DOI: 10.21917/ijct.2025.0524

ADAPTIVE MULTI-FACTOR CO-EVOLUTIONARY ALGORITHM WITH LOCAL SEARCH FOR EFFICIENT CLUSTER HEAD SELECTION IN WIRELESS SENSOR NETWORKS

Minal Sudarshan Hardas[1] and M. Ezhilvendan[2]
[1]*Department of Electronics and Telecommunication, Shivajirao S. Jondhale College of Engineering, India*
[2]*Department of Artificial Intelligence and Machine Learning, Panimalar Engineering College, India*

Abstract

In a lot of fields, like watching the environment, watching military bases, and developing smart cities, wireless sensor networks (WSNs) are particularly significant. One of the main difficulties with WSNs is how to use energy intelligently so that the network lasts as long as feasible. Using cluster-based topologies is a fantastic way to get the most out of energy utilization. In these topologies, a set of nodes in the network known as Cluster Heads (CHs) are responsible for communicating with the base station and other sensor nodes. It is still hard to choose a CH because WSNs are continually evolving and there are a lot of elements to worry about, like energy, coverage, and node density. Traditional approaches for choosing a CH don't function well on heterogeneous and dynamic WSNs because they use static algorithms or single-factor optimization. To make the network more stable, energy-efficient, and long-lasting, we need an algorithm that can swiftly adjust to changes in the network and take a number of different things into account at once. The Adaptive Multi-Factor Co-Evolutionary Algorithm with Local Search, or AMCE-LS, is the main focus of this study. The approach has a co-evolutionary framework that uses a variety of adaptive fitness criteria to give nodes a score. The measures include coverage, the distance between nodes in the same cluster, the degree of each node, and the energy that is left behind. Adding a local search refinement step, which makes the CH selection even better, speeds up the convergence and makes the response better. Because it can monitor the network in real time, the adaptive technique can modify the weighting variables on the fly. The recommended AMCE-LS approach works better than earlier algorithms like LEACH, PSO, and DEEC when it comes to testing the durability of a network, its energy efficiency, and its packet delivery ratio. In dense node installations, the adaptive multi-factor technique can help networks persist and stay stable for up to 30% longer.

Keywords:

Wireless Sensor Networks, Cluster Head Selection, Co-Evolutionary Algorithm, Local Search, Energy Efficiency

1. INTRODUCTION

For many real-time monitoring jobs, like environmental sensing, industrial automation, healthcare, and military surveillance, WSNs are becoming a must-have [1]. Sensor nodes are the building blocks of these networks. They are spread out and have low processing power, range, and power consumption.

For the network to perform well, the procedures of obtaining and sending data must be efficient. This is necessary because energy is limited, and the network has to survive longer [2]. Sending data across clusters is a typical way to communicate. This strategy consists of grouping sensor nodes into groups and picking a leader for each group, which is known as a Cluster Head (CH) [3]. The Cluster Head is in charge of the cluster and sending data to the base station (BS).

Cluster-based communication has several benefits, but it also has a number of drawbacks. Choosing the correct CHs is the most crucial thing you can do because it has such a large effect on how reliable the network is and how much power it requires. Static or probabilistic selection methods, such as LEACH or TEEN, can't keep up with how quickly networks change in real time. This means that energy is used up in an uneven way, and CH is picked repeatedly [4]. We also need to find a smarter and more flexible strategy to identify CHs because WSNs have diverse energy levels, node density, and communication needs [5].

The existing means of picking CHs don't take into account that WSNs are complex and continually changing. They frequently utilize heuristics or strategies that focus on one goal. This is because these solutions don't consider how sophisticated WSNs are.

These approaches typically have issues in different contexts, like spending too much energy, converging too fast, and not being able to scale properly [6]. Because of this, it is quite vital to have a technique to choose CHs that can automatically look at a variety of things and learn from what it sees in the environment. This study aims to:

- Develop a robust and adaptive cluster head selection algorithm that integrates multiple decision factors.
- To create an adaptive cluster head selection approach that considers a lot of different aspects.
- To use a co-evolutionary framework to develop many solutions at the same time over a number of fitness landscapes.

