Power System Protection and Control
ISSN:1674-3415

A NOVEL SWARM INTELLIGENCE-BASED COMPRESSIVE SLEEPING APPROACH TO IMPROVE THE LIFETIME OF WIRELESS SENSOR NETWORKS

Mohammad Hassan Katebi Jahromi[1], Karamollah Bagherifard[2]*, Razieh MalekHoseini[3], Saeed Mehrjoo4

[1]Department of Computer Engineering, Yas.C., Islamic Azad University, Yasuj, Iran, m.katebijahromi@iau.ac.ir
[2]Department of Computer Engineering, Yas.C., Islamic Azad University, Yasuj, Iran, ka.bagherifard@iau.ac.ir
[3]Department of Computer Engineering, Yas.C., Islamic Azad University, Yasuj, Iran, malekhoseini.r@iau.ac.ir
[4]Department of Computer Engineering, Dari.C, Islamic Azad University, Dariun, Iran, saeed.mehrjoo@iau.ac.ir

Received: 25/12/2025 **Revised**: 23/01/2026 **Accepted**: 22/02/2026

***ABSTRACT*:**

Wireless Sensor Networks face serious energy limitations that directly constrain their operational lifetime. To address this, we propose a hybrid optimization framework that integrates compressive sensing, genetic algorithm, and ant colony optimization to decrease energy consumption while preserving data accuracy. The proposed method considers both spatial and temporal correlations but unlike existing approaches that rely on random node activation or single-hop routing, our method centrally selects active nodes using genetic algorithm and constructs a multi-hop data aggregation tree using ant colony optimization for energy-efficient transmission. This design offers a significant improvement over compared methods by jointly optimizing signal reconstruction and routing. Simulation results demonstrate that our method reduces signal reconstruction error by over 48% and extends network lifetime by more than 18% compared to leading baselines.

Keywords*: Accuracy, Compressive Sensing, Data Aggregation, Energy Consumption, Lifetime, Sleep/Wake Algorithm, Wireless Sensor Network.*

INTRODUCTION

Wireless Sensor Network (WSN) consists of compact, energy-constrained sensor nodes deployed across a region to monitor specific conditions and transmit sensed data to a Base Station (BS). WSNs are increasingly employed to monitor real-world phenomena in diverse applications such as water quality monitoring [1], precision agriculture [2], and forest fire detection [3], among others. As data transmission is the most energy-intensive task for sensor nodes, efficient power management is crucial in WSN design.

Consider a WSN in which the data from all n nodes is supposed to be periodically sent to the BS. In the simple method, as shown in Figure 1-A, the data is transmitted without compression, which results in a very high number of transmissions, on the order of n. Moreover, the nodes located near the BS act as bottlenecks, consuming the most energy. As a result, they die quickly, causing the entire network to subsequently fail.

To address energy constraints, reducing the transmission load through data compression is an effective strategy for lowering energy consumption and extending the WSNs' lifetime. However, traditional compression methods, including source and in-network coding often involve intensive computation, rendering them impractical for resource-constrained sensor nodes.

Recently, Compressive Sensing (CS) [4] has emerged as a powerful data acquisition method for WSNs, enabling the reconstruction of physical signals from fewer transmissions by exploiting sparsity and data correlation. It is particularly well-suited to WSNs because a spatial correlation can be observed among the densely deployed sensor nodes in the event region. Also CS imposes minimal computational load on energy-constrained sensor nodes, while shifting the heavy reconstruction burden to the BS, which typically has sufficient energy and processing capability.

 DOI: 10.46121/pspc.54.1.32

Hence, Compressive Data Gathering (CDG) was introduced—an efficient, low-complexity method. One early model, Plain-CDG [5], uses CS and network coding to send a fixed number of packets (M) per node (Figure 1-B, M=3). Later, Hybrid-CDG [6] reduced redundancy by adapting the data relay method based on node degree: low-degree nodes forward raw data, while high-degree nodes apply CS (Figure 1-C).

For WSNs in which there is not only spatial correlation but also temporal correlation among the nodes, it remains feasible to reduce transmissions even more. As stated in [7, 8] and will be explained in more detail in the preliminaries section of this paper, in such networks, it is not necessary for all sensors to participate in every transmission. Activating and transmitting data from only a portion of the nodes per round is sufficient. Other nodes would be in sleep mode to conserve energy. This method named Compressive Sleeping is shown in Figure 1-D.

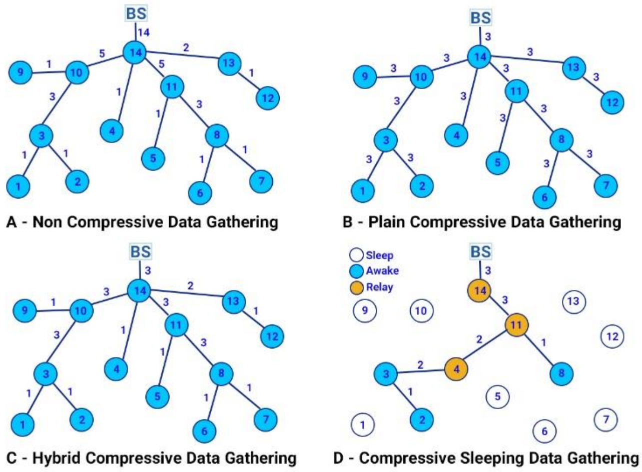

Figure 1 Data Gathering Methods

However, the performance of compressive sleeping method depends heavily on two interrelated factors: (i) the number and identity of active sensor nodes in each round, and (ii) the efficiency of routing the collected data of projection nodes to the BS. Existing approaches often rely on random or probabilistic node activation [7, 9, 10], or assume simplified single-hop routing models [11], both of which are impractical and inefficient in larger-scale deployments.

This paper suggests a centralized optimization technique that integrates Genetic Algorithm (GA) and Ant Colony Optimization (ACO) to jointly optimize node activation and multi-hop data routing. The GA intelligently selects the best subset of nodes by considering spatial-temporal data correlations and residual energy levels. ACO then constructs an efficient Steiner Tree to route the collected data with minimal energy cost.

Unlike prior method [11] that use convex relaxations to simplify node selection and a single hop data transmission, the proposed method directly address the NP-Hard nature of the problem using a GA for optimal node selection and an ACO-based approach for energy-aware routing. This integrated design achieves better trade-offs between signal reconstruction accuracy and network longevity.

