zhao, Yu-bo Yuan, Ya-dong Hong. "Multi-factor and Distributed Clustering Routing Protocol in Wireless Sensor Networks", Wireless Personal Communications, 2017 <1%
Publication

O. Deepa, J. Suguna. "An optimized QoS-based clustering with multipath routing protocol for Wireless Sensor Networks", Journal of King Saud University - Computer and Information Sciences, 2017 <1%
Publication

Radha Debal Goswami, Sayan Chakraborty, Bitan Misra. "Chapter 1 Variants of Genetic <1%

Algorithms and Their Applications", Springer Science and Business Media LLC, 2023
Publication

Rohitashw Kumar, Muneeza Farooq, Mahrukh Qureshi. "Advancing precision agriculture through artificial intelligence", Elsevier BV, 2024 Publication	<1%
github.com Internet Source	<1%
publications.waset.org Internet Source	<1%
spie.org Internet Source	<1%
www.tnpesu.org Internet Source	<1%
Amutha, S., and S. Nivethalakshmi. "Solution for multicast routing problem using particle swarm optimization", 2015 International Conference on Computing and Communications Technologies (ICCCT), 2015. Publication	<1%
Ravi Mahesh Babu, Krishna Prasad Satamraju, Bellagubbala Neeharika Gangothri, Balakrishnan Malarkodi et al. "A HYBRID MODEL USING GENETIC ALGORITHM FOR ENERGY OPTIMIZATION IN HETEROGENEOUS	<1%

INTERNET OF BLOCKCHAIN THINGS", Telecommunications and Radio Engineering, 2024
Publication

Sadegh Afzal, Behrooz M. Ziapour, Afshar Shokri, Hamid Shakibi, Behnam Sobhani. "Building energy consumption prediction using multilayer perceptron neural network-assisted models; comparison of different optimization algorithms", Energy, 2023
Publication <1%

Sadeq Kord, Touraj Taghikhany, Ali Madadi, Omar Hosseinbor. "A novel triple-structure coding to use evolutionary algorithms for optimal sensor placement integrated with modal identification", Structural and Multidisciplinary Optimization, 2024
Publication <1%

---

132 Submitted to The University of Memphis
Student Paper <1%

---

133 Wu, . "Some Variants of Particle Swarm OptimiSation", Chapman & Hall/CRC Numerical Analy & Scient Comp Series, 2011.
Publication <1%

---

134 Z. Al-Hamouz. "APPLICATION OF PARTICLE SWARM OPTIMIZATION ALGORITHM FOR OPTIMAL REACTIVE POWER PLANNING", Control and Intelligent Systems, 2007 <1%

	Publication	
135	acetamritsar.ac.in Internet Source	<1%
136	dokumen.pub Internet Source	<1%
137	dspace.lpu.in:8080 Internet Source	<1%
138	gch2020.eu Internet Source	<1%
139	grietinfo.in Internet Source	<1%
140	hindustanuniv.ac.in Internet Source	<1%
141	openaccess.altinbas.edu.tr Internet Source	<1%
142	technodocbox.com Internet Source	<1%
143	www.bartleby.com Internet Source	<1%
144	www.ijraset.com Internet Source	<1%
145	www.ncbi.nlm.nih.gov Internet Source	<1%
146	www.taci-journal.org	

Internet Source

<1%

147 www.techscience.com
Internet Source

<1%

148 Shelke Kavita, Shinde S. K.. "Metaheuristic Evolutionary Algorithms: Types, Applications, Future Directions, and Challenges", 2023 3rd International Conference on Intelligent Technologies (CONIT), 2023
Publication

<1%

149 Sreenu Konda, Chandrashekhar Goswami, Somasekar J, Ramana K, Ramesh Yajjala, N. S. Koti Mani Kumar Tirumanadham. "Optimizing Diabetes Prediction: A Comparative Analysis of Ensemble Machine Learning Models with PSO-AdaBoost and ACO-XGBoost", 2023 International Conference on Sustainable Communication Networks and Application (ICSCNA), 2023
Publication

<1%

150 Wenli Li. "PSO Based Wireless Sensor Networks Coverage Optimization on DEMs", Lecture Notes in Computer Science, 2012
Publication

<1%

151 "Nature Inspired Computing for Wireless Sensor Networks", Springer Science and Business Media LLC, 2020
Publication

<1%

152	M. Neema, E.S. Gopi, Pulakunta Sandeep Reddy. "Optimizing Broadband Access and Network Design in Wireless Mesh Networks using Multi-Objective Particle Swarm Optimization", Procedia Computer Science, 2023 Publication	<1%
153	salford-repository.worktribe.com Internet Source	<1%

Exclude quotes	Off	Exclude matches	Off
Exclude bibliography	Off		