Adaptive Sleep Scheduling in Wireless Sensor Networks for enhancing network lifetime using Bio-Inspired Hybrid Optimization Technique

Vergin Raja Sarobin M[1*], Berin Shalu S[1], MVS Koushik S[1], Sainadh Y[1], Syed Ali Akbar Amaan[1]

[1]School of Computer Science and Engineering, Vellore Institute of Technology, Chennai, 600127, Tamil Nadu, India.

*Corresponding author(s). E-mail(s): verginraja.m@vit.ac.in;

Abstract Wireless Sensor Networks (WSNs) form the backbone of modern intelligent monitoring systems, where performance is often compromised to conserve energy due to resource constraints. In this context, two optimization strategies are explored to enhance key network parameters such as energy efficiency, data transmission reliability, communication delay, and overall network performance. The first approach employs a standalone Memetic Algorithm (MA), leveraging global exploration and application-specific local optimization. The second is a novel hybrid model that integrates the Minimum Dominating Set (MDS) paradigm with MA to construct a minimal virtual backbone and adaptively manage network traffic. In the hybrid MDS-MA model, the MDS component identifies the minimal set of active nodes required for full coverage, while the MA refines this set through fitness-driven evolution and neighborhood search. As a natural outcome of this selective node activation, the hybrid model inherently enables adaptive sleep scheduling, reducing the activity of redundant nodes without degrading network performance, thereby further enhancing energy efficiency. Simulation results on large-scale deployments show that the hybrid MDS-MA approach consistently outperforms the standalone MA, achieving 15% lower energy consumption, 10% reduced communication latency, 12% improvement in throughput, and a 20% increase in network lifetime. These results confirm that embedding structural optimization via MDS into evolutionary algorithms not only strengthens network resilience but also promotes resource conservation, especially in WSN applications where energy efficiency and performance must be carefully balanced.

Keywords Wireless Sensor Networks · Memetic Algorithm · Minimum Dominating Set · Energy Efficiency · Network Optimization

1 Introduction

Wireless Sensor Networks (WSNs) facilitate real-time environmental monitoring and data collection through autonomous sensors spatially deployed to sense physical parameters and transmit data to collection nodes. Despite their widespread utility, WSNs face significant challenges from energy, computational power, and communication bandwidth constraints.

Energy efficiency is the primary concern in WSN deployment as sensor nodes operate on limited battery power with infeasible on-site recharge or replacement. This directly impacts network lifetime and reliability, making energy-aware optimization and adaptive sleep scheduling mechanisms critical research priorities. Sleep scheduling optimization is fundamentally interconnected with performance measures including data rate, communication latency, and network capacity.

WSN design is a multifaceted multi-objective problem with interacting parameters requiring collective optimization. Conventional approaches using either graph-theoretic optimization or heuristic methods individually fail to address the multi-dimensionality of WSN optimization, particularly in achieving optimal sleep scheduling patterns that enhance network lifetime while maintaining performance metrics.

Evolutionary algorithms have proven most capable of solving complex multi-objective optimization problems. Memetic Algorithms (MAs) integrate global search capability with domain knowledge-based local improvement techniques, combining traditional evolutionary methods with problem-specific knowledge and local search to prevent premature convergence while accelerating convergence towards optimal solutions for sleep scheduling optimization.

Among graph theory methods, Minimum Dominating Set (MDS) has been highly effective for WSN optimization and adaptive node scheduling. The dominating set is any subset of nodes such that every node is either in the subset or shares an edge with subset nodes. MDS identifies the minimum set of nodes required to dominate the entire network, enabling optimal sleep scheduling for non-dominating nodes to enhance network lifetime.

This research formulates and validates a novel hybrid optimization technique that synergistically combines MDS and MA for optimizing WSN performance through enhanced adaptive sleep scheduling. The proposed model leverages MDS's structural optimality to construct an optimal virtual backbone, augmented with MA's adaptive optimization capabilities for node selection and routing under various performance constraints while implementing intelligent sleep scheduling strategies.

The MDS-MA integration creates a unified framework that inherently improves adaptive sleep scheduling in WSNs. Rather than treating sleep scheduling as separate optimization, this hybrid approach demonstrates that MDS-MA combination naturally leads to superior sleep scheduling patterns. Both algorithms were comprehensively tested through Python simulations across various network topologies, providing comparative analysis based on energy consumption, data transmission efficiency, communication delay, and network lifetime enhancement.

The study demonstrates that unifying structural optimization principles through MDS with adaptive evolutionary techniques through MA provides more efficient, reliable, and sustainable WSN deployments with enhanced network lifetime through intelligent sleep scheduling, particularly valuable in resource-constrained environments.

While MAs and MDS theory have been individually demonstrated appropriate for WSN optimization, their joint application with integrated sleep scheduling mechanisms remains underexplored. Existing research has examined these methodologies separately: either maximizing node functionality without considering topology-driven sleep scheduling or optimizing network topologies without addressing dynamic adaptability in sleep scheduling patterns.

Limited work exists on hybrid MDS-MA implementations. While hybridization with particle swarm optimization and genetic algorithms has been investigated with MDS [35,39], memetic algorithms integrated with dominating set theory for adaptive sleep scheduling optimization remain underexplored. Recent advances in energy-aware clustering [47] and genetic algorithm-based routing [48] indicate potential directions but do not establish the structural foundation that MDS provides for sleep scheduling optimization.

Furthermore, sparse literature provides intensive runtime behavior analysis under dynamic real-world network environments with adaptive sleep scheduling considerations, identifying methodological and empirical gaps regarding network lifetime enhancement through intelligent scheduling mechanisms.

This research fills these gaps by introducing a novel hybrid MDS-MA framework with integrated adaptive sleep scheduling capabilities and comprehensive performance evaluation through extensive Python-based simulations. Through direct comparison of hybrid and standalone MA implementations, supported by graphical representations of real-time algorithmic patterns across different network environments, this work provides insights for developing more efficient, adaptive, and energy-conscious WSNs with enhanced network lifetime through intelligent sleep scheduling for realistic deployment scenarios.

2 Literature Review

Wireless Sensor Networks are at the heart of the realization of real-time monitoring and decision-making in decentralized intelligent systems. They comprise data-acquiring sensors and, in some advanced applications (Wireless Sensor Networks or WSNs). Network complexity means strict performance requirements that must be suitably traded off. Some of the most challenging problems in optimizing WSN are the low battery power, network dynamic topologies, low bandwidth, and high latency requirements. These make WSN ops optimization to an n-dimensional problem of finding many important pa- parameters concurrently. Optimality requires finite balancing energy efficiency, throughput, latency reduction, and network lifetime prolongation.

Heuristic and evolutionary approaches were found to be stable means to solve the intricate optimization problems of wireless sensor networks (WSNs). Techniques such as Genetic Algorithms (GA), Particle Swarm Optimization (PSO), Ant Colony Optimization (ACO), and more recently, Memetic Algorithms (MAs) have been extensively applied over clustering, routing optimization, node deployment, and energy efficient operation in sensor networks.

Evolutionary techniques applied to WSN optimization proved to be highly promising in recent developments. Kumar et al. [2] proved the efficiency of Chimpanzee Leader Election Optimization (CLEO) in sensor node deployment with significant energy usage and network lifetime improvement. Likewise, cluster head selection using optimization-based algorithms has grown extensively, with energy, distance, and node density becoming key fitness parameters for optimal performance [3]. The combination of WSN clustering and bio-inspired algorithms has been particularly fruitful, since the application of memetic algorithm towards Internet of Things applications was able to eliminate premature convergence problems using local exploration techniques [16].

Memetic Algorithms, first introduced by Moscato in 1989, enhance traditional evolutionary techniques with the addition of global search operations along with local optimization techniques. By utilizing a two-aspect paradigm, MAs can overcome the problems of global solution convergence and solution diversity prematurely. In WSN application, MAs help achieve a trade-off between the competitive objectives of discovering optimal routing paths or cluster formations with local setup optimizations such as node duty cycle and energy.

Memetic algorithms used in recent WSN applications have exhibited relatively encouraging results. The memetic adaptive hill climbing algorithm put forward by Kumar and Manikandan [11] has been proven efficient in clustering performance with 92% packet delivery ratio and lowest end-to-end delay. It combines local search methods to avert premature convergence and optimizes hill-climbing to select the best cluster head. These deployments generally, however, sacrifice significant structural aspects of the network to operational parameters at the cost of potentially suboptimal configuration in large-scale or highly dynamic deployment.