To evaluate the effectiveness of the proposed technique, extensive experiments were conducted to analyze the impact of parameters such as sparsity of the data signal and the active nodes number. Additionally, energy consumption and overall network lifetime were examined.
A more comprehensive description of the notations employed in this study is presented in this section:

- Numbers are indicated by lowercase letters.
- Matrices are indicated by bold uppercase letters.
- Column vectors are indicated by bold lowercase letters.
- Sets are indicated by uppercase calligraphic letters.

Other notations:

- $(\cdot)^T$: Transpose of matrix

 DOI: 10.46121/pspc.54.1.32

- $(\cdot)^{-1}$: Inverse of matrix
- $Tr(\cdot)$: Trace of matrix
- x_i: ith element of x
- $X_{i,i}$: ith diagonal element of X
- X_ζ: Submatrix of X, selecting columns indexed by the set ζ
- ζ^c: Complementary set of ζ
- $\mathbb{E}_x(\cdot)$: Expectation with respect to the distribution of the random vector x
- $\binom{n}{m}$: Number of k-combinations of a dataset with n elements
- μ: mean vector
- Σ: covariance matrix
- $N(\mu.\Sigma)$: Multivariate normal distribution
- I_n: Identity matrix of size n
- $\|\cdot\|_i$: l_i norm of vector

The rest of this paper is structured as follows: Section 2 reviews related studies. Section 3 describes the system model along with the compressive sensing procedure. Section 4 explains the proposed GA-ACO optimization framework. Section 5 presents the simulation results, and Section 6 concludes the work.

RELATED WORKS

Extensive research has aimed to enhance the efficiency of CDG. For example, [12] used a multi-objective evolutionary method to jointly optimize the sensing matrix and the number of CS measurements. In [13], The gray wolf optimization algorithm was employed to design a sensing matrix that reduces data reconstruction error. To further cut energy usage, combining compression with sleep scheduling has proven effective. Sleep scheduling helps avoid redundant data collection in densely deployed networks by reducing the number of active nodes without sacrificing coverage or connectivity. It often works in tandem with other techniques to optimize energy use. For instance, [14] proposed a method for opportunistic routing in linear networks, adjusting sleep cycles based on traffic and node energy levels. Likewise, [15] introduced a pairing strategy where nodes take turns sleeping or staying active depending on load and battery status, though this method did not address total transmission load or data recovery performance.

Several studies have proposed combining CS with sleep/wake scheduling to enhance the lifetime of the network [7, 9-11]. The approach used in these studies involves activating only a subset of sensor nodes during each period, while the remaining nodes are turned off to conserve energy. In [7, 9, 10], sensor nodes are randomly turned on with a specified probability in a distributed manner. However, the random selection of active nodes lacks the flexibility to adapt to changes in signal sparsity or channel conditions, which can vary over time. Moreover, randomly generated projection matrices tend to exhibit significantly lower performance compared to optimally designed ones in various applications [11].

The approach presented in [11] centrally determines the awake nodes at the BS for the subsequent round. However, it employs a single-hop transmission scheme, which presents several challenges. First, single-hop transmission is often impractical in real-world WSNs due to potentially long distances between nodes and the BS, which may exceed communication range. Second, such long-range transmissions rapidly deplete node energy, while also increasing packet error and loss rates—negatively impacting both reconstruction accuracy and network lifetime.

Furthermore, [11] utilizes convex relaxation to transform the non-convex problem into a convex formulation. Convex relaxation simplifies complex problems to arrive at a solution. While convex relaxation simplifies complex optimization problems, it does not always guarantee a globally optimal solution. This limitation may lead to suboptimal results or convergence to local minima.

In [10], a CS-integrated sleep scheduling scheme uses reinforcement learning, leveraging a finite MDP and Q-learning for active node selection. To identify optimal decision-making strategies, a model-free Q-learning algorithm is utilized. The Q-learning reward function accounts for both the nodes' residual energy and sampling uniformity, aiming to balance energy usage. The proposed solution in this paper is distributed, which can reduce the cost of control packet transmissions and increase network lifetime. However, this method places less emphasis

DOI: 10.46121/pspc.54.1.32

on the critical trade-off between reconstruction accuracy and network lifetime and lacks an optimal projection matrix construction mechanism.

PRELIMINARIES

This study addresses WSNs made up of numerous sensor nodes spread across a target area to gather relevant data for various applications, including environmental observation, industrial supervision, security systems, and weather tracking. The physical phenomena being monitored often exhibit correlations across both spatial and temporal dimensions.

To implement compressive sleeping approach, the proposed method utilizes the following strategy shown in Figure 2. Given that the signal support changes slowly in the presence of high temporal correlation [16, 17], a GA at the base station (BS) utilizes the support information from the previously reconstructed signal to decide which nodes will be active in the current round. Then after, by finding the active nodes, an optimal data aggregation tree is constructed using an ACO. We refer to this phase as “BS Scheduling & Routing”.

Then, a control packet is transmitted along the constructed tree to inform the selected nodes of their active status. Each node receiving this packet will perform sensing and transmission operations in the current round. The packet also specifies the parent and children of each awake node. This is referred to as the “**Initialized Phase”** in the figure. Next, during the “**CS Sampling Phase”**, awake nodes perform sensing. Finally, in the “**Forwarding Phase”**, data is sent to the BS, which then reconstructs the original signal.

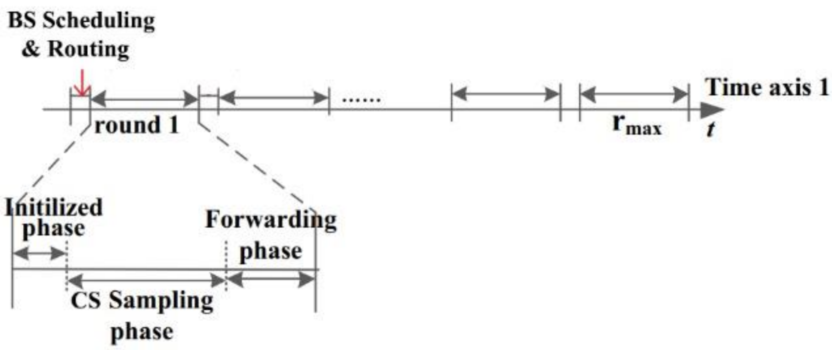

Figure 2. Phases of the proposed method

1.1. Data Collection and Transmission

Let the monitored parameter of each node be denoted by f_i $(i = 1.2. \dots . n)$ which must be reported to the BS. The data collected by different sensor nodes exhibits high spatial correlation. Therefore, the signal $f \in \mathbb{R}^n$ is denoted by sparse vector $x \in \mathbb{R}^n$ under a basis $\Psi \in \mathbb{R}^{n \times n}$:

$$f = \Psi x \quad (1)$$

Ψ which is sparsifying basis is predetermined or adaptively updated using algorithms such as PCA [18] or dictionary learning [19]. A vector x is called **sparse** if only s (where $s \ll n$) of its elements are non-zero while the remaining are zero; in other words, $\|x\|_0 = s$.

