Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

An Enhanced NSGA-II Approach for Multi-Criteria Energy-Efficient Barrier Coverage Scheduling in Wireless Sensor Networks

Subash Harizan

subashharizan@gmail.com

SRM Institute of Science and Technology

Akhilendra Khare

Galgotias University

Rajeev Kumar

Nalanda College of Engineering

Avneesh Vashistha

Sharda University

Research Article

Keywords: Wireless sensor networks, sensor scheduling, barrier coverage, connectivity, evolutionary algorithms, NSGA-II

Posted Date: November 16th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11361222/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

An Enhanced NSGA-II Approach for Multi-Criteria Energy-Efficient Barrier Coverage Scheduling in Wireless Sensor Networks

Subash Harizan, Akhilendra Khare[2], Rajeev Kumar[3], Avneesh Vashistha[3]

[1,2,3,4]*Department of Computer Science and Engineering*
[1]*SRM Institute of Science and Technology, Delhi NCR, Ghaziabad, Uttar Pradesh, India*
[2]*Galgotias University, Greater Noida, Uttar Pradesh, India*
[3]*Nalanda College of Engineering, Bihar, India*
[4]*Sharda University, Agra, Uttar Pradesh, India*

Abstract

Intruder detection is a critical application in wireless sensor networks (WSNs), where barrier coverage emerging as a key area of interest for researchers and the scientific community. Barrier coverage involves using sensors to create protective boundaries that safeguard vital assets and infrastructure, such as national borders, coastlines, and battlefields. Among the various approaches, sensor node scheduling is a highly efficient technique for extending the lifetime of WSNs. However, ensuring energy efficient barrier by scheduling minimum number of sensor nodes with long haul operation is a challenging due to the potential failure of sensor nodes. To address this problem an eminent multi-objective evolutionary algorithm named NSGA-II (Non-dominated sorting genetic algorithm) is propose for multi-objective challenges of scheduling to form barrier. Our approach features with a Linear Programming (LP) formulation for the proposed problem with an optimized depiction of chromosomes and a well-designed of multiple fitness function to assess their performance. Additionally, we introduce a novel mutation operation to enhance algorithm performance and convergence speed. Extensive simulations across various network scenarios demonstrate the effectiveness of our proposed algorithm, showing improved results compared to existing approaches. Additionally, hypothesis testing of the simulation results is carried out through ANOVA followed by detailed post-hoc evaluation.

Keywords: Wireless sensor networks, sensor scheduling, barrier coverage, connectivity, evolutionary algorithms, NSGA-II

1. Introduction

Barrier coverage [1, 2] has become a basic component of wireless sensor networks (WSNs) in the current environment of widespread surveillance and security, especially for applications requiring intrusion detection citebutun2013survey, bukhari2024secure, rajasoundaran2024secure. Barrier coverage guarantees constant surveillance of a border or area, therefore stopping unwelcome entrance or noting possible incursions. This is especially important in sensitive regions like military bases, border security, and critical infrastructure protection. But getting good barrier coverage is

*Corresponding author

Email addresses: subashharizan@gmail.com (Subash Harizan), a.khare0007@gmail.com (Akhilendra Khare[2]), rajeevsinghs2006@gmail.com (Rajeev Kumar[3]), avneeshvashistha@gmail.com (Avneesh Vashistha[3])

difficult because there are restrictions on sensor energy resources and a need for best deployment techniques. Long-term dependability of these sensor networks depends on effective use of finite energy resources together with strong surveillance coverage.

Because of their limited sensing and communication ranges, scheduling sensor nodes to create barrier coverage is difficult. Key problems include adequately covering regions while maintaining connection between nodes and with the base station. Because low-energy nodes deplete more quickly, causing frequent changes and possible performance reduction, integrating nodes with different energy levels makes the process even more difficult. Thinking about the residual energy of every node helps to solve these problems. This complexity emphasizes the need of sophisticated optimization methods able to change with several situations and provide better performance. Using a population-based approach that develops several solutions concurrently, NSGA-II citedeb2002fast2,harizan2020novel efficiently satisfies this demand. Its quick non-dominated sorting algorithm and calculation of crowding distance support diversity in solutions and stop early convergence to subpar outcomes. These features make NSGA-II especially fit for tackling multi-objective issues like reaching energy-efficient barrier coverage in wireless sensor networks (WSNs). By balancing several goalsincluding maximizing coverage, lowering energy use, and increasing network lifetimeNSGA-II helps one to develop deployment plans both efficient and environmentally friendly. Energy economy is very important considering sensors' restricted battery life. Due to their remote or dangerous location, replacing or recharging sensors is difficult and expensive; therefore, it is imperative to increase the working life of a sensor network. This can be done by deliberately turning on required sensors while maintaining others in low-power mode. To balance energy use across the network and reduce needless coverage, NSGA-II improves sensor activation schedules, therefore saving energy and improving overall reliability and performance. Furthermore NSGA-II's ability to simultaneously address several goals goes beyond energy and coverage issues. It can include dynamic environmental elements, sensor installation difficulties, and geographical limits. For instance, the algorithm can change coverage plans in real time to handle sensor failures or environmental changes, hence guaranteeing ongoing and efficient barrier coverage in unexpected surroundings. Integrating NSGA-II into barrier coverage design for intrusion detection is a major improvement for WSNs. Energy-efficient, dependable, and adaptive barrier covering solutions may be created to address the complicated issues of intrusion detection by means of NSGA-II's multi-objective optimization features. This approach not only improves sensor network performance and longevity but also gives a flexible framework to satisfy the changing requirements of current security apps. Advanced evolutionary algorithms like NSGA-II will help to stimulate creativity and increase the efficacy of barrier coverage systems as research in this area advances.

1.1. Our contribution

Given the aforementioned considerations, this study introduces a multi-objective solution for energy-efficient sensor node scheduling aimed at optimizing barrier coverage under conflicting objectives, utilizing an NSGA-II-based approach. The proposed algorithm exhibits superior performance. The significant contributions of this work are outlined as follows.

- Modeling the scheduling problem for energy eminent barrier coverage through Linear Programming (LP).
- Introducing an NSGA-II-based algorithm to solve the multi-objective scheduling problem for barrier coverage, addressing four conflicting objectives: (i) minimizing the number of selected sensor nodes, (ii) ensuring complete barrier coverage, (iii) maintaining connectivity between selected nodes and the base station, and (iv) prioritizing the selection of sensor nodes with higher energy levels during each scheduling cycle.

- Designing robust chromosome structures for the scheduling problem for barrier coverage to ensure the validity of solutions following mutation and crossover operations.
- Design of multiple conflicting objective functions to assess particle performance.
- Enhancing the performance of NSGA-II with a novel mutation operation.
- Performing a thorough simulation of the proposed algorithm, along with a comparative analysis of its results against established and related algorithms..

1.2. Outline of the paper

The subsequent sections of the paper are structured as follows. Section 2 represents the related works pertinent to our work. Section 3 provides an overview of the NSGA-II. The system model, terminologies used, and problem formulation are detailed in Section 4. The proposed algorithm and its corresponding simulation results are discussed in Sections 5 and 6, respectively. Lastly, Section the paper is concluded with remark in section 7.

2. Related works

In [3], authors provide a general intrusion detection system (IDS) for wireless sensor networks (WSNs) to fight complex, empowered attackers. Using machine learning algorithms for great detection accuracy and minimal false alarms, the planned IDS blends anomaly detection, signature-based approaches, and distributed design. The system is meant to be energy-efficient and scalable, therefore addressing the limitations of WSNs while yet guaranteeing strong protection against a broad spectrum of assaults. By means of simulations, the authors show how well their technique identifies several kinds of intrusions, therefore presenting a means of protecting WSNs against sophisticated attacks. In [4], an improved maximum collaborative intrusion detection system (IDS) for wireless sensor networks (WSNs) is presented. Improved accuracy and lower false positives are achieved by the system's integration of several sensor nodes working together to identify invasions. Using a hybrid method blending machine learning techniques and a cooperative framework, the authors improve detection performance. This technique targets the resource limits of WSNs by reducing energy use and maintaining great detection rates. The suggested IDS is proven to be able to identify several attacks in dynamic network settings by simulations validating its performance.