The novelty of this work lies in the fusion of a Multi-Factor Co-Evolutionary Algorithm (MF-CEA) with an adaptive local search scheme, tailored specifically for dynamic WSN environments. Unlike traditional models, our approach adaptively weighs factors such as residual energy, node density, coverage redundancy, and intra-cluster distance during fitness evaluation, ensuring context-aware CH selection. Key contributions of the study include:

- The proposed model dynamically adjusts weighting parameters for fitness criteria, enabling better responsiveness to environmental changes.
- By maintaining sub-populations that evolve in parallel, the algorithm mitigates premature convergence and enhances global search capability.
- Incorporating a local refinement phase leads to more accurate and energy-efficient CH assignments.

2. RELATED WORKS

The problem of cluster head selection in WSNs has been extensively studied using various optimization and heuristic approaches.

2.1 LEACH AND PROBABILISTIC APPROACHES

The Low-Energy Adaptive Clustering Hierarchy (LEACH) is one of the earliest and most cited protocols for CH selection [7]. LEACH uses random rotation of CHs to balance energy usage but lacks adaptability and fails in heterogeneous environments. Its derivatives like LEACH-C and TEEN incorporate centralized control and threshold-based techniques respectively but still rely on probabilistic mechanisms [8][9].

2.2 FUZZY LOGIC-BASED MODELS

Fuzzy logic techniques have been used to manage uncertainty and imprecision in CH selection. Models like Fuzzy LEACH apply fuzzy inference systems to evaluate nodes based on energy, distance, and node degree [10]. However, the static rule base limits scalability and adaptability in dynamic networks [11].

2.3 SWARM INTELLIGENCE AND EVOLUTIONARY ALGORITHMS

Particle Swarm Optimization (PSO) and Genetic Algorithms (GA) have been widely adopted for CH selection. PSO models utilize the collective behavior of particles to find optimal CHs based on multiple criteria like residual energy and coverage [12]. GA-based models use crossover and mutation operations to evolve candidate solutions but often suffer from slow convergence and require extensive tuning [13].

2.4 HYBRID ALGORITHMS

Recent studies have combined different techniques to overcome individual limitations. For example, hybrid GA-PSO approaches aim to leverage the exploration capabilities of GA with the exploitation power of PSO [14]. These hybrid models offer better performance but increase algorithmic complexity and computational overhead.

2.5 MULTI-OBJECTIVE AND METAHEURISTIC APPROACHES

Researchers have studied multi-objective optimization models including NSGA-II and MOEA/D [15]. People have utilized these models to find the best balance between energy, latency, and coverage. These methods work, but they need too much processing power to be used in real-time or embedded systems.

A lot of research has been done on how to pick the ideal CH, but most of these methods either don't take changing scenarios into account at all or involve central processing. There haven't been many studies on co-evolutionary algorithms that are specific to WSNs, and even fewer that have employed local search approaches to improve them over time. Our proposed method fixes these issues by providing a decentralized, adaptable, and multi-factor solution that may grow and alter quickly in response to changes in the network.

3. PROPOSED METHOD

The proposed AMCE-LS method uses co-evolutionary optimization, which is backed up by local search, to intelligently identify the optimal CHs in WSN by dynamically evaluating a large number of decision criteria. At the beginning of the process, a number of smaller groups are made. Each of these smaller groups of people could be a group of CHs, which are provided as viable solutions. An adaptive fitness function rates each individual based on four primary things: the node is remaining energy, the distance between clusters, the strength of connectivity between nodes, and the quantity of redundant coverage.