In compressive sleeping WSNs, only m ($where$ m < n) sensor nodes are active during different rounds to transmit their sensed physical parameters, while the remaining nodes are turned off to conserve energy and resources. Due to inevitable data loss in WSNs, the received signal vector at the BS, $y \in \mathbb{R}^k$ is expressed as:

$$y = H\Phi f + z \quad (2)$$

Here, z represents measurement noise , $\Phi \in \mathbb{R}^{m \times n}$ is a projection matrix in which all entries are zero except for m elements located in distinct rows and columns, and $\mathrm{H} \in \mathbb{R}^{k \times m}$ $(where\ k < m)$ represents the packet loss matrix, with all elements zero except for the k positions corresponding to successfully received packets.

Φ’s entries denote the operational state of the sensor nodes —whether they are active or in sleep mode. For instance, if sensor node i is active, there is a value of 1 in the i'th column of Φ; otherwise, all elements in that column are zero. Likewise, the entries of the packet loss matrix H represent which sensor measurements were

 DOI: 10.46121/pspc.54.1.32

successfully received. For instance, a 1 in the $i-th$ column of H indicates that the data packet from sensor node i was received successfully; otherwise, it was lost during transmission.

BS can use acknowledgment (ACK) and timeout mechanisms to detect lost message packets, and employ cyclic redundancy check (CRC) algorithms to identify any alterations in the transmitted data. Based on equations (1) and (2), the signal vector at the receiver can be rewritten as:

$$y = H\Phi\Psi x + z = Ax + z \tag{3}$$

where $A = H\Phi\Psi \in \mathbb{R}^{k\times n}$ is the sensing matrix.

Note that the sensing error z arises from limitations in sensor devices, whereas transmission-related issues—such as packet loss—are captured by the matrix H.

3-2 Data Reconstruction Process

In compressive sensing, signal reconstruction involves solving the bellow optimization problem:

$$\min_{x} \|x\|_1 \tag{4}$$
$$s.t. \|Ax-y\|_2^2 \le \varepsilon.$$

where ε defines the upper bound of errors introduced by both compression and measurement noise. According to compressive sensing theory [20] approximately $O(s\log\frac{n}{s})$ measurements are sufficient for accurate reconstruction of an **s**-sparse signal in an n-dimensional space.

In WSNs, sensor measurements from natural environments typically exhibit strong spatial and temporal correlation. For example, data collected at a single time point across sensors tends to be spatially correlated, while data collected over time from a single sensor shows temporal consistency. This property has been observed in real-world datasets, such as those from the Intel Berkeley Research Lab [21], as demonstrated in [22].

With the goal of requiring fewer measurements for successful reconstruction, temporal correlation can be exploited in addition to spatial sparsity. Specifically, assuming that the data signal evolves gradually over time, the signal support from the previous time instant can be used as signal support of current signal [16, 17]. Consequently, the optimization problem below is formulated to reconstruct the original signal while utilizing prior knowledge of the signal support:

$$\min_{x} \|x\zeta^c\|_1 \tag{5}$$
$$s.t. \|Ax-y\|_2^2 \le \varepsilon.$$

Here, ζ is indicative of the signal support.

2. Proposed Method

This section presents our proposed method, which first selects active nodes via a GA and then builds an optimal Steiner Tree using ACO to ensure efficient routing.

In [11], a formula was introduced to determine which nodes to be active in the next round, aiming to minimize reconstruction error while taking network lifetime into account:

$$\min_{\tilde{\Phi}_{i.i}} Tr\left(\left(\Psi^T\zeta Q^{1/2}\tilde{\Phi}Q^{1/2}\Psi\zeta\right)^{-1}\right) - \beta\sum_{i=1}^{n}\frac{e_i\tilde{\Phi}_{i.i}}{d_i}\ s.t. \|Ax-y\|_2^2 \le \varepsilon \tag{6}$$
$$s.t. \tilde{\Phi}_{i.i} \in \{0.1\}. i = 1. \dots . n$$
$$Tr(\tilde{\Phi}) = m.$$

where $\tilde{\Phi}$ is an $n \times n$ diagonal matrix defined as follows:

$$\tilde{\Phi} = \begin{cases} 1. \text{awake} \\ 0. \text{sleep} \end{cases} \tag{7}$$

$\tilde{\Phi}$ can be expressed as a row permutation of the concatenated matrix comprising the sensor matrix Φ and an $(n-m)\times n$ zero matrix, as shown:

$$\tilde{\Phi} = \prod \begin{bmatrix} \Phi \\ 0_{(n-m)\times n} \end{bmatrix} \tag{8}$$

Where $\prod$ is a row permutation matrix and $\Phi^T\Phi = \tilde{\Phi}$.

We also assume that the estimated signal ζ from the previous round is treated as the actual support signal and reused for reconstruction.

DOI: 10.46121/pspc.54.1.32

Power System Protection and Control
ISSN:1674-3415

$Q \in \mathbb{R}^{m \times m}$ is a diagonal matrix, where each diagonal element q_i $(i = 1.2.\dots.m)$ denotes the probability of a packet being successfully received.

In [11], a penalty term was introduced to consider both the residual energy of nodes and their distance to the BS, encouraging the selection of sensors with higher energy (e_i) and shorter distances (d_i).

Solving the optimization problem in equation (6) yields a node selection matrix designed to minimize the lower bound of the mean squared error. Because the variables $\widetilde{\Phi}_{i.i}$ are binary, the optimization problem is non-convex, with (n choose m) possible combinations to find the optimal solution. For large-scale sensor networks, this requires a computationally expensive exhaustive search.