Authors in [5] offers an improved empirical component analysis (ECA) method for intrusion detection in wireless sensor networks (WSNs). By means of analysis of the data elements of the sensor network, the technique helps to spot anomalies suggestive of intrusions therefore enhancing detection accuracy. The improved ECA combines machine learning with feature extraction methods to differentiate normal from damaging actions. Simulations show that this method reduces false positives while yet correctly spotting a range of attacks. The suggested approach provides a strong and effective means of protecting WSNs against complex attacks.

[6] authors present a method for intrusion detection in wireless sensor networks (WSNs) based on an evolutionary algorithm intended to reduce energy usage and complexity. The suggested approach uses evolutionary algorithms to maximize parameter adjustment and feature selection, so enabling the system to detect intrusions efficiently and so help to lower computational overhead. This method seeks to solve the issue of scarce resources in WSNs by reducing the processing and energy demands usually connected with conventional detecting methods. Results of simulations reveal the evolutionary-based model's great detection accuracy for many attacks makes it appropriate for dynamic and resource-constrained network settings.

Writers in [7] give a thorough review on how swarm and evolutionary algorithms help to power intrusion detection systems (IDS). The writers investigate how these naturally inspired algorithmsincluding genetic algorithms, particle swarm optimization, and ant colony optimizationhelp IDS in network security run better. The survey emphasizes these algorithms' sensitivity in spotting sophisticated intrusions in terms of adaptability, scalability, and efficiency. The article looks at several studies that have used swarm and evolutionary methods for intrusion detection and notes their benefits for lowering false alarms, lowering calculation complexity, and enhancing detection accuracy. Additionally noted are the difficulties and next research directions for including these methods into real IDS systems.

In [2] work offers a better way to install sensor nodes to provide barrier coverage in wireless sensor networks (WSNs). The writers suggest a fresh approach using a genetic algorithm (GA) to maximize the deployment of sensor nodes, therefore guaranteeing that the region is sufficiently covered while lowering energy use and installation costs. The technique improves classic GA by adding adaptive elements including dynamic mutation rates and fitness functions fit for barrier coverage criteria. The suggested method was assessed using simulations and contrasted with already used techniques. The findings show that in WSNs it greatly lowers energy use, boosts coverage, and guarantees more effective resource management.

Authors in [8, 9] present an extensive analysis of the use of evolutionary algorithms to tackle connection and coverage problems in wireless sensor networks (WSNs). The chapter looks at several evolutionary methods, including genetic algorithms, particle swarm optimization, and multi-objective optimization strategies, for resolving important WSN issues as sensor node deployment, scheduling, and energy economy. The authors discuss the advantages and drawbacks of these methods with reference to their part in balancing network coverage, connections, and energy usage. The study illuminates design frameworks and real-world factors to provide ideas on the creation of strong and effective WSNs for many uses.

Authors present in [10] a machine learning-based intrusion detection system (IDS) meant for wireless sensor networks (WSNs). The study examines the weaknesses in WSNsdistributed and sometimes unprotectedwhich make them vulnerable to many attacks. To spot and categorize possible intrusions, the writers suggest a hybrid IDS combining several machine learning approachessupervised and unsupervised ones To spot anomalies, the system is developed on network traffic data and includes characteristics including signal strength, node behavior, and communication patterns. The suggested IDS is highly accurate, has low false positive rates, and can change with the network conditions.

The studies in [11] seek to maximize sensor node scheduling in wireless sensor networks (WSNs) to provide good coverage and connectivity while using energy friendly. This paper presents a better genetic algorithm to plan sensor nodes in target-based WSNs that strikes a balance between energy economy, network connection, and adequate area coverage. Likewise, paper [12] broadens this study by suggesting a fresh NSGA-II-based strategy fit for industrial WSNs. This technique improves multi-objective optimization by concurrently tackling coverage, connectivity, and energy limits. Both studies show, using simulations, how well their techniques work by exhibiting better results over conventional approaches in WSN management.

In [12], authors present a new technique for sensor node scheduling in industrial wireless sensor networks (WSNs) utilizing the Non-dominated Sorting Genetic Algorithm II (NSGA-II). The suggested approach seeks to maximize two important elements for reliable and efficient sensor network operations: coverage and connectivity. To address the multi-objective optimization problem, the NSGA-II algorithm is used to maximize area coverage while simultaneously preserving network connectivity and reducing energy usage. To mimic actual industrial settings, the authors include many limits that guarantee the approach is both possible and successful.

Using a GSWO-CatBoost technique, [13] provides a fresh approach to improve intrusion detection in wireless sensor networks (WSNs). The authors suggest combining the CatBoost machine learning method with the General Simultaneous Weight Optimization (GSWO) approach to increase intrusion detection accuracy. The GSWO approach guarantees that just the most pertinent features are employed for classification therefore streamlining the feature selection process. Then, CatBoosta gradient boosting methodis used to identify anomalies linked with possible assaults and categorize the network data. Using multi-objective evolutionary algorithms (MOEAs), authors in [14] concentrate on the best possible positioning of relay nodes in cluster-based wireless sensor networks (WSNs). The authors present a fresh way to handle the twin difficulties of optimizing network coverage and maintaining connectivity while installing relay nodes. The optimization is presented as a multi-objective issue with the objectives of improving coverage, guaranteeing reliable communication between clusters, and cutting energy use. Conflicting goals are balanced in the study using cutting-edge evolutionary algorithms. Particularly in big real-world applications like environmental monitoring and industrial automation where both coverage and connectivity are vital, the study emphasizes the possibilities of MOEAs for tackling difficult deployment issues in WSNs.

3. An overview of Non-dominated Sorting (NSGA-II)

The Non-dominated Sorting Genetic Algorithm II (NSGA-II) [15, 16, 17]is a widely used multi-objective evolutionary algorithm (MOEA) designed to optimize multiple objectives simultaneously without being influenced by other solutions. Like the genetic algorithm (GA), NSGA-II uses an initial population and genetic operations including selection, crossover, and mutation. The initial population is composed of a predetermine number (denoted as P_S) of randomly generated chromosomes. Initially, a population P_0 is created, and the fitness values for each objective are measured for an individual chromosome. Domination of an individual chromosome is determine by comparing with all other chromosomes. Based on domination levels non-dominated fronts $\psi_1, \psi_2, \ldots, \psi_k$ are identified. Chromosomes are assigned ranks corresponding to their respective front numbers, with smaller front numbers representing higher (better) ranks.

Next, same size (P_S) of offspring population O_0 are generated from an initial population $\mathbb{P}_0$ by applying the operations as selection, crossover and mutation. At a given generation (say g), both the populations P_g and O_g are combined together to form a merged population R_t of size $2 \times P_S$. Fast non-dominated sorting is then utilize to R_g to identify the non-dominated fronts $\psi_1, \psi_2, \ldots, \psi_k$. The next generation population P_{g+1} of size P_S is formed by filling it with chromosomes from ψ_1, followed by ψ_2, and so on. If ψ_k is the last front required to fill the population, and it contains more chromosomes than needed, the remaining chromosomes are selected based on crowding distance to maintain diversity. Chromosomes with less crowding are preferred to ensure a well-distributed population. This process is repeated to generate subsequent populations until the stopping criteria are met. Figure 1 depicts the flowchart of the NSGA-II algorithm.