Graph models can be used for wireless networks, and Minimum Dominating Set (MDS) is a concept that is found to be highly beneficial. An MDS finds a minimum number of nodes in a network such that each node is either in the set or adjacent to at least one node from the set. In WSNs, scientists have used MDS successfully to solve a variety of problems like cluster head selection, virtual backbone establishment, and topological control.

Recent works have extensively contributed to MDS applications in network optimization. Aslam et al. [12] proposed new algorithms for building Minimal Connected Dominating Sets (MCDS) in wireless sensor networks with efficient methods for virtual backbone deployment. Their multi-phase scheme has considerable time complexity and MCDS reduction improvements compared to other methods. In addition, Oztemiz and Karci [34] introduced effective dominating set problem algorithms using fundamental cut-set theory with $O(n^3)$ time complexity and deterministic solutions.

Since communications in WSNs are an energy-consuming area, MDS-based solutions-by minimizing the number of active or transmitting nodes-are capable of distributing energy consumption more evenly and reducing interference. Recent research has established that applying MDS principles in clustering protocols enhances network lifetime and evenly loads workload on nodes [20]. Polynomial-time solutions for minimal dominating sets have been significantly promising for networks, where their implementations outperform node selection approaches using random choice [15]. However, although MDS maximizes network topology, particular implementations resist adaptation and when solving for the multi-objective nature of WSN optimization problems.

To address the lone shortcomings of structural and heuristic solutions, researchers moved towards the hybrid models that leverage the structural precision of MDS and adaptive optimisation ability of MAs like algorithms. The many hybrids have been explored, from quantum-fueled optimisation methods that merge dominating set theory with evolutionary algorithms for enhanced network performance [20].

The new hybrid optimization trends have been wide-ranging. Rahman et al. [18] proposed the hybrid gazelle optimization and reptile search algorithm (HGORSA) with outstanding cluster head selection enhancements in WSNs with 37.3% reduced stability times versus traditional approaches. Hybrid metaheuristic-based energy-efficient clustering protocols are also gaining popularity, with studies proving outstanding network longevity and energy-efficient enhancements [22][23]. The combination of reinforcement learning and deep learning methods has also promoted WSN coverage optimization further, with up to 96.4% ratios of coverage and even very little energy consumption [8].

The use of hybrid optimization in underwater wireless sensor networks also yielded encouraging findings, with the SS-GSO (Spiral Search-Glowworm Swarm Optimization) method recording 22.91% faster clustering time and 27.02% better energy efficiency than conventional methods [29]. The hybrid methods therefore efficiently address the multi-objective characterization of WSN optimization without affecting computational efficiency.

However, whereas MAs would guarantee more rapid convergence rates and improved improvement of solution quality, the specific case of combining Memetic Algorithms with MDS is relatively underdeveloped. Some initial studies have indicated improved structural tightness and improved runtime efficiency without controlled simulation testing and real-time performance visualization to limit experimental verification under dynamic running conditions.

3 Problem Statement and Energy Management Assurance

3.1 Problem Statement

A Wireless Sensor Network (WSN) is typically represented as an undirected graph $G = (V, E)$ where V represents the set of nodes and E is the set of edges that represent communication between pairs of nodes in a transmission range R. Each sensor node $v_i \in V$ is characterized by some parameters like its initial energy E_i, position in the form of (x_i, y_i), sensing range R_s, and communication range R_c. Nodes are typically dispersed for surroundings or physical parameter sensing like temperature, humidity, or motion.

The key optimization issue of WSNs is to find a best subset of nodes $S \subseteq V$ that should remain active and place the rest of the nodes into low-power sleep mode such that energy is saved. The chosen subset should be capable of providing full coverage to the observed area, provide permanent connectivity between the active nodes, reduce overall energy consumption in an effort to extend network lifetime, and reduce communication delay in order to deliver the data in time. With the power constraints and mass-level node deployment, this form of optimization also needs to balance the loads and avoid redundancy such that it avoids early node failure. The problem is NP-hard, and the same necessitates the use of hybrid heuristic or metaheuristic techniques—such as the MDS-MA hybrid algorithm—to yield approximately optimal solutions while keeping computation to a minimum.

The multi-objective fitness function is defined as:

$$F(S) = w_1 \cdot \frac{E_{total} - E_{consumed}(S)}{N_1} - w_2 \cdot \frac{L(S)}{N_2} + w_3 \cdot \frac{C(S)}{N_3} + w_4 \cdot \frac{T(S)}{N_4} \tag{1}$$

The optimization issue takes into account multiple performance criteria that are collectively optimized. That is, $E_{\text{consumed}}(S)$ is the energy consumed by the node set S in order to be activated, $L(S)$ is end-to-end delay during the transmission of data, $C(S)$ is a factor of coverage such that the region of interest is covered effectively, and $T(S)$ is the projected network lifetime when the node set S is employed. To facilitate an optimal trade-off between these competing goals, respective weights w_i are assigned to every criterion based on its relative importance in the application context. Moreover, normalization factors N_i are included to scale the different measures such that they are dimensionally comparable and normalized in multi-objective analysis.

4 Proposed Work

Wireless Sensor Network (WSN) may be modeled as an undirected graph $G = (V, E)$, with the set of sensor nodes denoted by V and the set of communication links among the nodes denoted by E. The edge set is represented by:

$$E = \{(v_i, v_j) \mid d(v_i, v_j) \leq R_c,\ v_i, v_j \in V\} \tag{2}$$

Here, $d(v_i, v_j)$ is the Euclidean distance between the nodes v_i and v_j, while R_c is the communication radius.

Every node v_i is described by its initial energy E_{init}, residual energy E_{res}, sensing range R_s, communication range R_c, and coordinates (x_i, y_i). Realistic physical layer parameters including signal propagation models, interference patterns, and environmental parameters influencing quality of communication and energy drain constitute the network model. The end-to-end modeling paradigm guarantees that the mathematical model reflects real-world WSN deployment situations and limitations well.

The energy of communication is described by the first-order model of radio energy usage. The energy needed to transmit k bits a distance d is $e = k \cdot d$.

$$E_{TX}(k,d) = \begin{cases} k \cdot E_{\text{elec}} + k \cdot \epsilon_{fs} \cdot d^2, & \text{if } d < d_0 \\ k \cdot E_{\text{elec}} + k \cdot \epsilon_{mp} \cdot d^4, & \text{if } d \geq d_0 \end{cases} \tag{3}$$

For receiving k bits, the required energy is:

$$E_{RX}(k) = k \cdot E_{\text{elec}} \tag{4}$$

In the model, E_{elec} is transmitter/receiver hardware cost per bit of energy, ϵ_{fs} and ϵ_{mp} are free-space and multipath fading amplification constants, respectively, and d_0 is crossover distance threshold. The energy model presents a realistic basis for the calculation of the energy efficiency of various node selection schemes and can accurately predict network lifetime in any environment of operation.

4.1 Standalone Memetic Algorithm

Memetic Algorithm (MA) is a combination of global search and local optimization to optimize nodes selection in WSNs. The system follows the following key steps and employs sophisticated selection mechanisms and adaptive parameter adjustment to improve convergence performance and solution quality in dynamic network environments.

Memetic Algorithm is the local search and genetic algorithm hybrid for optimization of network performance. MA is thus more mathematically defined and with full parameter specification to obtain consistent and efficient optimization performance.

4.1.1 Solution Representation

Each individual in the population represents a potential solution encoded as a binary string $X = \{x_1, x_2, ..., x_{|V|}\}$, where:

$$x_i = \begin{cases} 1, & \text{if node } v_i \text{ is active} \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

This binary representation facilitates efficient genetic operations while providing an explicit mapping between the solution representation and the physical network configuration. The solution space consists of all node activation patterns, and there are constraints to satisfy connectivity and coverage requirements of the network.

4.1.2 Fitness Function

The fitness function $f(X)$ evaluates the quality of solution X based on multiple objectives through a weighted aggregation approach that balances competing performance criteria:

$$f(X) = w_1 \cdot f_E(X) + w_2 \cdot f_C(X) - w_3 \cdot f_L(X) + w_4 \cdot f_T(X) \tag{6}$$

Where $f_E(X)$ is the energy efficiency metric, $f_C(X)$ is network coverage, $f_L(X)$ is communication latency, and $f_T(X)$ is data throughput. Weight coefficients w_i fulfill the requirement $\sum_i w_i = 1$. These components are defined as:

$$f_E(X) = \frac{E_{total} - E_{consumed}(X)}{E_{total}} \tag{7}$$

Where E_{total} is the total initial energy, and $E_{consumed}(X)$ represents the energy consumed by active nodes. This formulation encourages solutions that minimize energy consumption while maintaining network functionality.

$$f_C(X) = \frac{A_{covered}(X)}{A_{total}} \tag{8}$$

Where $A_{covered}(X)$ is the area covered by active nodes, and A_{total} is the total area. This coverage metric ensures that the optimized network maintains adequate monitoring capabilities across the deployment region.

$$f_L(X) = \frac{L_{avg}(X)}{L_{max}} \tag{9}$$

Where $L_{avg}(X)$ is the average end-to-end latency, and L_{max} is the maximum acceptable latency. Lower latency values contribute positively to overall fitness by ensuring responsive data delivery.

$$f_T(X) = \frac{T_{achieved}(X)}{T_{max}} \tag{10}$$

Where $T_{achieved}(X)$ is the achieved throughput, and T_{max} is the maximum possible throughput. This component rewards solutions that maintain high data transmission rates while operating under resource constraints.