To simplify the problem, [11] applied convex relaxation by allowing $0 \leq \widetilde{\Phi}_{i.i} \leq 1$, and solved it using the interior-point method. While convex relaxation helps simplify complex problems, it may not always yield the optimal solution. Hence, to achieve better solutions, our proposed method uses a GA to construct the $\widetilde{\Phi}_{i.i}$ matrix. Each chromosome is a binary string of length n, where a 1 at position i indicates that node i is awake, and a 0 means it is asleep. The fitness function used is the same as equation (6).

Once the awake nodes are selected for the current round, we need to determine an optimal path to transmit their data to the BS. In [11], data is transmitted to the BS in a single hop, which introduces several issues previously discussed. This approach overlooks the challenge of interconnecting the awake nodes and linking them to the BS. Our proposed method addresses this by constructing an aggregation tree that efficiently routes data from active nodes to the BS. These active nodes act as terminal nodes and must be part of the tree structure. However, a tree built solely from terminal nodes is usually insufficient, necessitating the inclusion of additional intermediate nodes. We call such nodes as relay node, Figure 1-D. This leads to the Steiner Tree problem an NP-Hard optimization challenge.

We address this problem using ACO algorithm to construct the Steiner Tree. Initially, an ant is deployed at each terminal node that should be connected. An ant moves to one of its neighboring sensor nodes, in each iteration. Although the selection of the next node is stochastic, it is influenced by pheromone trails, making ants more likely to follow paths previously traversed by others. Each ant maintains a *taboo list*—a record of the nodes it has already visited—to avoid cycles. When an ant encounters another ant or intersects with the trail of a different ant, it merges with that path, forming a shared subtree. Once all ants have reached the BS, a Steiner tree is constructed in each iteration based on the collective paths taken. The proposed algorithm's pseudocode is illustrated in Figure 3.

```
for each pair (i, j) ∈ V × V, compute dist_{i,j}
for k = 1 to k_max
        initialize:
        initialize A by placing ants at the nodes in T.
        for each edge (i, j) ∈ E, τ_{i,j} = τ_0
        for each ant m ∈ A, T^(m) = {}
        for each ant m ∈ A, S^(m) = {}
        while |A| > 1 do
                select an ant m ∈ A in node i (using any rule)
                for each edge (i, j) ∈ E, compute η_{i,j}^(m)
                determine next node j for ant m (greedily or stochastically)
                update:
                T^(m) = T^(m) ∪ {j}
                S^(m) = S^(m) ∪ {(i, j)}
                if ∃m' ∈ A such that j ∈ T^(m')
                        merge:
                        T^(m') = T^(m') ∪ T^(m)
                        S^(m') = S^(m') ∪ S^(m)
                        A = A − {m}
                        if location of m' = j
                                resolve conflict:
                                compute N = {n ∈ T^(m') | ∃n' ∉ T^(m'), (n, n') ∈ E}
                                for each n ∈ N, compute ψ_n^(m')
                                place m' in node n ∈ N
                                        so that ∀n' ∈ N, ψ_n^(m') ≤ ψ_n'^(m')
                        end if
                end if
        end while
        compute S = ∪_{m∈A} S^(m)
        compute Δτ = Q / cost(S)
        for each edge (i, j) ∈ E, update τ_{i,j}
end for
```

Figure 3: Pseudocode for constructing a Steiner tree using the ant algorithm

DOI: 10.46121/pspc.54.1.32

In this pseudocode, V denotes the set of vertices (sensors) in the graph (sensor network) and is equal to the number of sensor nodes. E denotes the set of edges in the graph, where each edge connects two nodes. The number of edges varies in each simulation. $\mathrm{dist}_{i.j}$ represents the distance between nodes i and j, computed at the start of the algorithm for each node pair. k is the counter of iterations and k_{max} limits the number of iterations. A denotes the set of artificial ants employed in the algorithm to simulate the process of constructing a minimum Steiner tree.

Initially, the ants are placed on the nodes of the tree T. $\tau_{i.j}$ denotes the pheromone level on edge $(i.j)$. Initially, pheromone levels are set to zero for all edges. This parameter is updated during the algorithm as ants traverse the edges. T_m is the partial tree being constructed by ant m which is initially empty. S_m represents the set of edges visited by ant m, which is also initially empty. $\eta_{i.j}^{(m)}$ is the heuristic value guiding ant m in selecting edge $(i.j)$ to proceed to the next node. $\psi_n^{(m')}$ is a heuristic used to resolve conflicts when multiple ants attempt to merge their trees.

This heuristic assists ant m' in selecting the optimal next node n during conflict resolution. S denote the final set of edges forming the Steiner tree after all ants have completed their paths and merged their individual trees. The function $\mathrm{cost}(S)$ represents the total cost of the Steiner tree and is calculated as the sum of the weights of all edges in S. The constant Q is a tunable parameter used in updating the pheromone levels. $\Delta\tau_{i.j}$ indicates the amount of pheromone to be deposited on edge $(i.j)$ based on the quality of the solution obtained. This value is computed and updated at the end of each iteration.

Ant m, currently located at node i, selects a neighboring node j that has not been visited previously, i.e. $j \notin T^{(m)}$, where $T^{(m)}$ is the set of nodes already visited by ant m. Ants tend to prefer edges with higher pheromone levels $\tau_{(i.j)}$. Additionally, ants are more likely to move towards paths that have already been traversed by other ants, promoting faster convergence. This behavior helps reduce the cost of the resulting Steiner tree. To facilitate this, we define a potential parameter for each node j as follows:

$$\Psi_j^{(m)} = \min_k\{dist(j.k)\}. \tag{9}$$

and

$$k \in \bigcup_{m' \neq m} T^{(m')} \tag{10}$$

Here, m' denotes another ant and $dist(j.k)$ represents the minimum distance between nodes j and k. This distance is calculated as the total length of all edges directed from node j to node k . The Floyd-Warshall algorithm is utilized to efficiently compute the shortest paths between all pairs of nodes.