4. System models and problem statement

4.1. Network prototype

This work considers a rectangle WSN area with randomly deployed N number of sensor nodes, which are capable of continuously or periodically monitoring a predefined area where they are deployed/placed. Each sensor node has a sensing range R_s and can detect events within this range. We adopt the binary sensing model as described in [18]. The data transmission process is organized into rounds, following the LEACH protocol [19]. During each round, sensor nodes convey the sensed

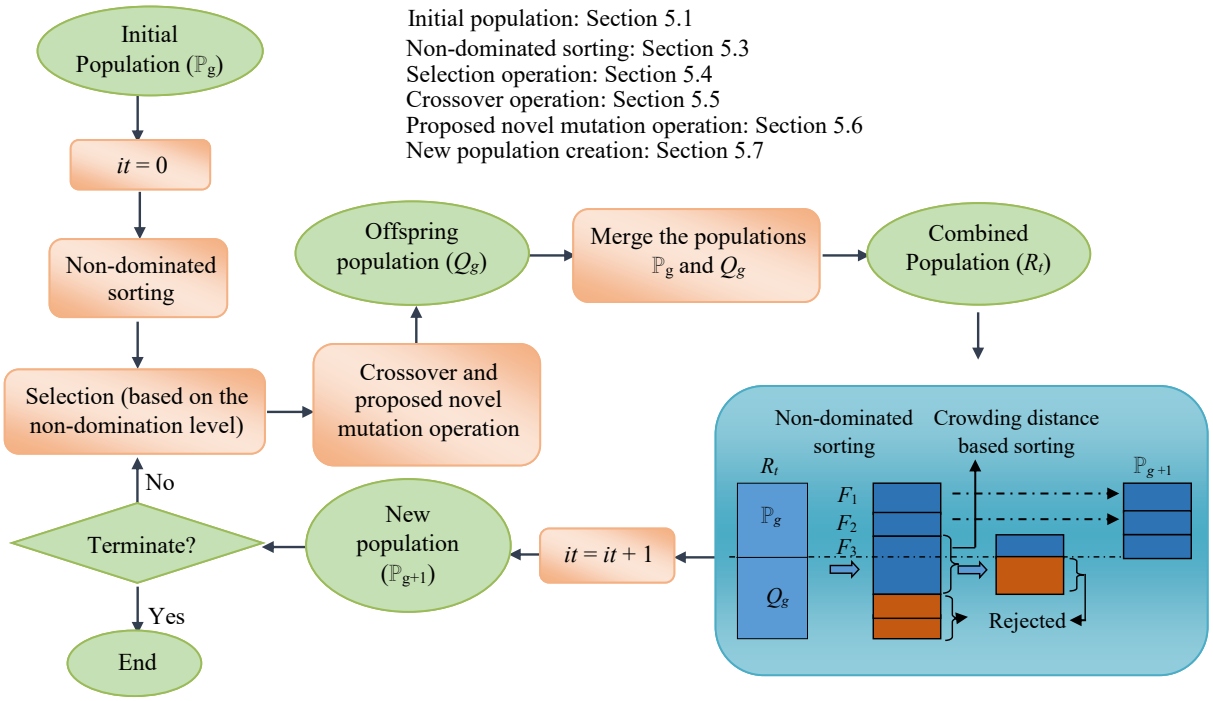


Fig. 1: NSGA-II flowchart.

and accumulated data to the base station (BS), either directly or via other nodes. Communication between sensor nodes is only possible if they are within each other's communication range R_c.

4.2. Problem statement

The problem can be define as:

"To activate or schedule minimum number of sensor nodes with maximum energy level from N number of deployed sensor nodes to form an energy efficient connected barrier coverage which can detects intruder trying to cross the network region through crossing path."

As an example consider a rectangular WSN as shown in Fig. 2 of length L and width W with 28 randomly deployed sensor nodes $\{s_1, s_2, s_3, \ldots, s_{28}\}$. We assume sensor nodes can send its sening data to its nearby base station (BS) situated at any one of edge. From Fig.2 it can be observed that out of 28 SNs only 10 Sns $\{p_2, p_5, p_7, p_9, p_{10}, p_{14}, p_{17}, p_{19}, p_{20}, p_{21}, p_{25}, p_{28}\}$ are schedule to form barrier similarly for second round 12 SNs are schedule to form barrier. Thus, this process of scheduling of SNs remain continue till there is situation that no barrier coverage is formed with the active sensor nodes.

4.3. Terminological terms used

Here, we first define the key terminologies used in the proposed algorithm:

- $S = s_1, s_2, s_3, \ldots, s_N$ represents the set of N deployed sensor nodes.
- R_s and R_c denote to the sensing and communication ranges of the respectively.
- $dis(s_i, s_j)$ represents the Euclidean distance between node s_i and sensor node s_j.
- $rE_(s_i)$ represents the residual energy of sensor node s_i.

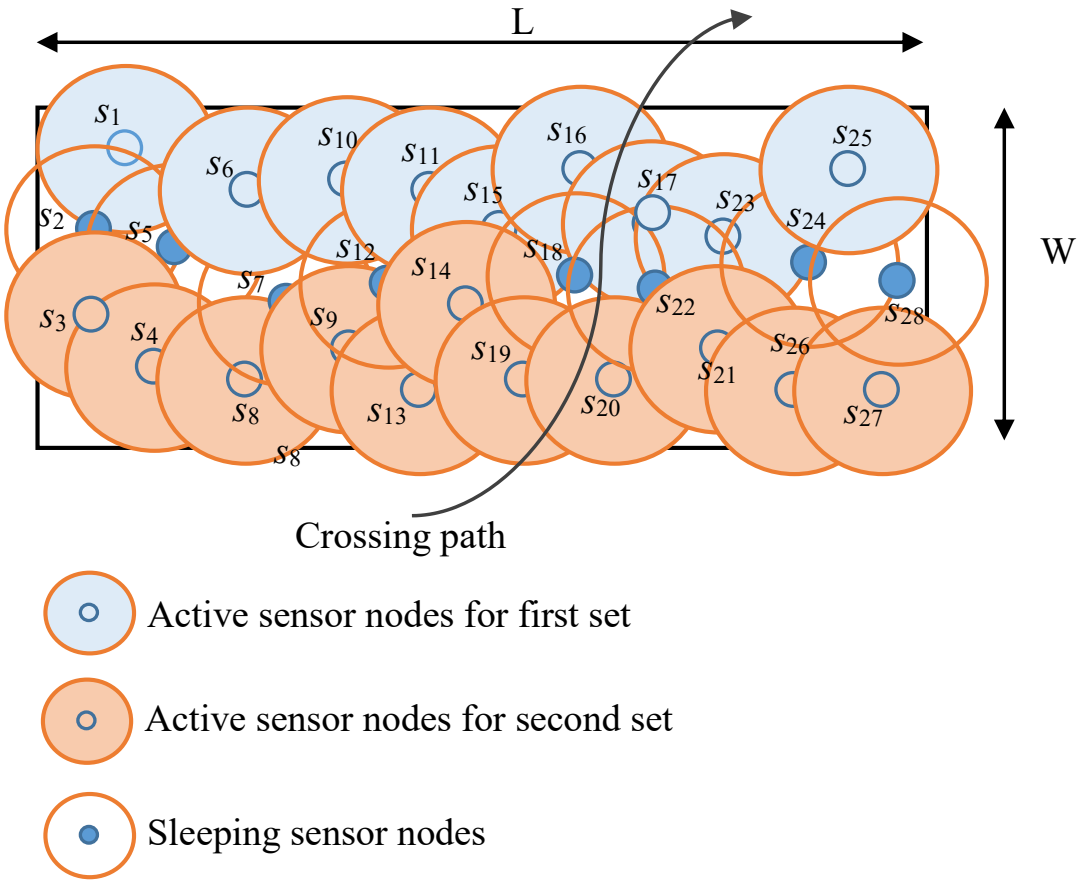


Fig. 2: Network scenario of barrier coverage.

4.4. Formulation of linear programming (LP) for the define problem

The above problem can now be formulated as a Linear Programming (LP) problem. Let α_i, β_{ij}, and ρ_{ij} be the Boolean variables, all of which are defined as follows:

$$\alpha_i = \begin{cases} 1, & \text{if } s_i \text{ is chosen for present scheduling} \\ 0, & \text{else.} \end{cases} \tag{1}$$

$$\beta_{ij} = \begin{cases} 1, & \text{if } s_i \text{ lies within } R_c \text{ of } s_j \\ 0, & \text{else.} \end{cases} \tag{2}$$

$$\rho_i = \begin{cases} 1, & \text{if } s_i \text{ lies within } R_c \text{ of BS} \\ 0, & \text{else.} \end{cases} \tag{3}$$

The Linear Programing Problem (LPP) for the define problem can be formulated as follows:

$$\text{Minimize } F = \sum_{i=1}^{N} \alpha_i \tag{4}$$

$$\Big(\sum_{j=1}^{K} \beta_{ij} \times \alpha_j + \rho_i\Big) \times \alpha_i \geq 1, \forall i, 1 \leq i \leq N, \beta_{ij} \in \{0,1\}, \rho_j \in \{0,1\} \tag{5}$$

$$E_R(s_i) \times \alpha_i \geq Th, \forall i, 1 \leq i \leq N, \alpha_i \in \{0,1\} \tag{6}$$

Equation 4 ensures the activation of minimum number of sensor nodes where constraint 5 guarantees that every selected sensor node either has a data forwarding node or is within the direct transmission range of the base station (BS). The variable ρ_i indicates whether the sensor node s_i lies within the transmission range of the BS. Finally, constraint 6 ensures that the energy of each sensor node remains above the specified threshold Th.