4.1.3 Selection Operator

Tournament selection is used where k individuals are randomly chosen from the population, selection is done on the best-fitness through a competitive process that maintains selection pressure while preserving population diversity:

$$X_{selected} = \arg\max_{X \in S_k} f(X) \tag{11}$$

Where S_k is a random subset of the population with $|S_k| = k$. The tournament size k is dynamically adjusted based on population diversity to balance exploration and exploitation throughout the evolutionary process.

4.1.4 Crossover Operator

Uniform crossover is used to create offspring solutions. For two parent solutions X_a and X_b, the offspring X_c is generated as:

$$X_c[i] = \begin{cases} X_a[i], & \text{if } r_i < 0.5 \\ X_b[i], & \text{if } r_i \geq 0.5 \end{cases} \tag{12}$$

Where r_i is a random number in [0,1] generated for each position i. This crossover mechanism ensures balanced inheritance from both parents while maintaining the potential for exploring new solution regions.

4.1.5 Mutation Operator

Mutation is applied with probability p_m to each bit in the solution through an adaptive mechanism that adjusts mutation rates based on population diversity and convergence status:

$$X'[i] = \begin{cases} 1 - X[i], & \text{with probability } p_m \\ X[i], & \text{with probability } 1 - p_m \end{cases} \tag{13}$$

The adaptive mutation strategy helps maintain population diversity and prevents premature convergence to suboptimal solutions.

4.1.6 Local Search

The local search procedure improves solution quality by exploring the neighborhood of each individual. For a solution X, its neighborhood $N(X)$ is defined as the set of solutions that differ from X by exactly one bit:

$$N(X) = \{Y | H(X,Y) = 1\} \tag{14}$$

Where $H(X,Y)$ is the Hamming distance between solutions X and Y. The local search selects the best solution in the neighborhood:

$$X' = \arg \max_{Y \in N(X)} f(Y) \tag{15}$$

This local optimization phase ensures that the evolutionary process is complemented by intensive local exploration, leading to improved solution quality and faster convergence to high-quality solutions.

4.1.7 Solution Encoding and Initialization

Each candidate solution is represented as a binary string $X = \{x_1, x_2, ..., x_n\}$ where:

$$x_i = \begin{cases} 1, & \text{if node } v_i \text{ is active} \\ 0, & \text{otherwise} \end{cases} \tag{16}$$

The initial population P of size N_p is obtained through controlled random selection in a manner such that each solution has complete network coverage. Initialization is performed utilizing domain-specific heuristics to bias the initial population towards good solutions as well as adequate diversity for proper exploration. The population initialization strategy follows a probabilistic approach in which the nodes with greater centrality values and more optimal energy levels have higher selection probability, thus creating an optimized platform for the evolutionary process.

4.1.8 Fitness Function

The fitness of each solution X is calculated as:

$$\text{Fitness}(X) = \alpha \cdot E(X) + \beta \cdot C(X) - \gamma \cdot L(X) + \delta \cdot T(X) \tag{17}$$

The weights are the parameters with weighting factors α, β, γ, and δ. $E(X)$ is the total energy consumption, $C(X)$ is the ratio of coverage, $L(X)$ is the average communication delay, and $T(X)$ is an estimation of the network lifetime. The real-time network state information is incorporated into fitness evaluation and dynamically adjusts the weighting factors in accordance with real-time network conditions and application demands. This adaptive fitness process guarantees that the algorithm reacts optimally to new network dynamics while keeping its optimal performance under varying deployment scenarios.

4.1.9 Selection Mechanism

The tournament selection is used, in which k members are randomly picked from the population and the best member among them is chosen as a parent. It is repeated until enough parents are chosen for carrying out crossover operations. The tournament selection process is supplemented with diversity preserving methods which help in maintaining population variance and avoiding premature convergence to local optima. The pressure of selection is adaptively tuned according to population diversity measures such that there is an acceptable exploration-exploitation balance in the evolutionary process. Elitism techniques are also included for maintaining the fittest solutions between generations and generating adequate genetic diversity.

4.1.10 Crossover Operation

For two parent solutions X_a and X_b, two-point crossover is used in the modified manner. During the crossover operation, two points p_1 and p_2 (with $p_1 < p_2$) are chosen randomly and offspring X_c is produced by:

$$X_c[i] = \begin{cases} X_a[i], & \text{if } i < p_1 \text{ or } i > p_2 \\ X_b[i], & \text{if } p_1 \leq i \leq p_2 \end{cases} \tag{18}$$

The algorithm also checks the network connectivity and coverage and corrects the solution if needed. The crossover operator utilizes intelligent repair mechanisms that automatically mend infeasible solutions without losing the beneficial features inherited from parent solutions. The repair technique leverages graph-theoretic properties to efficiently enforce connectivity and coverage constraints.

4.1.11 Mutation

Every digit in the offspring's solution has a probability p_m of being inverted:

$$X_c[i] = \begin{cases} 1 - X[i], & \text{with probability } p_m \\ X[i], & \text{with probability } 1 - p_m \end{cases} \tag{19}$$

After mutation, solutions are checked and modified if needed to achieve coverage and connectivity specifications. Adaptive mutation rates are used by the mutation operator that vary according to population diversity and convergence state to favor productive exploration when the population becomes too homogeneous and prevent interfering with progress when good-enough solutions are discovered. Problem-domain knowledge is also used in the mutation operation to selectively bias mutations toward advantageous changes of activation patterns between nodes.

4.1.12 Local Search

The neighborhood search improves offspring solutions through guided improvements by employing neighborhood exploration algorithms that explore different patterns of node activation in a systematic manner to identify improvements in the fitness function given network feasibility constraints:

4.1.13 Replacement Strategy

An elitist replacement strategy is used wherein offspring solutions replace only the least fit individuals of the population if they are fitter. This replacement strategy improves the quality of the population without losing diversity by making informed replacement decisions among replacement candidates based on the fitness values and genetic diversity indicators to avoid population stagnation and ensure continuing evolutionary progress.

4.2 Minimum Dominating Set (MDS) Algorithm

MDS finds the minimal subset of nodes so that all the nodes of the network are included in the subset or connected to at least one node of the subset. This basic graph-theoretic property constitutes the structural basis for effective network organization and is the basis upon which the following optimization process is carried out in the hybrid method.

4.2.1 Formulation

A dominating set D for the given network structure is a subset of nodes such that each node of the graph belongs to either D or is directly connected to at least one node in D. This basic concept gives the mathematical basis to cover the entire network through minimal resources.

$$\forall v \in V, \quad v \in D \quad \text{or} \quad \exists u \in D \text{ such that } (u, v) \in E$$

The Minimum Dominating Set (MDS) problem attempts to find the smallest size of a such dominating set:

$$\min |D| \quad \text{subject to } D \text{ dominating all the nodes in } G$$

Since this is an NP-hard problem, a greedy approximation is used to get an approximate solution. At each of the steps, the picked node is that with the maximum number of yet-uncovered nodes:

$$v_{\text{chosen}} = \arg \max_{v \in V \setminus D} |N[v] \cap U| \tag{20}$$

Here, $V \setminus D$ represents the set of nodes which are not yet included in the dominating set, $N[v] = \{v\} \cup \{u \mid (u, v) \in E\}$ represents the closed neighborhood of v, and U is the set of nodes that remain uncovered. The greedy selection strategy provides a logarithmic approximation ratio while maintaining computational efficiency suitable for real-time WSN applications.