Thus, a node's potential reflects the minimum additional cost required to link it to any of the existing partial trees. In this approach, we aim to guide ants towards nodes with lower potential, as these nodes are closer to the partial tree. However, the cost of transferring an ant from its current location to other nodes must also be considered in the calculation. Therefore, to account for both factors, we define the attractiveness of a node j for ant m, which is currently at node i, as follows:

$$\eta_{i.j}^{(m)} = \frac{1}{w(i.j) + \gamma\Psi_j^{(m)}} \tag{11}$$

In the above equation, γ represents a constant coefficient. The proposed algorithm directs the ants towards selecting optimal edges. $w(i.j)$ incorporates two parameters: the remaining energy of node j and the distance between nodes i and j, expressed as $w(i.j) = \frac{e_j}{d_{ij}}$.

The algorithm simultaneously aims to guide the ants along paths with high pheromone levels and potential. By considering both factors, it enables ants to preferentially move towards edges with a higher value of the product of pheromone intensity and desirability. The probability of selecting node j is then calculated using the following formula:

$$\boldsymbol{p_{i.j}} = \frac{[\boldsymbol{\tau_{i.j}}]^{\alpha}[\eta_{i.j}^{(m)}]^{\beta}}{\sum_{k \notin T^{(m)}}[\boldsymbol{\tau_{i.j}}]^{\alpha}[\eta_{i.j}^{(m)}]^{\beta}} \tag{12}$$

DOI: 10.46121/pspc.54.1.32

Power System Protection and Control
ISSN:1674-3415

The value $\boldsymbol{p_{i,j}}$ represents the probability of selecting the edge $(\boldsymbol{i.j}) \in \boldsymbol{E}$. In the above equation, α and β are two constants that remain fixed throughout the algorithm.

The pheromone trail is updated as follows: ants collectively deposit pheromone along the edges of the path, inversely proportional to the total cost of the tree. The pheromone trail is updated at the end of each iteration. The pheromone update is performed according to the following formula:

$$\boldsymbol{\tau_{i,j} = (1-\rho)\tau_{i,j} + \rho\Delta\tau_{i,j}} \quad (13)$$

where $\boldsymbol{\rho}$ is the pheromone evaporation rate, representing the speed at which the pheromone on the path evaporates. The general form of gradual updates is given by the following equation:

$$\Delta\boldsymbol{\tau_{i,j}} = \begin{cases} \dfrac{\boldsymbol{Q_t}}{\boldsymbol{w(S_t)}} & \boldsymbol{if(i.j) \in E_{S_t}} \\ \boldsymbol{0\ else} \end{cases} \quad (14)$$

In this context, $\boldsymbol{w(S_t)}$ represents the cost of the Steiner tree, defined as the sum of the distances of its edges. The proposed method incorporates two types of updates, as described in [23]. The local update rule applies to the Steiner tree $\boldsymbol{S_t}$, which is used to compute the incremental pheromone updates $\Delta\boldsymbol{\tau_{i,j}}$. Specifically, the updates are performed using the tree calculated during the current iteration, denoted as $\boldsymbol{S_t = S_{local}}$.

The value of $\mathbf{Q}$ is maintained at a constant level, denoted as $\mathbf{Q}_{\boldsymbol{local}}$, for local updates. However, global round updates are also utilized, where the pheromone increment is applied only to the edges that are part of the best Steiner tree identified from the start of the iteration to the most recent round. In other words, the update is also performed for the state where $\boldsymbol{S_t = S_{best}}$, with the corresponding value of $\mathbf{Q}$ being $\mathbf{Q}_{\boldsymbol{global}}$.

As expressed in equation (14), the update value is inversely proportional to the total cost of the obtained Steiner tree. This approach ensures that more pheromone is deposited on edges with lower costs. Consequently, higher-quality trees contribute more pheromone to their edges, increasing the likelihood that subsequent ants will select these paths.
At this stage, the constructed tree is initially used to send control packets to the nodes that must participate in the current round. Upon receiving a control packet, each node identifies its parent and children in the tree and waits to receive the corresponding packets. Subsequently, this same tree is employed to transmit compressive sensor data from the awake nodes.

3. Simulations and Results

This section presents the results which is using synthetic data to evaluate the proposed technique in terms of signal reconstruction accuracy and network lifetime. The simulations were carried out in the MATLAB programming environment. To assess energy consumption, the model proposed in [24] was employed. Four states are defined for the sensor radio: transmit, receive, listen (active), and sleep.

Table 1 presents the energy consumption associated with each state, along with the corresponding required time. The energy consumption is calculated using the formula:

$$\text{Energy} = \text{Power} \times \text{Time} \quad (15)$$

For instance, the transmitter consumes some energy, E to transmit a data packet of length L which can be calculated using the following equation:

$$\text{E} = \text{Power} \times \text{Time} = \text{P}_t \times \text{Time} = \text{P}_t \times \text{L} \times T_B = 60\ (mW) \times 8\ (\text{Byte}) \times 0.416 \times 10^{-3}\ (\text{Second}) = 0.19968\ \text{mJ}$$

DOI: 10.46121/pspc.54.1.32

amount	symbol	
60 mW	P_t	Sending mode power consumption
45 mW	P_r	Receive mode power consumption
45 mW	P_l	Listening mode power consumption
90 μW	P_s	Sleep mode power consumption
5.75 μW	P_{pc}	State change power consumption
$0.416 \times 10^{-3}\ S$	T_B	Time of receiving or sending a byte
3 mS	T_{pc}	Status change time

Table 1 The parameters and their values in the energy consumption model used

The length of both data and control packets is assumed to be 64 bits. Sensor transmissions occur over an AWGN channel, leading to varying probabilities of data loss. For compressive sensor signal reconstruction, the CVX software package [25] is employed. The reconstruction performance is assessed using the mean relative error, defined as:

$$\frac{\|\boldsymbol{x}\|_2^2}{\|\hat{\boldsymbol{x}} - \boldsymbol{x}\|_2^2} \tag{16}$$

where $\boldsymbol{x}$ represents the original signal and $\hat{\boldsymbol{x}}$ the signal after reconstruction. The network lifetime is determined based on the time when the first sensor is deactivated. To assess the effectiveness of the proposed method, it is compared with several existing approaches: the method in [7], which relies solely on spatial correlation; the method in [11], that takes advantage of spatial and temporal correlation; and the method in [10] that uses a model-free Q-learning approach to identify optimal strategies for decision-making while simultaneously applying compressive sensing and sleep scheduling.