4.5. Multi-criteria optimization problem

In multi-criteria optimization problems, the criteria are typically in conflict with one another, meaning that optimizing one objective may hinder the optimization of another [20]. As a result, optimizing a solution for a single criteria often leads sub-optimal or adverse results for the remaining criterion. Achieving a solution that optimizes all objectives simultaneously is nearly unbearable. Thus, finding a rational solution which satisfies all criterion to an satisfactory level is extremely desirable.

The multi-criteria optimization problem (MOP) [21] is formulated as follows.

$$\begin{aligned} &\text{Minimize } F(\overrightarrow{O}) = \{F_1(\overrightarrow{O}), F_2(\overrightarrow{O}), \ldots, F_n(\overrightarrow{O})\} \\ &\text{Constrained by } \overrightarrow{O} \in \Theta \end{aligned} \tag{7}$$

Here, $\overrightarrow{O} = \{a_1, a_2, \ldots, a_n\}$ represents an n-dimensional vector or solution, Θ is the solution space, $F : \Theta \to \mathbb{R}^n$ contains m real-valued objectives $\{F_1, F_2, \ldots, F_n\}$, and objective space is represented by $\mathbb{Q}^n$.

As any optimization problem can be reformulated as a minimization problem, we assume all objectives are minimization objectives without loss of generality. To better explain the terminology used in this context, we provide the following definitions:

Definition 1. [**Pareto dominance**] Assuming $\overrightarrow{\delta} = \{\delta_1, \delta_2, \ldots, \delta_n\}$ and $\overrightarrow{\lambda} = \{\lambda_1, \lambda_2, \ldots, \lambda_n\}$ be two solutions. We say that $\overrightarrow{\delta}$ dominates $\overrightarrow{\lambda}$, denoted by $\overrightarrow{\delta} \succ \overrightarrow{\lambda}$, if and only if $F_j(\overrightarrow{\delta}) \leq F_j(\overrightarrow{\lambda})$ for all $j \in \{1, 2, \ldots, n\}$, and there is at least one instance of $i \in \{1, 2, \ldots, n\}$ so that $F_i(\overrightarrow{\delta}) < F_i(\overrightarrow{\lambda})$ (for minimization objectives).

Definition 2 (**Pareto-optimal**)**.** Solution $\overrightarrow{\delta^*} \in \Omega$ is considered Pareto optimal provided that no alternative solution $\overrightarrow{\lambda} \in \Omega$ is found where $\overrightarrow{\lambda} \succ \overrightarrow{\delta^*}$.

Definition 3 (**Pareto set and Pareto front**)**.** The set containing all Pareto optimal solutions is known as Pareto set (PS), where $PS = \{\overrightarrow{\delta^*} \mid \nexists \overrightarrow{\lambda} \in \Omega, \overrightarrow{\lambda} \succ \overrightarrow{\delta^*}\}$. Similarly, the collection of objective vectors associated with the solutions in the Pareto set is called the Pareto front (PF), where $PF = \{(F_1(\overrightarrow{\delta^*}), F_2(\overrightarrow{\delta^*}), \ldots, F_n(\overrightarrow{\delta^*})) \mid \forall \overrightarrow{\delta^*} \in PS\}$.

5. Proposed NSGA-II based algorithm

The proposed algorithm includes several key components: chromosome representation, initialization of the population, design of fitness function with function of multiple fitness, genetic operators (selection, crossover, and mutation), and creation of subsequent generation population utilizing crowding distance and fast non-dominated sorting. Each of individual steps are explained in detail with examples in the subsequent sections.

5.1. Chromosome representation and initialization of initial population

A complete individual solution to the problem is represented by a chromosome. In this work, binary string $\{g_i | g_i \in \{0,1\}, \forall i, 1 \leq i \leq N\}$ of length N is used to depict it. Here, N represents the deployed sensor nodes across the network. According to the rule given below, the i^{th} gene value g_i determines the corresponding sensor node s_i is active or not.

$$g_i = \begin{cases} 1, & \text{If } s_i \text{ is activated.} \\ \\ 0, & \text{Otherwise.} \end{cases} \tag{8}$$

Illustration 1. Let us assume a WSN with twenty sensor nodes $S = \{s_1, s_2, \ldots, s_{20}\}$ as shown in Figure 3(a). Figure 3(b) represents the corresponding chromosome for the given network of length 20 with only nine active sensor nodes s_2, s_5, s_7, s_9, s_{10}, s_{14}, s_{17}, s_{19}, s_{20}. From Figure 3(b) it is apparent that corresponding value of gene is 1 at position 5 (i.e., $g_5 = 1$). This indicates that the sensor node at which implies that s_5 is chosen as an active node in the current round. Likewise, sensor nodes s_2, s_7, s_9, s_{10}, s_{14}, s_{17}, s_{19}, s_{20} are also designated as active nodes in the current round. The value of genes g_1, g_3, g_4, g_6, g_8, g_{11}, g_{12}, g_{13}, g_{15}, g_{18} is 0 which implies that the sensor nodes s_1, s_3, s_4, s_6, s_8, s_{11}, s_{12}, s_{13}, s_{15}, s_{18} will be in sleep mode in this round.

A set of randomly generated chromosomes of predefine size forms the initial population. The algorithm for generating an initial population is presented in Algorithm 1.

Algorithm 1 : Initialization of Initial-Population

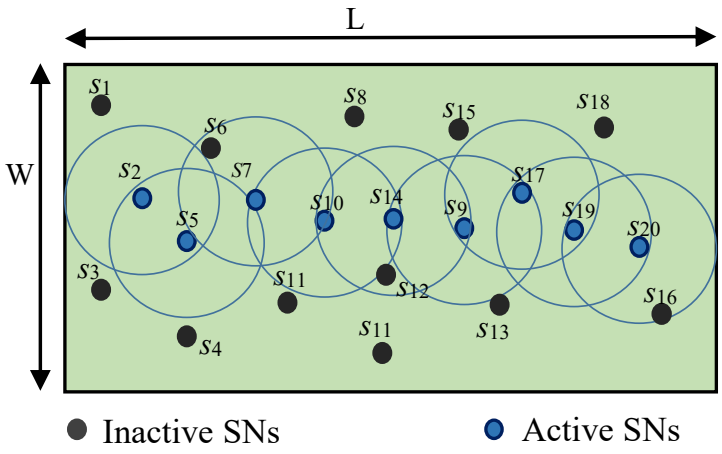
Input: (1) M: total number od sensor nodes deployed.
(2) P_S: predefine size of population, i.e.. chromosome count.

Output: Population of size P_S i.e., number of chromosomes.

```
1: Pop = 0
2: Boolean g_j, ∀j, 1 ≤ i ≤ M
3: for k = 1 to P_S
4:     for j = 1 to M
5:         g_j = rand()% 2          /*rand() yields a random positive whole number*/
6:     end for
7:     Pop = Pop ∪ {g_j | ∀j, 1 ≤ j ≤ M}
8: end for
```

5.2. Computation of the different fitness values

Individual chromosomes are evaluated based on their fitness values. The goal is to achieve barrier coverage of the network area by selecting the less units of sensor nodes as active, while ensuring that the selected sensor nodes are connected to the sink for data transmission. Thus, the multiple objectives of the proposed work are outlined as follows.



(a)

s_1	s_2	s_3	s_4	s_5	s_6	s_7	s_8	s_9	s_{10}	s_{11}	s_{12}	s_{13}	s_{14}	s_{15}	s_{16}	s_{17}	s_{18}	s_{19}	s_{20}
0	1	0	0	1	0	1	0	1	1	0	0	0	1	0	0	1	0	1	1

(b)

Fig. 3: (a) A network scenario with total 20 sensor nodes and 9 active sensor nodes, (b) chromosome for the corresponding network scenario.

- *Goal 1 (Identifying the minimum set of sensor nodes)*: The number of ones in a chromosome represents number of selected sensor nodes. Thus, first goal can be represented as follows.

$$Goal\ 1\text{: Min } O_1 = \sum_{i=1}^{M} g_i \tag{9}$$

This indicates that chromosomes with less number of one are preferred.