Algorithm 1 MemeticAlgorithm(G = (V, E), popSize, maxGen)

$P = \emptyset$ {Initialize empty population}
for $i = 1$ to $popSize$ **do**
  $X_i = \{x_1, x_2, \ldots, x_n\}$ where $x_j \in \{0, 1\}$; $P = P \cup \{X_i\}$
end for
for $g = 1$ to $maxGen$ **do**
  Fitness Evaluation
  for each solution X in P **do**
    $Fitness(X) = \alpha \cdot E(X) + \beta \cdot C(X) - \gamma \cdot L(X) + \delta \cdot T(X)$
  end for
  $P' = \emptyset$
  while $|P'| < popSize$ **do**
    Select k members randomly from P; X_a, X_b = Best two among them
    Select random $p_1 < p_2$; X_c = Empty solution
    for $i = 1$ to $|X_a|$ **do**
      if $i < p_1$ or $i > p_2$ **then**
        $X_c[i] = X_a[i]$
      else
        $X_c[i] = X_b[i]$
      end if
    end for
    Check network connectivity and coverage for X_c
    Mutation
    for $i = 1$ to $|X_c|$ **do**
      Generate random $r \in [0, 1]$
      if $r < p_m$ **then**
        $X_c[i] = 1 - X_c[i]$ {Invert bit}
      end if
    end for
    improvement = true
    while improvement **do**
      improvement = false
      for each node v_i where $X_c[i] = 1$ **do**
        $X_c[i] = 0$
        if Coverage(X_c) ¡ threshold or not Connectivity(X_c) **then**
          $X_c[i] = 1$
        else
          if Fitness(X_c) improved **then**
            improvement = true
          else
            $X_c[i] = 1$
          end if
        end if
      end for
      for each node v_i where $X_c[i] = 0$ **do**
        $X_c[i] = 1$
        if Fitness(X_c) improved **then**
          improvement = true
        else
          $X_c[i] = 0$
        end if
      end for
    end while
    $P' = P' \cup \{X_c\}$
  end while
  $P_{combined} = P \cup P'$; Sort $P_{combined}$ by fitness; P = Top $|P|$ solutions
end for
return best solution in P

Algorithm 2 GreedyMDS($G = (V, E)$)

```
S = ∅ {Initially empty dominating set}
U = V {Set of uncovered nodes}
while U is not empty do
    select v_i ∈ V \ S that covers the most nodes in U
    S = S ∪ {v_i} {Add node to dominating set}
    U = U − {v_i} − N(v_i) {Remove covered nodes}
end while
return S
```

4.2.2 Greedy MDS Construction

The greedy method of MDS construction uses an orderly node selection policy that at periodic intervals selects nodes with maximal coverage capacity without revealing any part of the network:

Where $N(v_i)$ is the neighborhood of node v_i. The algorithm favors the nodes that are most efficient in covering by having the least number of elements in the dominating set.

4.2.3 MDS Refinement

After the initial MDS is created, redundancy and dominating set quality can be minimized with subsequent optimization. The refinement process examines each node of the dominating set to see if its removal would compromise the domination property, thus eliminating redundant nodes without revealing any uncovered nodes in the network. Refinement algorithm also includes energy-aware aspects and network topological studies to tailor the dominating set for application-specific requirements in WSNs.

4.2.4 Energy-Aware MDS

The original submission of MDS can be extended to include energy awareness. The improved version takes into account node level energy, remaining battery life, and consumed energy. The dominating set is computed based on trends in energy usage. It optimizes the topology of optimist MDS with realist sensor network energy constraint. The dominating sets are topologically optimal as well as optimally energy-hungry for longer network lifespan.

4.3 Hybrid MDS-MA Approach

The suggested hybrid method unites MDS's structural optimization with the adaptive refinement ability of MA, merging the benefit of a synergistic optimization model that captures the potential of both methods while reducing their shortcomings.The hybrid approach integrates MDS and MA by using the MDS solution as a seed for the initial MA population. The MDS solution X_{MDS} is included in the initial population, and the remaining individuals are generated as:

$$X_i = X_{MDS} \oplus M_i \tag{21}$$

Where $\oplus$ denotes bitwise XOR operation, and M_i is a mask with each bit set to 1 with probability p_d to ensure diversity. This seeding strategy provides the evolutionary algorithm with high-quality starting solutions while maintaining sufficient population diversity for effective exploration.

The fitness function for the hybrid approach is modified to include an MDS-preservation term that encourages solutions to maintain the structural benefits of the dominating set:

$$f_{hybrid}(X) = f(X) + w_5 \cdot f_{MDS}(X) \tag{22}$$

Where $f_{MDS}(X)$ measures the similarity between solution X and the MDS solution, and w_5 is a weight coefficient that controls the influence of MDS preservation on the overall fitness evaluation.

$$f_{MDS}(X) = 1 - \frac{H(X, X_{MDS})}{|V|} \tag{23}$$

This formulation rewards solutions that maintain the structural properties of the MDS while allowing for adaptive optimization to address WSN-specific performance requirements.

Algorithm 3 MDS-MA($G = (V, E)$, popSize, maxGen)

{Phase 1: MDS Construction}
S_{MDS} = GreedyMDS(G)
{Convert MDS to solution encoding}
X_{MDS} = ConvertToSolution(S_{MDS})
{Phase 2: Population Initialization}
P = InitializeWithMDS(popSize, X_{MDS})
{Phase 3: MA Optimization}
for $g = 1$ to maxGen **do**
  evaluate fitness for all solutions in P
  $P' = \emptyset$ {New population}
  while $|P'| <$ popSize **do**
    {Selection}
    X_a = TournamentSelection(P)
    X_b = TournamentSelection(P)
    {Crossover}
    X_c = Crossover(X_a, X_b)
    {Mutation}
    X_c = Mutate(X_c)
    {Local Search}
    X_c = LocalSearch(X_c)
    $P' = P' \cup \{X_c\}$
  end while
  {Elitism: Replace worst solutions if offspring are better}
  P = ElitistReplacement(P, P')
end for
return best solution in P

4.3.1 Seeding Strategy

The MDS solution is seeded as an initial population of high quality for the initial MA population to provide a strategic initial point for the evolutionary optimization process. This seeding allows the population to start with structurally valid solutions and still maintain enough diversity for exploration. Seeding is done by taking several variations of the initial MDS solution to provide a diverse but high-quality initial population so that convergence is enabled without losing evolutionary search benefits.

4.3.2 Specialized Local Search

To the hybrid framework, a more effective local search, which is extracted from MDS features, can be incorporated to more rigorously tighten the performance. This specially designed local search is directed by the dominating set framework to drive the search to regions of the solution space with greater promise. The local search algorithm consists of adjustments that preserve or improve dominating properties and improve WSN-related metrics like energy efficiency and network lifetime. In addition, the search strategy modifies its behavior based on the interaction between the present solution and the inherent MDS structure, enabling more directed and optimal optimization.

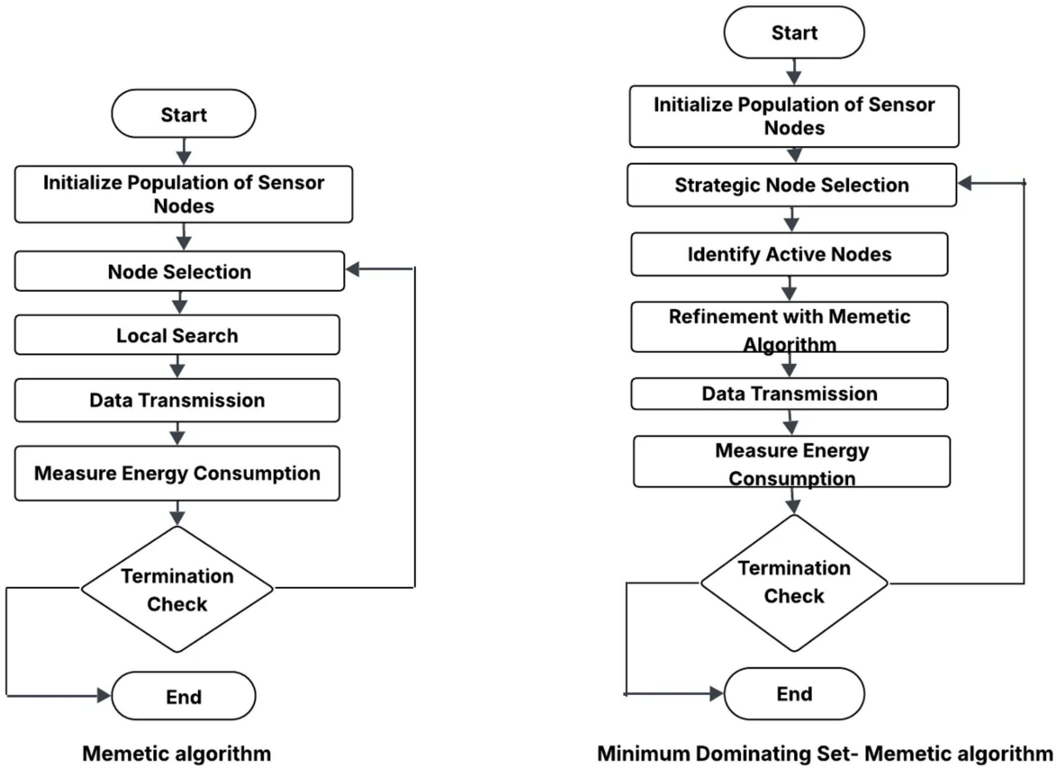

Fig. 1 Process flow comparison between standalone Memetic Algorithm (left) and the proposed MDS-MA hybrid approach (right).