Within the simulations, n wireless sensor nodes are scattered randomly over a square area with dimensions of d × d m^2. The communication radius of the sensors is R meters. One corner of the environment hosts the BS, specifically at coordinates (0, 0). All sensor nodes initially possess an energy level E_{int}. The data signal x is generated randomly at different time instants, with the s non-zero components being independently and identically distributed (i.i.d.) following a Gaussian distribution with zero mean and unit variance. Initially, Ψ is constructed as an $n \times n$ matrix whose entries are independent and identically distributed according to a Gaussian distribution $N(0.1)$, after which the columns of Ψ are scaled to unit norm. The wireless channel's signal-to-noise ratio (SNR) is set to 20 dB, that, in combination with modulation type and packet length, governs the likelihood of data loss [26]. Table 2 shows the simulation parameters and values.

Table 2: WSN simulation parameters and values

symbol	amount	explanation
n	1000	The number of nodes
R	5 meters	Communication range length
E_{int}	5 joules	Initial energy of nodes
E_{death}	0.1 joule	Final energy of nodes
d	100 x 100 square meters	Network size

The value of β is set to 10^{-3}. Additionally, sensor measurements are subject to zero-mean Gaussian noise, producing a signal-to-noise ratio of 20 dB. In all experiments, it is assumed that 20% of the estimated signal is erroneous.

For the ACO algorithm simulation, For the experiments, 50 ants and 100 iterations were adopted. The parameter values were set as follows: $\rho = 0.1$, $\gamma = 5$, $\lambda = 2$, $\alpha = 1$, and $\beta = 1.5$. The value of Q was experimentally determined to be 550 after several trials.

As mentioned earlier, the proposed method in this paper employs a GA to optimally construct $\widetilde{\Phi}_{i,i}$. The length of the binary chromosomes is set to $n = 1000$ and the population size is 50. Roulette wheel selection was applied with a selection rate of 80%, and both two-point crossover and point mutation (with a mutation rate of $\frac{m}{n}$) were

DOI: 10.46121/pspc.54.1.32

utilized. The loop termination condition was set to 100 iterations. Figure 4 depicts how the fitness function converges over different generations of the GA for m = 200 and s = 100.

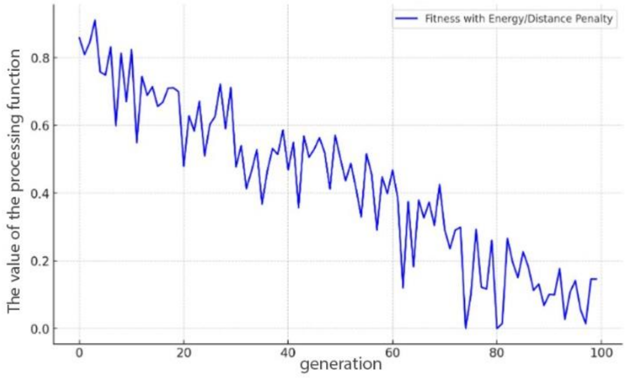

Figure 4: Convergence of the fitness function in different generations of the proposed GA

Figure 5 illustrates the reconstruction performance of the proposed approach in comparison with the methods described in [7], [10], and [11] for varying values of s, with m fixed at 200. It is evident that for all methods, the reconstruction error diminishes as s decreases. Notably, the proposed method consistently achieves a lower reconstruction error compared to the other methods. Specifically, the proposed method achieves approximately a 48% reduction in reconstruction error compared to the method in [7]. This improvement can be attributed to the strategic selection of appropriate nodes for data transmission in each round, whereas in [7], the selection of awake nodes is performed randomly.

Compared to the method in [11], the proposed method demonstrates a 59% improvement in reconstruction error. This significant reduction can be attributed to the use of a GA for selecting awake nodes. In contrast, [11], applies convex relaxation to reduce the complexity of the problem, which does not yield as optimal solutions as the GA. Additionally, the method proposed in [11] is distributed and unable to efficiently identify suitable awake nodes, resulting in a shadow matrix that is suboptimal. As a result, the reconstruction error for this method is greater than the value achieved by the proposed method. Specifically, the proposed algorithm achieves a 61% improvement in reconstruction accuracy over the method in [10].

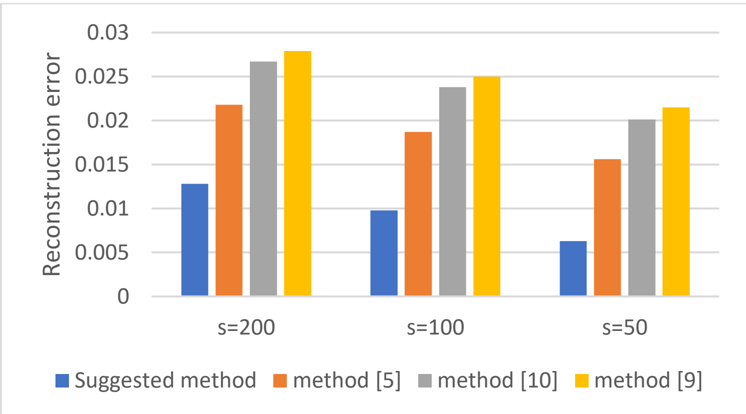

Figure 5: Reconstruction error for different sparsity levels

Figure 6 presents a comparison of network lifetime for the proposed method and three other methods, with s = 100. The network lifetime is evaluated for different values of m, denoting the number of nodes that remain awake during each round. The results indicate that he proposed approach considerably surpasses the compared methods in [7] and in terms of network lifetime. his improvement in performance is due to the optimal choice of awake nodes through the GA, taking into consideration the leftover energy of the sensors, and the use of the ant colony algorithm for optimal routing. The method proposed in [10] shows a relatively similar network lifetime to our

DOI: 10.46121/pspc.54.1.32

approach. This is due to the distributed nature of their algorithm, which reduces the overhead associated with sending control packets.

Specifically, our proposed method demonstrates an average network lifetime improvement of approximately 18% compared to the method in [7], 32% compared to the method in [11], and 1% compared to the method in [10].