- *Goal 2 (Ensure connectivity of the activated sensor nodes)*: The calculation of the connectivity cost $CON_{cost}(s_i)$ of sensor node s_i is calculated as:

$$CON_{cost}(s_i) = \begin{cases} +1, & \text{if } |C_{con}(s_i)| \geq 1 \\ -1, & \text{otherwise.} \end{cases} \tag{10}$$

where, $C_{con}(s_i)$represents the set of sensor nodes that lie within the communication range (R_{com}) of s_i in the direction of the BS. It implies that s_i is capable of transmitting data to any of the sensor nodes in $C_{con}(s_i)$. In other words,

$$\begin{aligned} C_{con}(s_i) \quad = \quad & \{s_j | (g_j = 1) \& (dis(s_i, s_j) \leq R_{com}) \\ & \& (dis(s_i, BS) \geq dis(s_j, BS)), \forall j, 1 \leq j \leq M, i \neq j\} \end{aligned} \tag{11}$$

The connectivity amongst the active sensor nodes can be assured as follows.

$$\textit{Goal } 3\text{: Max } O_2 = \sum_{i=1}^{M} \big(g_i \times CON_{cost}(s_i)\big) \tag{12}$$

- *Goal 3 (Maximize average energy level of the sensor nodes)*: The sensor nodes chosen should retain enough energy to function for a designated number of rounds. A higher minimum energy among the selected nodes signifies a better choice. Therefore, the fourth objective is expressed as:

$$\textit{Goal } 3\text{: Max } O_3 = Min\{E_R(s_i)|g_i = 1, \forall i, 1 \leq i \leq M\} \tag{13}$$

The fitness value for an individual chromosome $\{g_1, g_2, g_3, \ldots, g_M\}$, $\forall g_i \in \{0, 1\}, 1 \leq i \leq M$, with respect to there goals O_1, O_2 and O_3 can be accessed according to equations (9), (12), and (13) respectively.

Remark 5.1. *It is evident that the aforementioned objectives are mutually conflicting. Thus, to maintain the efficient connectivity amongst the active sensor nodes large quantity of sensor nodes are need to be selected, hampers the goal 1.*

5.3. Sorting with fast non-dominated

Once the population (initial) is generated and fitness values are assessed, the non-dominated sorting algorithm categorizes chromosomes into non-dominated fronts$\{\psi_1, \psi_2, \ldots, \psi_m\}$. Each chromosome $\omega(i)$ is analyzed against all others to determine dominance. During this process, the algorithm calculates D_i, the set of chromosomes dominated by $\omega(i)$, and $d_c(i)$, the number of chromosomes that dominate $\omega(i)$. Chromosomes with $d_c(i) = 0$ are assigned to the first front, ψ_1. For the second front, ψ_2, the domination count for each chromosome in D_i is reduced by one. When $d_{count}(j)$ of any chromosome $\omega(j)$ becomes zero, it is moved to ψ_2. This iterative sorting continues, forming ψ_3, ψ_4, and subsequent fronts until all chromosomes are ranked.

5.4. Selection

During this phase, chromosomes are selected for crossover and mutation. Selection methods such as roulette-wheel, rank, and tournament are commonly used. For single-objective problems, selection operation is usually determined by fitness, favoring chromosomes with superior fitness. In contrast, the proposed multi-objective algorithm selects chromosomes based on their domination level and crowding distance.

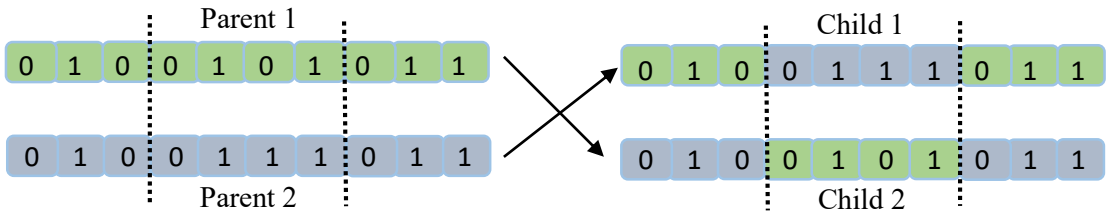
Once the initial population P_0 is established and subsequent offspring populations (Q_0) are generated, selection is initially based on non-domination levels. In the following stages, the Crowded-Comparison operator [17] is applied for selection. Considering two chromosomes, $\omega(i)$ and $\omega(j)$ the Crowded-Comparison operator $\succ_{CC}$ asserts that $\omega(i)$ is favored over $\omega(j)$ if:

$$\omega(i) \succ_{CC} \omega(j) \Longleftrightarrow \{(R_i < R_j) \vee ((R_i == R_j) \wedge (cd(i) > cd(j)))\} \tag{14}$$

Here, R_i denotes the non-domination level, and $cd(i)$ signifies the crowding distance of $\omega(i)$. A lower non-domination level and a higher crowding distance are preferred. The method for calculating crowding distance is detailed in Section 5.7.

5.5. Crossover

Crossover operation is carried out once selection operation is performed. Different crossover methods are available, such as one-point, two-point, and uniform crossover. Our proposed work employs a two-point crossover, where the crossover points are randomly determined, and the chosen parents swap their genetic information after these points (see Figure 4).



(a)

Fig. 4: (a) Crossover operation and (b) Mutation operation.

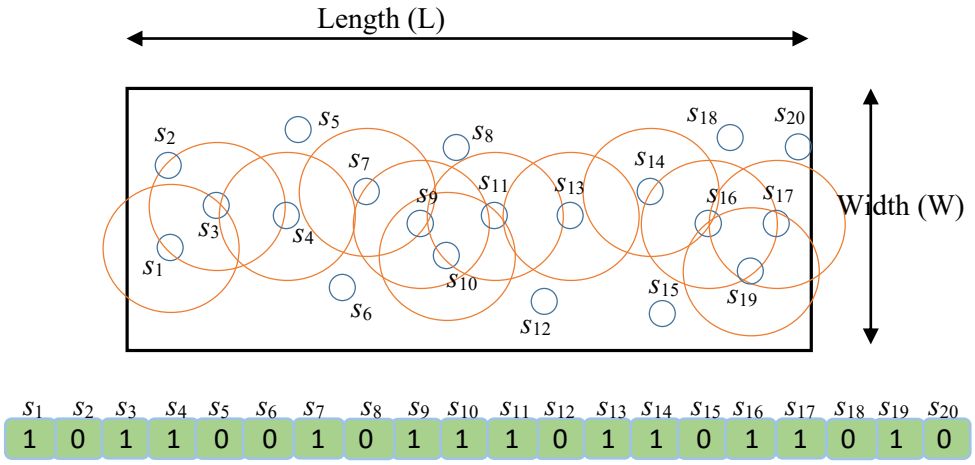
5.6. Mutation

After crossover, mutation is performed by setting a randomly chosen gene's value to zero, intending to enhance the offspring. This mutation targets a random gene position, assigning it a value of 0. As compared to this traditional mutation approach, in our novel mutation, we begin by detecting redundant sensor nodes activated during barrier coverage formation where deactivation of those sensor nodes neither effect the barrier nor the connectivity of the barrier coverage. We deactivated these sensor nodes from consideration of barrier coverage which can consume energy as well as leads to more efficient chromosomes.

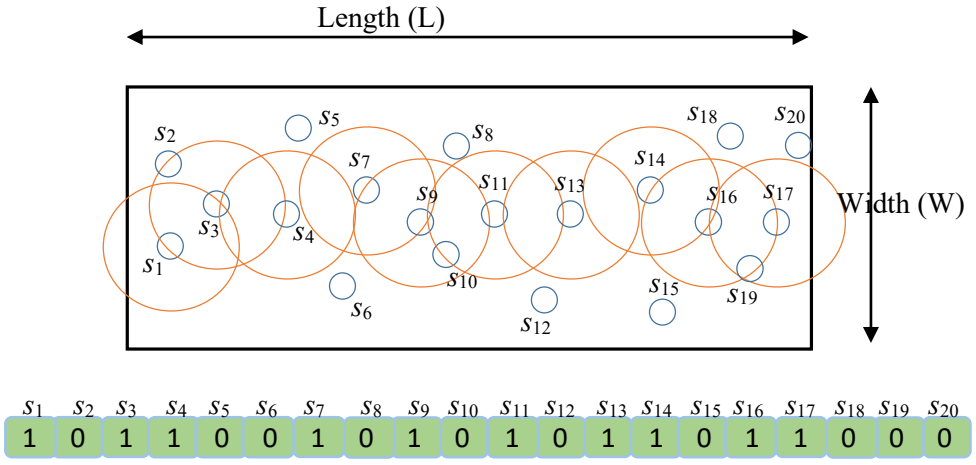
Illustration 2. Let us assume a rectangular WSN of length L and width W with 20 sensor nodes. Amongst the given 20 sensor nodes only 12 sensor nodes are activated to form the barrier coverage. The given network scenario with corresponding chromosome is shown in Fig. 5 (a). From Fig. 5 (a) it can be perceived that removal of sensor nodes s_{10} and s_{19} neither hamper formation of barrier coverage and connectivity. Therefore, our proposed novel mutation operation will deactivate sensor nodes s_{10} and s_{19}. The corresponding network scenario and chromosome after novel mutation operation is depicted in Fig. 5 (b). The chromosome obtained after proposed mutation operation is optimized and better.