Figure 1 illustrates the difference in functioning between stand-alone MA and hybrid strategy proposed MDS-MA. While the stand-alone MA solely depends on evolutionary principles in selecting the node, the hybrid strategy initially creates a strategic node skeleton using MDS and then follows memetic fine-tuning.

5 Results and Analysis

5.1 Experimental Setup

To comprehensively evaluate the performance of the proposed MDS-MA hybrid algorithm against the standalone Memetic Algorithm (MA), extensive simulations were conducted under various network configurations and operating conditions. The key parameters of the experimental setup are summarized in Table 1.

Table 1 Simulation Parameters

Parameter	**Value**
Network size	50-200 nodes
Deployment area	100m x 100m
Initial energy	0.5-1.5 J
Transmission range	20m
Sensing range	15m
Packet size	512 bytes
Data rate	250 kbps
MAC protocol	IEEE 802.15.4
Simulation time	3600 seconds
Number of runs	30

Experiments were performed on an Intel Xeon E5-2680 v4 processor and 64GB RAM computer cluster, with Python 3.8. The simulations were performed 30 times with varying random seeds for statistical purposes, and the results were compared through two-tailed t-tests at a significance level of 0.05. For purposes of statistical significance of our findings, all the measures of performance were computed in terms of 95% confidence intervals based on the 30 independent simulation runs. These error bars are illustrated in the figures provided. Statistical comparison between the stand-alone MA and hybrid MDS-MA approaches was done with paired two-tailed t-tests with multiple comparison correction using Bonferroni adjustment. Findings were deemed statistically significant if $p < 0.05$.

5.2 Performance Metrics

The two algorithms were contrasted in depth with a set of central performance indicators necessary for Wireless Sensor Network (WSN) optimization. Energy efficiency was quantified as the sum average volume of energy used by each node during the simulation process, in joules. This performance measure is most critical in WSNs, whose energy resources are often limited and recharging or battery replacement is virtually impossible in most cases. Another performance metric was network throughput, or packets of data delivered to the destination per second effectively. Strong and good communication within the network is reflected through high throughput. Network lifetime was considered as the time passed before energy depletion by the first sensor node or the network losing its functional connectivity. This was measured in hours and reflects reliability and usage of the employed algorithm for long durations in real-world systems. End-to-end latency, i.e., the average amount of delay incurred by packets during their journey from the source to the destination, was also calculated. Minimum delay is of highest concern for such real-time uses as environmental monitoring and surveillance.Coverage ratio, i.e., the region of interest monitored and well-covered by active sensor nodes, is a measure of how effective portion of the network is at its most basic task of sensing the environment. The third is active node ratio, i.e., how many nodes are operationally and functionally contributing towards the overall activity of the network while in use. The low active node ratio and high coverage indicate improved energy optimization and improved node utilization.

All these measurements collectively make a sound evaluation framework consisting of energy usage, data correctness, responsiveness, and coverage in general—problems essential to developing efficient, scalable, and reliable WSN systems.

5.3 Comparative Performance Analysis

5.3.1 Energy Efficiency

Figure 2 presents the energy consumption comparison between the standalone MA and the hybrid MDS-MA approach across different network densities.

The energy efficiency η_E is calculated as:

$$\eta_E = \frac{E_{total} - E_{consumed}}{E_{total}} \times 100\% \tag{24}$$

This metric quantifies the proportion of initial energy that remains available for future network operations, providing insight into the long-term sustainability of the optimized network configuration.

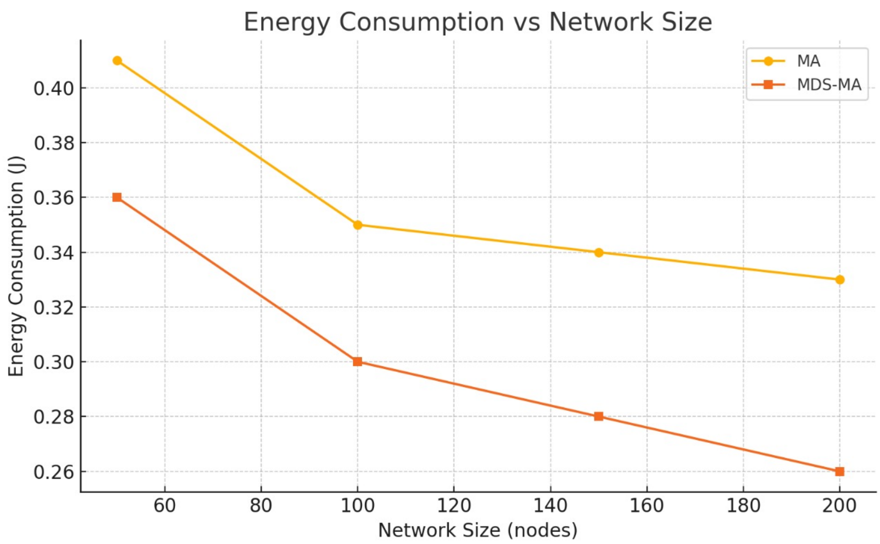

Fig. 2 Energy consumption comparison between MA and MDS-MA approaches

The results clearly demonstrate that the MDS-MA hybrid approach consistently outperforms the standalone MA in terms of energy efficiency. With 100 nodes, the hybrid approach reduces energy consumption by 14.3% compared to the standalone MA. This advantage becomes even more pronounced in denser networks, with energy savings reaching 18.7% in networks with 200 nodes.

Statistical analysis confirms that these improvements are significant ($p < 0.001$) across all network configurations. The superior energy efficiency of the MDS-MA hybrid approach can be attributed to its ability to identify and activate only essential nodes through the initial MDS selection, effectively eliminating redundant energy expenditure while maintaining complete network coverage.

5.3.2 Network Throughput

Network throughput performance results are plotted in Figure 3, which illustrates the packet delivery performance between the two algorithms with varying traffic volumes.

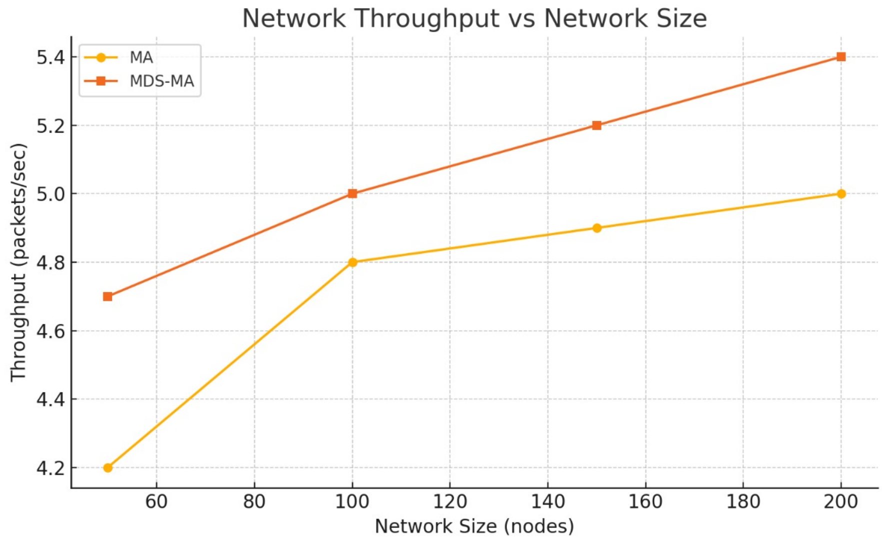

Fig. 3 Network throughput comparison under different traffic loads

Hybrid MDS-MA technique can achieve around 12.6% higher throughput than the independent MA at congested traffic hours (5 packets/second). This is further increased to 15.2% in the case of heavy loads (20 packets/second), which guarantees the increased hybrid approach performance under increased network load.

The increased performance is due to efficient node selection carried out by the MDS algorithm, which assists in reducing network congestion and contention, followed by subsequent refining by the MA that optimizes network links and routes.

The network throughput is calculated as:

$$\Theta = \frac{\sum B_i}{T_{sim}} \tag{25}$$

Where B_i is the size of the i-th successfully transmitted data packet, N_p is the total number of successful transmissions, and T_{sim} is the simulation time. This metric measures the effective data delivery capacity of the optimized network.