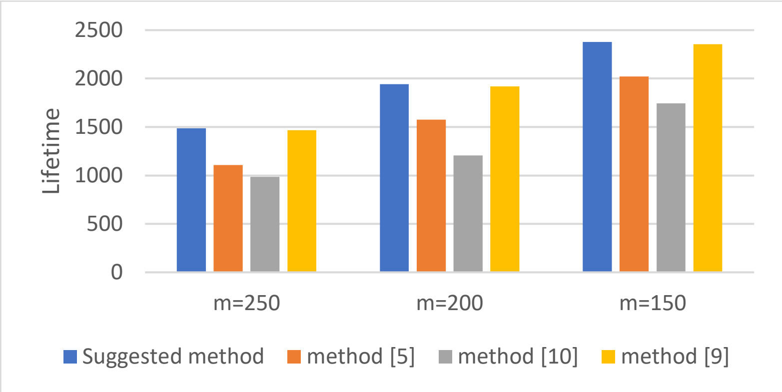

Figure 6: Network lifetime per number of awake nodes

Figure 7 illustrates the reconstruction error for the simulated methods across different values of m, with s = 100. As observed, increasing the number of active nodes per round leads to a lower reconstruction error for all evaluated methods. Furthermore, the results clearly show that the proposed approach achieves lower reconstruction error compared to the other two evaluated methods. The reasons for this superiority, as previously discussed, are attributed to the optimal selection of awake nodes and the integration of the GA and ant colony algorithm, which result in enhanced performance in terms of both reconstruction accuracy and energy efficiency.

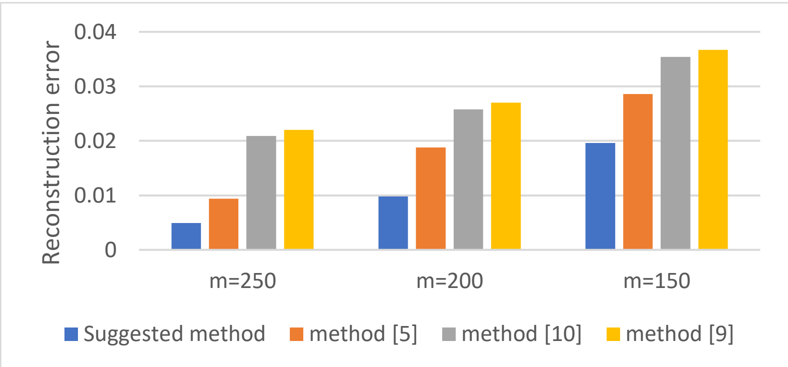

Figure 7: Reconstruction error per number of awake nodes

In a sensor network, both the precision of the received data signal and the longevity of the network are crucial factors. Figure 8 presents an evaluation that simultaneously examines the reconstruction error as well as the network's lifetime, expressed in terms of the performance metric $P = \frac{Lifetime}{\mathrm{error} * 10^4}$ for various values of m and s in the proposed method. This evaluation aims to determine the optimal value of m, balancing reconstruction accuracy with network longevity. This approach allows for a comprehensive assessment of the compromise between preserving data accuracy and achieving energy efficiency in the sensor network.

DOI: 10.46121/pspc.54.1.32

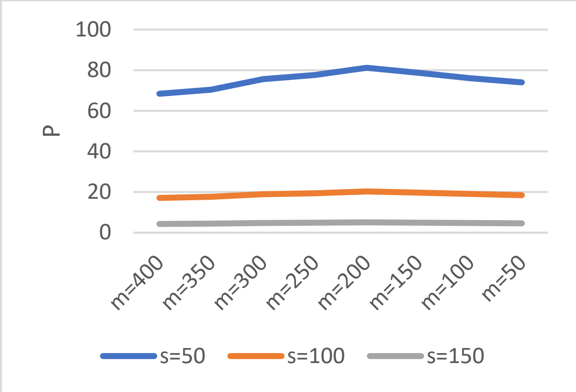

Figure 8: Simultaneous examination of lifetime and accuracy for the number of different awake nodes

The results demonstrate that increasing m, showing the count of active nodes at every phase, reduces the reconstruction error; however, this improvement comes at the expense of reduced network longevity. Based on the conducted experiments, $m = 200$ was identified as the optimal number of awake nodes, striking a balance between minimizing reconstruction error and maintaining network longevity.

As the final evaluation of the overall performance of all four methods, both the lifetime and reconstruction error parameters were simultaneously assessed using the performance metric $P = \frac{Lifetime}{\text{error}*10^4}$. This experiment was conducted with the optimal value of $m = 200$, providing a comprehensive comparison of the methods in terms of their ability to balance accuracy and energy efficiency.

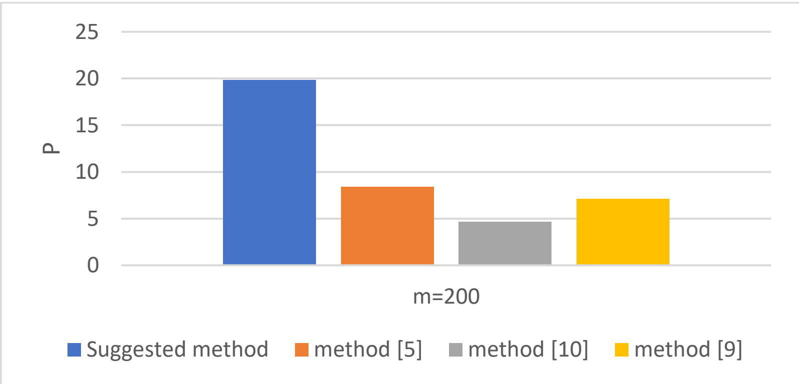

Figure 9: Simultaneous examination of the lifetime and accuracy of the evaluated methods

As illustrated in the Figure 9, when both network lifetime and reconstruction error are considered simultaneously, the proposed method outperforms all three baseline methods. This superior performance is attributed to the combined use of a GA for the optimal selection of awake nodes and ACO algorithm for efficient routing. These strategies, as previously discussed, lead to more accurate signal reconstruction and improved energy efficiency, thereby extending network lifetime.

CONCLUSION

Compressive sensing provides an effective strategy for extending the operational duration of WSNs through minimizing the number of nodes participating in data transmission. Nodes not selected for transmission can be transitioned into sleep mode to conserve energy. However, random or distributed selection of awake nodes often results in high reconstruction error. To address this, we proposed a centralized approach that employs a GA to optimally select awake nodes. The ACO algorithm was further used to design an energy-efficient communication path linking the selected nodes with the BS.