5.7. Formation of subsequent generation of population

We now generate next generation population P_{g+1} of same size P from the existing current population P_g. Initially, genetic operations are applied on the current population P_g to generate the new offspring population Q_g of same size P. Thereafter, merged population $R_g = P_g \cup Q_g$ of size $2 \times P$ is formed by combining the Q_g and P_g. Next, all the non-dominated fronts $(\psi_1, \psi_2, \psi_3, \ldots.)$ of various levels are identified by applying the fast non-dominated sorting. Then, chromosomes from subsequent fronts beginning from ψ_1, ψ_2, ψ_3 and so on are added in P_{g+1} generation until its size become P. It should be noted that the last front say (ψ_{z+1}) may include more chromosomes than necessary number to complete the new population P_{g+1} of size P. In other words, $|\psi_1 \cup \psi_2 \cup \psi_3 \cdots \cup \psi_z| < P$ while $|\psi_1 \cup \psi_2 \cup \psi_3 \cdots \cup \psi_{z+1}| > P$. In such situation, chromosomes from final front ψ_{z+1} are ranked based on crowding distance. Chromosomes having higher crowding distance are chosen for fill the remaining slots of P_{g+1} ensuring its total size as P.



(a)



(b)

Fig. 5: Proposed novel mutation operation on the given barrier coverage scenarios, (a) before mutation applied, (b) after mutation applied.

The crowding distance measures how far, on average, a point in a d-dimensional space lies from the two neighboring points along each dimension. Thus, when the crowding distance of a point is high, it implies that the point resides in a region with lower solution density. For a chromosome in front ψ_k the crowding distance can be determined using the following procedure.

- For an individual objective $O_n, \forall n, 1 \leq n \leq N_o$, chromosomes of front ψ_k are arranged in ascending order accordingly. The i^{th} chromosome in the sorted sequence for objective O_n is represented by $Ch_n(i)$.Here, N_o denotes the total number of objectives, and in this work $N_o = 3$. .
- For an individual objective O_n, set the crowding distance of the boundary chromosomes to ∞, i.e., $cd_n(Ch_n(1)) = cd_n(Ch_n(|f_k|)) = \infty$. The crowding distance of the remaining chromosomes, $Ch_n(i), \forall i, 2 \leq i \leq (|\psi_k| - 1)$ is then calculated using the following expression.

$$cd_n(Ch_n(i)) = \frac{O_n(Ch_n(i+1)) - O_n(Ch_n(i-1))}{O_n^{max} - O_n^{min}}, \forall i, 2 \leq i \leq (|\psi_k| - 1) \tag{15}$$

where, O_n^{max} and O_n^{min} are the maximum and minimum value of the objective O_n.

- For a chromosome $Ch(i)$ the overall crowding distance $cd(Ch(i))$ represents the cumulative sum of its crowding distances across all objectives, i.e.,

$$cd(Ch(i)) = \sum_{n=1}^{N_o} cd_n(Ch_n(i)) \tag{16}$$

The new population P_{g+1} of size P, is created, and genetic operators such as selection, crossover, and mutation are applied. Subsequently, non-dominated sorting and crowding distance sorting are used to generate the next generation. The process is iteratively performed until the stopping criterion, defined by a fixed number of generations or iterations, is satisfied.

6. Simulation results

Simulation results were obtained using the improved NSGA-II approach, implemented in C programming and MATLAB R2012b. The experiments ran on a Windows 10 system with an Intel i5 processor, 3.4 GHz CPU, and 16 GB RAM. Two network configurations, Scenario (200×100) and Scenario (300×100), as illustrated in Fig. 6 (a) and (b), were analyzed. Sensor nodes were randomly deployed across the field in both scenarios. The simulations assume an obstacle-free sensing area, with the base station positioned along one of the vertical edges.

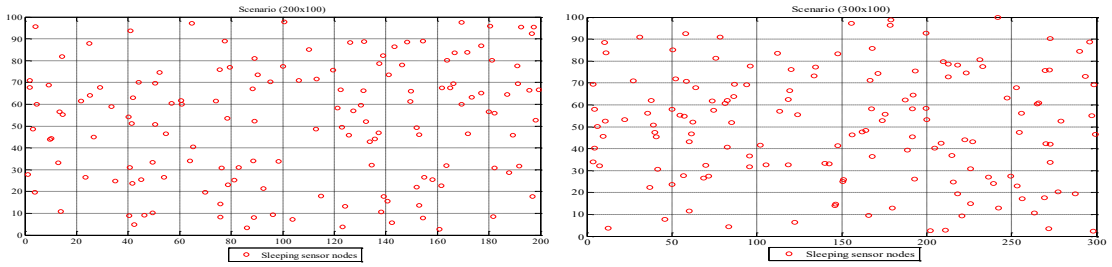
The parameters outlined in Table 1 were used to perform the simulation of the proposed algorithm.

Table 1: Simulation parameters .

Number of deployed sensor nodes	150-300
Initial population	60
Mutation rate	5%
Sensing range	30
Communication range	60

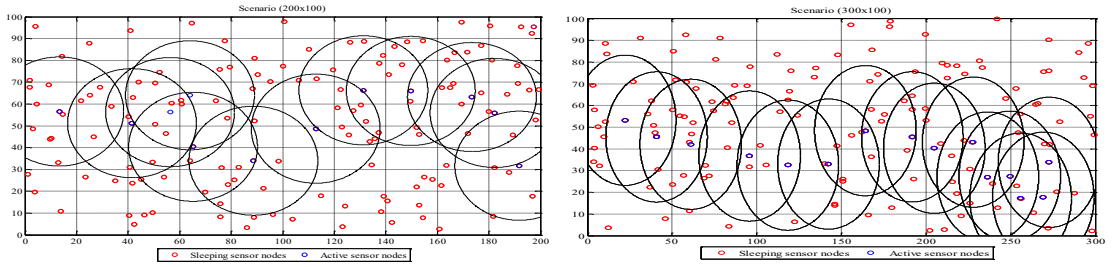
We first executed the proposed improved NSGA-II for the both network scenario Scenario (200×100) and Scenario (300×100) with randomly deployed 300 sensor nodes. In the first round, for the Scenario (200×100) proposed algorithm selects 12 sensor nodes whereas for the Scenario (300×100) 15 sensor nodes are selected, which is shown in Fig. 7 (a) and (b). The obtained barrier coverage for the given network scenarios are shown in Fig. 7 (a) and (b).

For the sake of comparison we have also executed the traditional genetic algorithm (GA) [22, 11] and greedy algorithm (Greedy) [23]. We kept the same simulation parameters for all the algorithms. We executed all the algorithms by varying the length of the networks. The number of sensor nodes selected by different algorithms for various area of networks are depicted in Table 2. Additionally, the algorithms were executed by varying the sensor node count in both network scenarios, Scenario (200×100) and Scenario (300×100) to select sensor nodes for formation of barrier coverage for first round as depicted in Fig.8 (a) and (b). Fig.8 (a) and (b) illustrate that as the sensor nodes increases the number of selected sensor nodes selected for formation of barrier



(a) (b)

Fig. 6: Network scenarios, (a) Scenario (200×100) and (b) Scenario (300×100)).