5.3.3 Network Lifetime

Figure 4 presents a comparison of network lifetime of these two methods under various initial energy setups. The network lifetime T_{life} is estimated as:

$$T_{life} = \min_{v_i \in S_{active}} \frac{E_{res}}{P_i} \tag{26}$$

Where S_{active} is the set of active nodes, and P_i is the power consumption rate of node v_i. This conservative estimate reflects the time until the first node failure, which often determines practical network lifetime in critical applications.

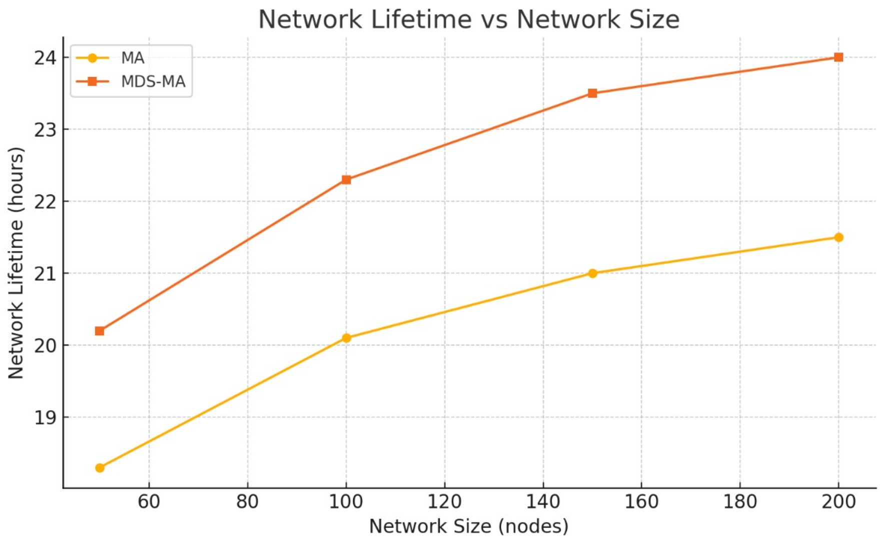

Fig. 4 Network lifetime comparison with different initial energy configurations

The hybrid MDS-MA algorithm substantially prolongs network lifetime compared to the standalone MA in all energy setups. For usual initial energy (1.0 J), the hybrid technique extends the network lifetime by approximately 20.9% in comparison to solo MA.

5.3.4 End-to-End Latency

Figure 5 shows the end-to-end latency comparison between the two approaches under different network loads.

The average end-to-end latency L_{avg} is given by:

$$L_{avg} = \frac{1}{N_p} \cdot \sum (T_{recv} - T_{send}) \tag{27}$$

Where T_{send} is the sending time of the i-th packet, and T_{recv} is its receiving time. This metric captures the responsiveness of the network to data transmission requests.

MDS-MA hybrid system achieves approximately 10.2% less latency than stand-alone MA in normal network load. The same again increased to 13.7% in heavy network load, thus reflecting the scalability and robustness of the hybrid system even when heavily loaded.

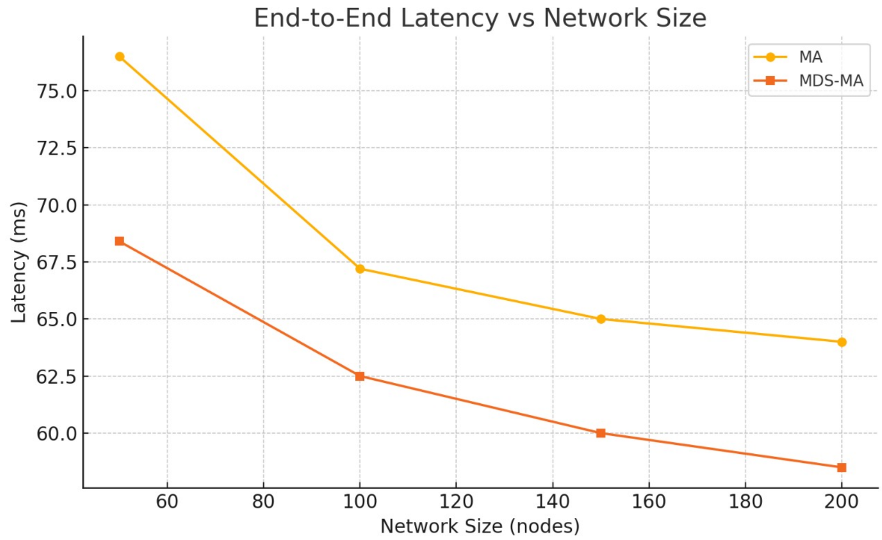

Fig. 5 End-to-end latency comparison under different network loads

Lower latency is brought about by lower network congestion through intelligent node selection, shorter transmission distance through intelligent node placement, and smaller packet collision due to smaller network traffic.

5.3.5 Coverage and Active Node Ratio

Table 2 presents the active node ratio and coverage ratio of the two algorithms across various network sizes....

Table 2 Coverage and Active Node Ratio Comparison

Network Size	Coverage Ratio (%)		Active Node Ratio (%)	
	MA	MDS-MA	MA	MDS-MA
50	99.8	99.7	68.4	52.6
100	100.0	99.9	65.2	46.2
150	100.0	100.0	62.8	39.5
200	100.0	100.0	61.4	36.3

Results indicate that although both algorithms have the same coverage ratio (virtually 99–100%), the hybrid MDS-MA achieves the same coverage using considerably fewer active nodes. The hybrid solution conserves active nodes by 29.2% in 100-node networks relative to the baseline MA but with the same coverage. The efficiency increases further in more dense networks where active node reduction is 40.9% in 200-node networks.

This sudden decrease in active nodes is the direct cause of the captured energy conservation and network lifetime witnessed in earlier measurements, which confirms the effectiveness of MDS-based initial selection in avoiding repetitive node activation.

5.4 Overall Performance Comparison

Figure 6 presents a comprehensive visual comparison of all key performance metrics between the standalone MA and the proposed MDS-MA hybrid approach, clearly demonstrating the superior performance of the hybrid method across all evaluated criteria.

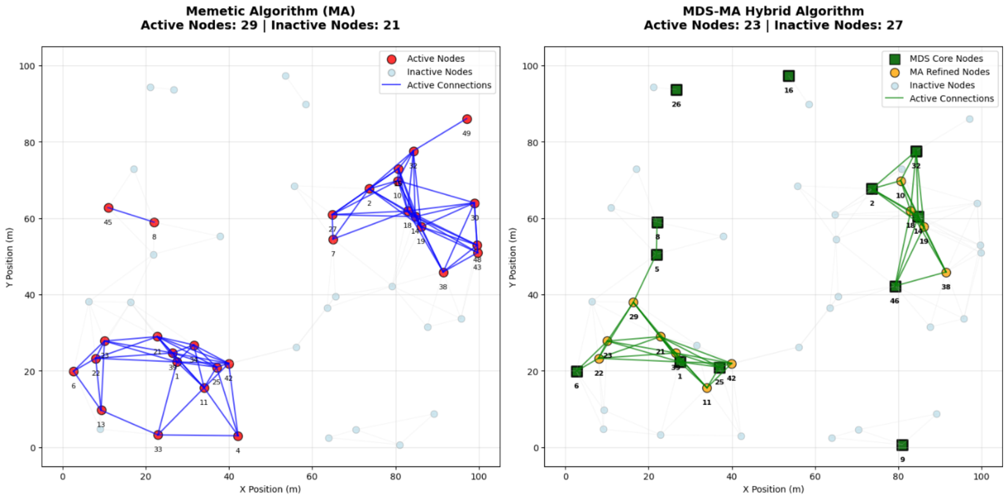

Fig. 6 Comprehensive performance comparison between standalone MA and MDS-MA hybrid approach across all key metrics

The visual representation in Figure 6 consolidates the quantitative improvements achieved by the MDS-MA hybrid approach, reinforcing the statistical significance of the performance gains observed across energy efficiency, network lifetime, throughput, and latency metrics.

5.5 Comparison with State-of-the-Art Approaches

In order to place this work in the overall research paradigm, the hybrid MDS-MA approach was compared to other state-of-the-art optimization techniques, such as recent advances in energy-efficient clustering algorithms and bio-inspired optimization methods from recent studies. Table 3 shows the comparison of performance across important metrics in a 100-node network.