DOI: 10.46121/pspc.54.1.32

Results from the simulations show that the proposed technique leads to considerable improvements in network efficiency, achieving both longer network lifetime and lower signal reconstruction error compared to existing approaches.

In this work, the selection of awake nodes was based on a relationship involving the signal from the previous time step. For future research, we aim to enhance node selection accuracy by incorporating predictive models that estimate the upcoming signal values through temporal correlation in sensor data. These predictions will then be used to further refine the process of awake node identification.

RESOURCES

1. G. A. López-Ramírez and A. Aragón-Zavala, "Wireless sensor networks for water quality monitoring: a comprehensive review," IEEE access, vol. 11, pp. 95120-95142, 2023.
2. M. N. Mowla, N. Mowla, A. S. Shah, K. Rabie, and T. Shongwe, "Internet of things and wireless sensor networks for smart agriculture applications-a survey," IEEE Access, 2023.
3. D. R. Zaidan, A. G. Wadday, M. M. Abbood, A. F. Al-Baghdadi, and B. J. Hamza, "Forest fire detection based wireless sensor networks-survey," in AIP Conference Proceedings, 2023, vol. 2776, no. 1: AIP Publishing.
4. D. L. Donoho, "Compressed sensing," IEEE Transactions on information theory, vol. 52, no. 4, pp. 1289-1306, 2006.
5. C. Luo, F. Wu, J. Sun, and C. W. Chen, "Compressive data gathering for large-scale wireless sensor networks," in Proceedings of the 15th annual international conference on Mobile computing and networking, 2009, pp. 145-156.
6. L. Xiang, J. Luo, and C. Rosenberg, "Compressed data aggregation: Energy-efficient and high-fidelity data collection," IEEE/ACM transactions on Networking, vol. 21, no. 6, pp. 1722-1735, 2012.
7. Q. Ling and Z. Tian, "Decentralized sparse signal recovery for compressive sleeping wireless sensor networks," IEEE Transactions on Signal Processing, vol. 58, no. 7, pp. 3816-3827, 2010.
8. T. Xue, X. Dong, and Y. Shi, "Multiple access and data reconstruction in wireless sensor networks based on compressed sensing," IEEE Transactions on Wireless Communications, vol. 12, no. 7, pp. 3399-3411, 2013.
9. S. Mehrjoo, F. Khunjush, and A. Ghaedi, "Fully distributed sleeping compressive data gathering in wireless sensor networks," IET Communications, vol. 14, no. 5, pp. 830-837, 2020.
10. X. Wang, H. Chen, and S. Li, "A reinforcement learning-based sleep scheduling algorithm for compressive data gathering in wireless sensor networks," EURASIP Journal on Wireless Communications and Networking, vol. 2023, no. 1, p. 28, 2023.
11.]W. Chen and I. J. Wassell, "Optimized node selection for compressive sleeping wireless sensor networks," IEEE Transactions on Vehicular Technology, vol. 65, no. 2, pp. 827-836, 2015.
12. M. Al Mazaideh and J. Levendovszky, "A multi-hop routing algorithm for WSNs based on compressive sensing and multiple objective genetic algorithm," Journal of Communications and Networks, vol. 23, no. 2, pp. 138-147, 2021.
13. A. Aziz, W. Osamy, A. M. Khedr, A. A. El-Sawy, and K. Singh, "Grey Wolf based compressive sensing scheme for data gathering in IoT based heterogeneous WSNs," Wireless Networks, vol. ,26no. 5, pp. 3395-3418, 2020.
14. K. P. Mhatre and U. P. Khot, "Energy efficient opportunistic routing with sleep scheduling in wireless sensor networks," Wireless Personal Communications, vol. 112, pp. 1243-1263, 2020.

 DOI: 10.46121/pspc.54.1.32

15. N. M. Shagari, M. Y. I. Idris, R. B. Salleh, I. Ahmedy, G. Murtaza, and H. A. Shehadeh, "Heterogeneous energy and traffic aware sleep-awake cluster-based routing protocol for wireless sensor network," IEEE Access, vol. 8, pp. 12232-12252, 2020.
16. N. Vaswani and W. Lu, "Modified-CS: Modifying compressive sensing for problems with partially known support," IEEE Transactions on Signal Processing, vol. 58, no. 9, pp. 4595-4607, 2010.
17. W. Lu and N. Vaswani, "Regularized modified BPDN for noisy sparse reconstruction with partial erroneous support and signal value knowledge," IEEE Transactions on Signal Processing, vol. 60, no. 1, pp. 182-196, 2011.
18. G. Quer, R. Masiero, G. Pillonetto, M. Rossi, and M. Zorzi, "Sensing, compression, and recovery for WSNs: Sparse signal modeling and monitoring framework," IEEE Transactions on Wireless Communications, vol. 11, no. 10, pp. 3447-3461, 2012.
19. M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," IEEE Transactions on Image processing, vol. 15, no. 12, pp. 3736-3745, 2006.
20. R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, "A simple proof of the restricted isometry property for random matrices," Constructive approximation, vol. 28, pp. 253-263, 2008.
21. W. H. P. Bodik, C. Guestrin, S. Madden, M. Paskin, and and R. Thibaux. Intel lab data [Online] Available: http://db.csail.mit.edu/labdata/labdata.html
22. W. Chen, M. R. Rodrigues, and I. J. Wassell, "A frechet mean approach for compressive sensing date acquisition and reconstruction in wireless sensor networks," IEEE Transactions on wireless communications, vol. 11, no. 10, pp. 3598-3606, 2012.
23. M. Dorigo, V. Maniezzo, and A. Colorni, "Ant system: optimization by a colony of cooperating agents," IEEE transactions on systems, man, and cybernetics, part b (cybernetics), vol. 26, no. 1, pp. 29-41, 1996.
24. V. Shnayder, M. Hempstead, B.-r. Chen, G. W. Allen, and M. Welsh, "Simulating the power consumption of large-scale sensor network applications," in Proceedings of the 2nd international conference on Embedded networked sensor systems, 2004, pp. 188-200.
25. M. Grant, S. Boyd, and Y. Ye, "CVX: Matlab software for disciplined convex programming, version 2.0 beta," ed, 2013.
26. J. G. Proakis, Digital communications. McGraw-Hill, Higher Education, 2008.

DOI: 10.46121/pspc.54.1.32