(a) (b)

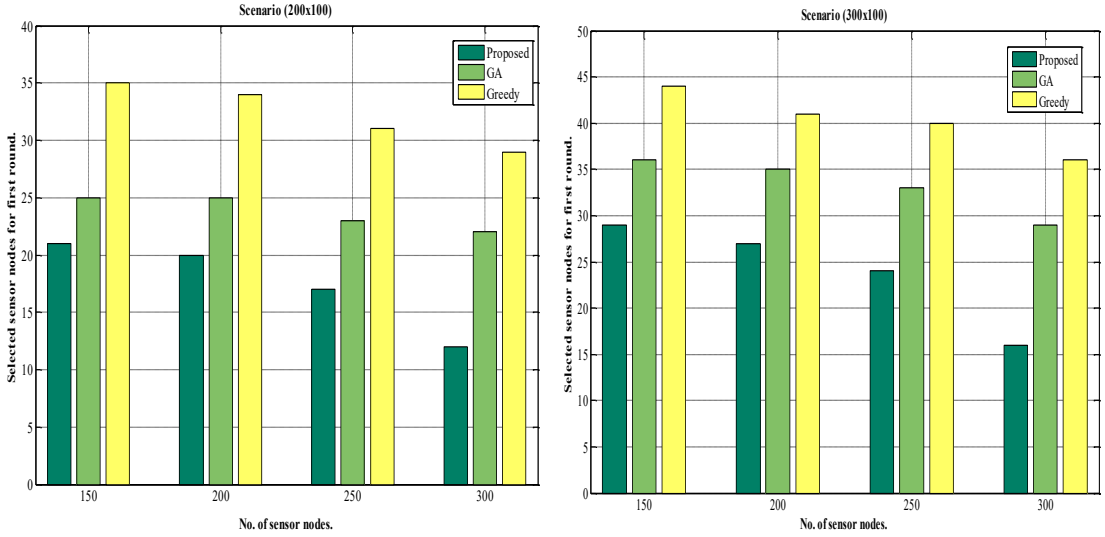
Fig. 7: Selected sensor nodes for network scenarios, (a) Scenario (200×100) with 12 active sensor nodes and (b) Scenario (300×100) with 15 active sensor nodes.

coverage also decreases. The reason is that with an increasing number of sensor nodes in the network, possibility of selecting better sensor nodes for barrier coverage also increases. Similarly, comparison amongst the algorithms for selection of sensor nodes by changing the quantity of sensor nodes is also depicted in Fig.9 (a) and (b). Here also we can observed that with the upsurge of sensor nodes, the number of selected sensor nodes also reduces.

In Fig. 10 (a) and (b), the network life comparison for two scenarios demonstrates the superiority of the proposed algorithm. The first sensor node dies later compared to GA and Greedy, primarily due to lower energy consumption. This is achieved by selecting fewer nodes, which naturally consume less energy than the larger numbers chosen by other algorithms. The proposed method also ensures longer network life by choosing nodes with higher energy levels and limiting the selection count. Nodes with greater energy reserves last longer, delaying the death of the first sensor node relative to GA and Greedy. Also, the limited range helps to reduce data packet forwarding overhead, therefore boosting network efficiency and prolong life.

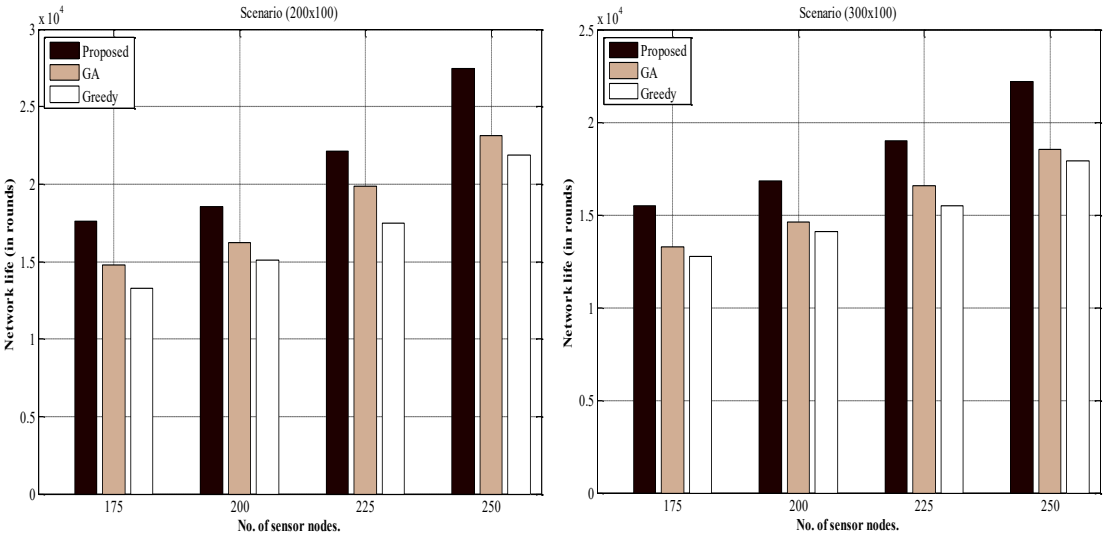
Table 2: Selected sensor nodes by varying the length of the network with randomly deployed 300 sensor nodes.

Area in sq. meter	Selected sensor nodes		
	Proposed	GA	Greedy
200×100	12	14	21
300×100	15	18	24
400×100	22	24	30
500×100	27	29	37



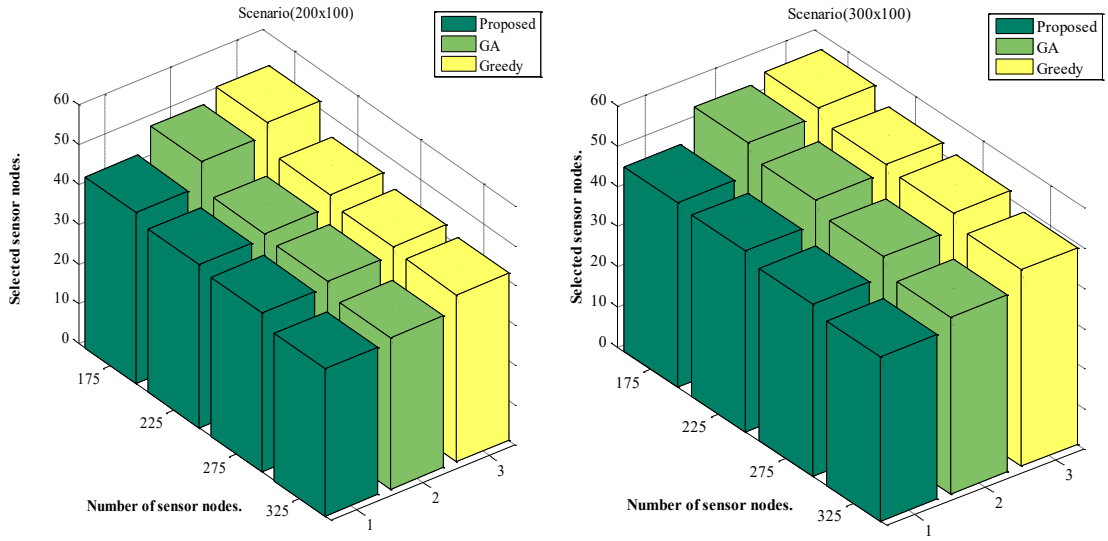
(a) (b)

Fig. 8: Network scenarios of 100×300 with 150 PPs, (a) WSN#1 (b) WSN#1.



(a) (b)

Fig. 10: Network scenarios of 100×200 with 150 PPs, (a) WSN#1 (b) WSN#1.



(a) (b)

Fig. 9: Network scenarios of 100×300 with 150 PPs, (a) WSN#1 (b) WSN#1.

6.1. Analysis of variance (ANOVA)

ANOVA (Analysis of Variance) [24, 25, 26] is a statistical approach designed to analyze variance across multiple samples. Hypothesis testing is used to determine whether the means of two or more samples are equal. The process involves two hypotheses: (i) the null hypothesis (H_{null}) and (ii) the alternative hypothesis (H_{alter}), which are stated as follows.

$$H_{null}: \quad \mu_{\text{Proposed}} = \mu_{\text{GA}} = \mu_{\text{Greedy}}$$
$$H_{alter}: \quad \text{Not all sample means are equal.}$$

The null hypothesis is adopted if the means of the obtained samples are equal. Otherwise, the alternative hypothesis is adopted. The ultimate results of ANOVA include the F-statistic, p-value, and F-critical values. H_{null} is adopted when the F-statistic falls below the F-critical threshold and the p-value surpasses the chosen confidence level (α). Otherwise, the alternative hypothesis (H_{alter}) is adopted.