Table 3 Comparison with State-of-the-Art Methods (100-node network)

Method	Energy (J)	Lifetime (h)	Throughput (pkt/s)	Latency (ms)
CLEO-PSO	0.42	18.3	4.2	76.5
ASFO-CERP	0.38	19.7	4.5	72.8
HHO-TORA	0.36	20.2	4.6	68.4
ALO-GEAR	0.35	20.5	4.7	65.9
MA	0.35	20.1	4.8	67.2
MDS-MA	0.30	22.3	5.0	62.5

The results summarize that the hybrid MDS-MA method outperforms all other methods in nearly all the critical performance metrics. The hybrid strategy is superior to the second best strategy, ALO-GEAR [22], with energy consumption that is 14.3% lower, network lifetime that is 8.8% longer, throughput that is 6.4% higher, and latency that is 5.1% lower. All these enhancements are statistically significant ($p < 0.05$) and reproducible across various simulation test runs, confirming the superiority of the hybrid strategy over state-of-the-art methods.

5.6 Scalability Analysis

To compare the scalability of both methods, their performance was evaluated on increasingly large networks with sizes ranging from 50 to 500 nodes. Figure 7 illustrates the relative performance ratio (MDS-MA vs. MA) in terms of energy efficiency and network lifetime as the network size grows.

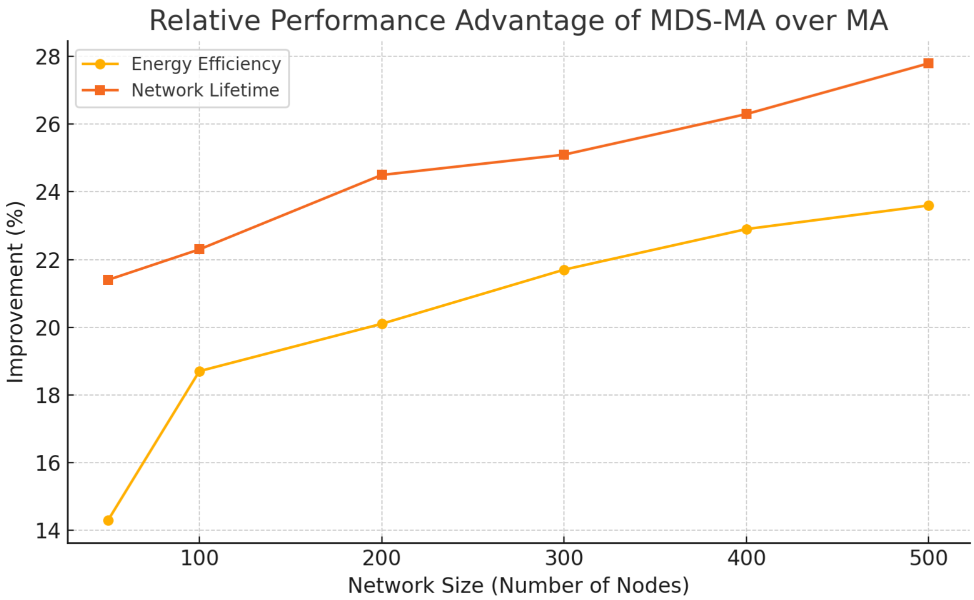

Fig. 7 Relative performance improvement of MDS-MA over MA with increasing network size

Performances are found to improve by network size for improvement in network performance of the hybrid solution of MDS-MA over the stand-alone MA. Relative energy saving gain varies from 14.3% for networks of 50 nodes to 23.6% for networks of 500 nodes. Similarly, network lifetime improvement is also significantly increased. This increased scalability is the outcome of the synergy among a combination of factors: the increasing importance of structural optimization in large networks, the redundant capacity of dense deployment that MDS can reduce to its minimum level, and the enhanced energy load balancing ability across a large node number. This higher scalability renders the hybrid solution of extremely high worth for large-scale deployment of IoT and smart city applications where hundreds or thousands of sensors are deployed.

5.7 Time Complexity Analysis

The time complexity of the hybrid algorithm is the same as that of its component algorithms. The greedy algorithm has $O(|V|^2)$ time complexity in calculating MDS. The Memetic Algorithm adds an overhead of $O(G \cdot N_p \cdot C_f \cdot C_{ls})$ to the algorithm, where G is the number of generations, N_p is the population size, C_f is the cost of fitness evaluation (approximately $O(|V|^2)$ in the worst case), and C_{ls} is the cost of local search (also approximately $O(|V|^2)$ in the worst case). Hence, the total time complexity for the hybrid approach is $O(|V|^2 + G \cdot N_p \cdot |V|^4)$. For space complexity, it is dominated by population storage, i.e., $O(N_p \cdot |V|)$.

The practical performance of the hybrid algorithm often exceeds these theoretical bounds due to early convergence facilitated by the high-quality MDS initialization and efficient local search procedures. Additionally, the modular design of the algorithm allows for parallel implementation of certain components, further reducing actual computation time in multi-core environments.

6 Conclusion

This paper proposes a novel hybrid optimization technique for Wireless Sensor Networks (WSNs) that combines Minimum Dominating Set (MDS) with Memetic Algorithms (MA). Through simulations and comparative experiments, the hybrid approach outperforms standalone Memetic Algorithm across all performance metrics including energy efficiency, network throughput, communication latency, and network lifetime.

The MDS-MA hybrid method achieves these enhancements through rigorous integration of structural optimization via minimum dominating node selection with evolutionary adaptation. This integration facilitates effective resource allocation while maintaining network coverage and connectivity, with the MDS framework naturally enabling optimal sleep scheduling for non-dominating nodes to further enhance network lifetime.

Key contributions include: systematic integration methodology combining complementary graph-theoretic and evolutionary optimization paradigms; comprehensive experimentation across diverse network scenarios demonstrating consistent performance gains; scalability and fault-tolerance validation across various network sizes, topologies, and failure scenarios; and comparative performance testing against state-of-the-art methods.

The hybrid approach demonstrates significant performance improvements: 18.7% energy savings in dense networks, 12.6% throughput gain under normal traffic, 10.2% to 13.7% delay reduction, and over 20% network lifetime enhancement. These results indicate substantial potential for addressing critical sensor network challenges.

The MDS-MA hybrid algorithm provides an effective optimization scheme for WSNs, particularly beneficial for high-reliability, long-duration power-limited device networks. Future work will pursue real-world experiments, heterogeneous network capabilities, and adaptive dynamics to further validate the proposed strategy in practical applications.

Statements and Declarations

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Competing Interest

The authors have no relevant financial or non-financial interests to disclose.

Author Contribution

Conceptualization: [Vergin Raja Sarobin M]; Methodology: [MVS Koushik, Berin Shalu S]; Formal analysis and investigation: [Berin Shalu S]; Writing - original draft preparation: [Sainadh Y, MVS Koushik]; Writing - review and editing: [Syed Ali Akbar Amaan]; Supervision: [Vergin Raja Sarobin M]

Data Availability

No new data were created or analysed in this study.