Table 3: ANOVA input summarized by the count of sensor nodes chosen as active nodes.

Factors	Sample Size	Sample Sum	Sample Mean	Standard Deviation	95% Interval Estimate of the Mean (Lower bound, Upper bound)
Proposed	10	128	12.800	1.14	(11.987, 13.612)
GA	10	157	15.7000	0.823	(15.111, 16.288)
Greedy	10	190	19.0000	0.298	(18.325, 19.674)

In this study, ANOVA test is employed evaluate the significant differences among the outcomes of the proposed algorithm (Proposed), GA, and Greedy algorithms on obtained experimental results, specifically in terms of the number of sensor nodes selected as active nodes. The algorithms were executed under the same environment, with $\alpha = 0.05$ as significance level. Inputs and ANOVA results are summarized in Table 3 and Table 4 respectively. The results depicts that the F-statistic

Table 4: Result of ANOVA test

Source of variations	Sum of Square (SS)	Degree of Freedom (DF)	Mean Square (MS)	F static	p value	F critical
Between Groups	2870.41	1	2870.41	699.02	4.79E-34	4.00
Within Groups	238.167	58	4.107			
Totlal	3108.583	59				

Table 5: Post hoc test (LSD).

Level of difference	Means difference	Standard error	95% Interval Estimate of the Mean (Lower bound, Upper bound)
Proposed-GA	-2.900	0.4363	(-3.795, -2.004)
Proposed-Greedy	-6.200	0.4363	(-7.095, -5.3042)

is greater than the F-critical value, while the observed p-value is less than the confidence level $\alpha = 0.05$. Thus, H_{alter} is adopted rejecting H_{null}.

However, ANOVA does not specify which specific groups among the samples differ significantly. To address this, we performed post-hoc test called Fisher's Least Significant Difference (LSD) to recognize statistically significant variations between the groups. If 95% confidence range for the difference does not includes zero than means of two sample are consider to be different. Table 5 displays the LSD test results. It can be concluded from Table 5 that the means difference between the proposed and related compared algorithms does not have zero value and also the intervals are very far away from zero. Thus we can say that the proposed, GA, and Greedy algorithms are statistically significant in terms of activation of sensor nodes during each scheduling, as depicted by post hoc test called LSD.

7. Conclusion

In this paper, we present an enhanced version of NSGA-II to schedule minimum number of sensor nodes from the given sensor to form an energy efficient barrier coverage. To address the problem we, formulate a linear programming model. Chromosomes are efficiently structured to guarantee the NSGA-II operations, selection, crossover and mutation. All objective functions are derived for an chromosome evaluation. A key improvement in our proposed NSGA-II is an introduction of a novel mutation operation in regard of traditional mutation. Proposed novel mutation operation eliminates redundant active sensor nodes from barrier while ensuring the network functionality as intact. The different network scenarios are thoroughly simulated. The results acquired demonstrates that our proposed approach outperforms existing related algorithms. A statistical ANOVA test followed by post hoc analysis test is also conducted to show superiority of our approach statistically.

Declarations

Competing Interests: All authors declare that they have no conflicts of interest.
Funding Information: Not Applicable
Author Contribution: Not Applicable
Data Availability Statement: Not Applicable
Research Involving Human/ or Animals: Not Applicable

References

[1] S. Kumar, T. H. Lai, A. Arora, Barrier coverage with wireless sensors, in: Proceedings of the 11th annual international conference on Mobile computing and networking, 2005, pp. 284–298.

[2] S. Harizan, P. Kuila, R. Kumar, A. Khare, R. Clonia, A. Perti, Improved genetic algorithm-based sensor nodes deployment for barrier coverage, International Journal of Sensor Networks 43 (3) (2023) 146–157.

[3] W. Wang, H. Huang, Q. Li, F. He, C. Sha, Generalized intrusion detection mechanism for empowered intruders in wireless sensor networks, IEEE Access 8 (2020) 25170–25183.

[4] S. A. Elsaid, N. S. Albatati, An optimized collaborative intrusion detection system for wireless sensor networks, Soft Computing 24 (16) (2020) 12553–12567.

[5] L. Zhiqiang, G. Mohiuddin, Z. Jiangbin, M. Asim, W. Sifei, Intrusion detection in wireless sensor network using enhanced empirical based component analysis, Future Generation Computer Systems 135 (2022) 181–193.

[6] T. Zhang, D. Han, M. D. Marino, L. Wang, K.-C. Li, An evolutionary-based approach for low-complexity intrusion detection in wireless sensor networks, Wireless Personal Communications (2022) 1–24.

[7] A. Thakkar, R. Lohiya, Role of swarm and evolutionary algorithms for intrusion detection system: A survey, Swarm and evolutionary computation 53 (2020) 100631.

[8] S. Harizan, P. Kuila, Evolutionary algorithms for coverage and connectivity problems in wireless sensor networks: a study, Design frameworks for wireless networks (2020) 257–280.

[9] S. Harizan, P. Kuila, Nature-inspired algorithms for k-coverage and m-connectivity problems in wireless sensor networks, Design frameworks for wireless networks (2020) 281–301.

[10] H. Sadia, S. Farhan, Y. U. Haq, R. Sana, T. Mahmood, S. A. O. Bahaj, A. Rehman, Intrusion detection system for wireless sensor networks: A machine learning based approach, IEEE Access.

[11] S. Harizan, P. Kuila, Coverage and connectivity aware energy efficient scheduling in target based wireless sensor networks: An improved genetic algorithm based approach, Wireless Networks 25 (4) (2019) 1995–2011.

[12] S. Harizan, P. Kuila, A novel nsga-ii for coverage and connectivity aware sensor node scheduling in industrial wireless sensor networks, Digital Signal Processing 105 (2020) 102753.

[13] T. M. Nguyen, H. H.-P. Vo, M. Yoo, Enhancing intrusion detection in wireless sensor networks using a gswo-catboost approach, Sensors 24 (11) (2024) 3339.

[14] S. Harizan, P. Kuila, A. Kumar, A. Khare, H. Choudhary, Multi-objective evolutionary algorithms for coverage and connectivity aware relay node placement in cluster-based wireless sensor networks, Wireless Personal Communications (2024) 1–30.

[15] K. Deb, Multi-objective optimisation using evolutionary algorithms: an introduction, in: Multi-objective evolutionary optimisation for product design and manufacturing, Springer, 2011, pp. 3–34.

[16] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: Nsga-ii, IEEE transactions on evolutionary computation 6 (2) (2002) 182–197.

[17] K. Deb, S. Agrawal, A. Pratap, T. Meyarivan, A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: Nsga-ii, in: Parallel Problem Solving from Nature PPSN VI: 6th International Conference Paris, France, September 18–20, 2000 Proceedings 6, Springer, 2000, pp. 849–858.

[18] J. Amutha, S. Sharma, J. Nagar, Wsn strategies based on sensors, deployment, sensing models, coverage and energy efficiency: Review, approaches and open issues, Wireless Personal Communications 111 (2) (2020) 1089–1115.

[19] M. Radhika, P. Sivakumar, Energy optimized micro genetic algorithm based leach protocol for wsn, Wireless Networks 27 (1) (2021) 27–40.

[20] A. Konak, D. W. Coit, A. E. Smith, Multi-objective optimization using genetic algorithms: A tutorial, Reliability engineering & system safety 91 (9) (2006) 992–1007.

[21] K. Deb, K. Sindhya, J. Hakanen, Multi-objective optimization, in: Decision sciences, CRC Press, 2016, pp. 161–200.

[22] D. E. Goldberg, Genetic algorithms, pearson education India, 2013.

[23] I. Cardei, M. Cardei, Energy-efficient connected-coverage in wireless sensor networks, International Journal of Sensor Networks 3 (3) (2008) 201–210.

[24] L. St, S. Wold, et al., Analysis of variance (anova), Chemometrics and intelligent laboratory systems 6 (4) (1989) 259–272.

[25] J. N. Rouder, C. R. Engelhardt, S. McCabe, R. D. Morey, Model comparison in anova, Psychonomic bulletin & review 23 (2016) 1779–1786.

[26] K. Okoye, S. Hosseini, Analysis of variance (anova) in r: One-way and two-way anova, in: R Programming: Statistical Data Analysis in Research, Springer, 2024, pp. 187–209.