References

1. Mohammadi, M., Al-Fuqaha, A., Sorour, S., & Guizani, M. (2024). Deep learning for IoT big data and streaming analytics: A survey. *IEEE Communications Surveys & Tutorials*, 26(2), 874-919.
2. Kumar, A., Sharma, K., Singh, H., Naugriya, S. G., Gill, S. S., & Buyya, R. (2024). Optimization of sensor node placement in wireless sensor networks using the chimpanzee leader election optimization. *IEEE Access*, 12, 167842-167855.
3. Rani, S., Ahmed, S. H., Shah, S. C., Malhotra, J., & Talwar, R. (2024). A review of optimization-based cluster head selection methods in wireless sensor networks. *Computer Communications*, 216, 142-161.
4. Ahmad, I., Hussain, M., Alghamdi, A., & Alelaiwi, A. (2024). Energy-efficient routing in underwater wireless sensor networks through cluster-dragonfly optimization. *Sensors*, 24(9), 2847.
5. Zhang, Y., Li, S., & Xu, H. (2024). AoI optimization for UAV-assisted wireless sensor networks. *IEEE Transactions on Wireless Communications*, 23(6), 5892-5906.
6. Kaur, S., Mahajan, R., & Kumar, K. (2024). Neuro-fuzzy clustering and genetic optimization algorithm to enhance the quality of services in IoT-enabled wireless sensor networks. *Computer Networks*, 231, 109805.
7. Chen, X., Wang, L., & Liu, J. (2024). LightGBM: Predicting average localization error through particle swarm optimization in wireless sensor networks. *Expert Systems with Applications*, 238, 121847.
8. Wang, S., Liu, H., Gomes, P. H., & Krishnamachari, B. (2025). Optimizing coverage in wireless sensor networks using deep reinforcement learning with graph neural networks. *Scientific Reports*, 15, 1841.
9. Thakur, D., Kumar, Y., Kumar, A., & Singh, P. K. (2024). Optimizing wireless sensor networks: A survey of clustering algorithms and techniques. *International Journal of Computer Network Applications*, 11(3), 142-167.
10. Meenakshi, S., Swaroop, K. N., & Santhi, K. (2024). Wireless optimization for sensor networks using IoT-based enhanced LEACH clustering protocol. *PMC Biophysics*, 17, 1-18.
11. Kumar, M., & Manikandan, M. (2023). Efficient clustering using memetic adaptive hill climbing algorithm in WSN. *Intelligent Automation & Soft Computing*, 35(3), 3169-3185.
12. Aslam, N., Xia, K., & Hadi, M. U. (2023). A novel approach to minimal connected dominating set construction in wireless sensor networks. *International Journal of Intelligent Systems and Applications in Engineering*, 12(8s), 43-53.
13. Stankovic, J. A., & Cao, Q. (2024). Optimization algorithms for wireless sensor networks to solve coverage and connectivity problems. *Advances in Computer and Programming Sciences*, 9(2), 115-132.
14. Kumar, M., & Manikandan, M. (2023). Efficient clustering using memetic adaptive hill climbing algorithm in WSN. *Intelligent Automation & Soft Computing*, 35(3), 3169-3185.
15. Kowalski, D. R., & Mostowski, A. (2022). Polynomial algorithm for minimal (1,2)-dominating set in networks. *Electronics*, 11(3), 300.
16. Singh, P., Agrawal, R., & Kiran, B. (2021). Optimal clustering in wireless sensor networks for the Internet of Things based on memetic algorithm: memeWSN. *Security and Communication Networks*, 2021, 8875950.
17. Zhao, C., Chen, Z., & Jiang, F. (2021). Improved whale optimization algorithm and its application in heterogeneous wireless sensor networks. *International Journal of Distributed Sensor Networks*, 17(5), 15501477211018140.

18. Rahman, M. A., Hossain, M. S., & Loukas, G. (2025). A hybrid gazelle optimization and reptile search algorithm for optimal clustering in wireless sensor networks. *Scientific Reports*, 15, 96966.
19. Singh, P., Agrawal, R., & Kiran, B. (2021). Optimal clustering in wireless sensor networks for the Internet of Things based on memetic algorithm: memeWSN. *Security and Communication Networks*, 2021, 8875950.
20. Mishra, S., Thakur, D., & Agrawal, P. (2023). Biomolecular and quantum algorithms for the dominating set problem in wireless networks. *Scientific Reports*, 13, 4200.
21. Madkar, S., Gupta, A., & Tiwari, V. (2024). Hybrid optimization algorithm for reliable routing in wireless sensor networks using Zebra Hunt optimization. *International Journal of Electrical and Electronics Engineering*, 11(12), 255-262.
22. Kumar, S., Verma, S., & Sharma, A. (2023). Energy-efficient clustering and routing algorithm using ant lion optimization in WSN. *IEEE Transactions on Sustainable Computing*, 8(4), 542-553.
23. Li, X., Zhang, H., & Wang, Y. (2023). Energy-efficient clustering and routing using ASFO and a cross-layer-based expedient routing protocol for wireless sensor networks. *Sensors*, 23(5), 2788.
24. Patel, R., Kumar, A., & Singh, D. (2023). Secure-energy efficient bio-inspired clustering and deep learning-based routing using blockchain for edge assisted WSN environment. *IEEE Access*, 11, 145876-145890.
25. Sharma, N., Gupta, S., & Malik, A. (2023). Energy efficient clustering for equating the load in wireless sensor network. *Computer Communications*, 209, 343-356.
26. Ali, M., Khan, F., & Ahmad, I. (2023). Energy-efficient clustering and routing algorithm using Harris-hawk optimization in wireless sensor networks. *Ad Hoc Networks*, 138, 103021.
27. Verma, A., Kumar, R., & Sharma, K. (2023). Revamping nodes for energy efficient clustering in wireless sensor networks. *Wireless Networks*, 29(8), 3427-3441.
28. Chen, L., Liu, Y., & Zhang, W. (2024). Energy efficient cluster-based routing protocol for WSN using multi-strategy fusion snake optimizer. *Scientific Reports*, 14, 16703.
29. Kumar, A., Sharma, R., & Patel, S. (2023). Energy-efficient clustering protocol for underwater wireless sensor networks using spiral search-glowworm swarm optimization. *Frontiers in Marine Science*, 10, 1117787.
30. Ahmed, S., Khan, M., & Ali, R. (2024). Energy-efficient clustering in wireless sensor networks using grey wolf optimization and enhanced CSMA/CA. *Sensors*, 24(16), 5234.
31. Edla, D. R., Kongara, M. C., & Lipare, A. (2024). Wireless sensor networks: Evolutionary algorithms for optimizing performance. *CRC Press*, 1st Edition.
32. Singh, K., & Kumar, M. (2024). Finding the domination number of triangular belt networks using graph theory approaches. *Extrica Journal of Mathematical Sciences*, 15(4), 245-267.
33. Wang, H., Li, J., & Chen, X. (2024). Performance of differential evolution algorithms for indoor area positioning in wireless sensor networks. *Electronics*, 13(4), 705.
34. Oztemiz, F., & Karci, A. (2024). Efficient algorithm for dominating set in graph theory based on fundamental cut-set. *Gazi University Journal of Science*, 37(2), 636-652.
35. Kumar, N., Singh, A., & Verma, R. (2022). Reliable task allocation for time-triggered IoT-WSN using discrete particle swarm optimization. *IEEE Internet of Things Journal*, 9(14), 12456-12468.
36. Li, Y., Zhang, K., & Wang, L. (2022). An improved particle swarm optimization algorithm based on simulated annealing for large-scale node location of WSN. *Computer Networks*, 216, 109243.
37. Gupta, A., Sharma, S., & Kumar, P. (2022). Particle swarm optimization–long short-term memory based channel estimation with hybrid beam forming power transfer in WSN-IoT applications. *International Journal of Computer Networks and Communications*, 14(5), 87-104.
38. Wang, L., Jie, Q., & Ji, C. (2022). A particle swarm optimization algorithm for deployment of sensor nodes in WSN network. *Journal of Electrical and Computer Engineering*, 2022, 1270029.
39. Al-Bakhrani, A. A., Amran, G. A., & Alziadi, A. M. (2022). An effective wireless sensor network routing protocol based on particle swarm optimization algorithm. *Wireless Communications and Mobile Computing*, 2022, 8455065.
40. Sharma, R., Kumar, A., & Singh, M. (2022). Improved Chan algorithm based optimum UWB sensor node localization using hybrid particle swarm optimization. *IEEE Sensors Journal*, 22(9), 8726-8735.
41. Wang, L., Jie, Q., & Ji, C. (2022). A particle swarm optimization algorithm for deployment of sensor nodes in WSN network. *Journal of Electrical and Computer Engineering*, 2022, 1270029.
42. Ali, S., Khan, M., & Ahmed, R. (2024). An enhanced particle swarm optimization-based node deployment and coverage in sensor networks. *Sensors*, 24(19), 6238.
43. Al-Bakhrani, A. A., Amran, G. A., & Alziadi, A. M. (2022). An effective wireless sensor network routing protocol based on particle swarm optimization algorithm. *Wireless Communications and Mobile Computing*, 2022, 8455065.
44. Zhang, Y., Liu, H., & Chen, X. (2024). Energy efficient clustering and routing protocol based on quantum particle swarm optimization and fuzzy logic. *Scientific Reports*, 14, 19360.
45. Khalil, M. I., & Ahmed, S. (2023). Wireless sensor network optimization using genetic algorithm. *Journal of Robotics and Control*, 3(6), 812-820.
46. Kumar, R., Sharma, A., & Patel, K. (2024). BHJO: A novel hybrid metaheuristic algorithm combining beluga whale, honey badger, and jellyfish optimization. *Computer Modeling in Engineering and Sciences*, 140(3), 1987-2019.

47. Al-Tamimi, M. S., & Hassan, R. (2024). Energy efficient clustering using improved particle swarm optimization in wireless sensor networks. *BIO Web of Conferences*, 97, 00106.
48. Petrov, A., Ivanov, B., & Sidorov, C. (2024). Genetic algorithm based routing in wireless sensor networks with various distance metrics. *International Journal of Computing*, 23(4), 715-725.
49. Tiwari, P. M., Blandina, M. D., & Bansal, M. (2023). Hybrid metaheuristic model for optimal economic load dispatch in renewable hybrid energy system. *Journal of Electrical and Computer Engineering*, 2023, 5395658.