A co-evolutionary algorithm keeps things interesting and stops them from coming together too soon by having subpopulations change at the same time through mutation, selection, and crossover. This helps the algorithm keep items from being the same. The approach is expected to function effectively when things change quickly, like when nodes fail or run out of juice. This is done by altering the weight of each fitness factor over time based on how well the network is doing. After a few generations, the best people go through a local search refinement to improve the CH placements even further. We can lower the amount of transmission overhead while still making sure that the CH position is good enough. Finally, the CH set is chosen as the best contender and then distributed out to build clusters. The plan is goals are to keep the network running as long as feasible, use energy in a way that is fair, and transfer data quickly and reliably.

3.1 INITIALIZATION

During the setup process, sensor nodes are spread out across a particular area of monitoring in a way that is impossible to predict. During the deployment process, each node gets a unique ID, an initial amount of energy, and a place to be. The system specifies the algorithm is parameters, such as the size of the population, the maximum number of generations, the rates of crossover and mutation, and the weights of the beginning fitness factors. Table 1 displays an example of 10 sensor nodes, including their beginning position (X,Y), energy level, and coverage range.

Table.1. Initial Sensor Node Deployment Parameters

Node ID	X-Coord	Y-Coord	Initial Energy (J)	Coverage Radius (m)
N1	12.5	43.2	2.0	10
N2	28.1	33.4	1.8	10
N3	40.3	12.9	2.0	10
N4	19.6	20.0	1.9	10
N5	35.0	50.0	2.0	10

Using this information as a fixed base, you may look at every potential way to build up the cluster head.

3.2 CANDIDATE SOLUTION ENCODING

Every alternative solution (individual) is a way to show a set of nodes that could be chosen as cluster heads (CHs). If a node is a CH, it is shown as 1 in binary code; if it is not a CH, it is shown as 0. This is how the encoding works. This is an example of what a chromosome might look like: The sequence is [0 1 0 1 0 0 0 1

0] when there are ten nodes. Based on this encoding, nodes N2, N5, and N9 have been chosen to be CHs for this particular candidate. There are three encoded persons in Table 2, one for each potential CH arrangement. These people are listed in alphabetical order.

Table.2. Candidate Solution Encoding (Sample Individuals)

Individual ID	Node Selection (Binary)	Selected CHs
I1	[0 1 0 0 1 0 0 0 1 0]	N2, N5, N9
I2	[1 0 1 0 0 0 1 0 0 0]	N1, N3, N7
I3	[0 0 0 1 1 1 0 0 0 0]	N4, N5, N6

With this concept, the computer can use genetic operations like mutation and crossover to change CH sets over time.

3.3 FITNESS EVALUATION

A fitness function with four criteria is utilized to provide a score to someone. Here are the standards:

- **Residual Energy (RE):** Remaining energy of selected CHs.
- **Average Intra-Cluster Distance (ICD):** Average distance from member nodes to the CH.
- **Node Degree (ND):** Number of neighbor nodes within a CH is coverage.
- **Coverage Redundancy (CR):** Overlap in sensing regions among CHs.

After the fitness score has been normalized, a weighted sum approach is utilized to put all the pieces together into one fitness score. The weights alter over time dependent on how well the network is working in this strategy. The fitness function is defined as:

$$\text{Fitness} = w_1 \cdot f_{RE} + w_2 \cdot f_{ICD} + w_3 \cdot f_{ND} + w_4 \cdot f_{CR} \qquad ()$$

The Table.3 shows a sample calculation of the four fitness factors for three individuals.

Table.3. Fitness Component Values for Sample Individuals

Individual	Residual Energy (f_1)	ICD (f_2)	Node Degree (f_3)	Coverage Redundancy (f_4)
I1	0.85	0.78	0.70	0.80
I2	0.90	0.60	0.60	0.75
I3	0.88	0.72	0.68	0.77

The Table.4 shows the final fitness scores calculated using the equation above.

Table.4. Final Fitness Scores of Individuals

Individual	Final Fitness Score
I1	0.817
I2	0.765
I3	0.791

Based on Table 4, individual I1 is currently the best candidate for CH selection due to its highest fitness score. This process is repeated in each generation, with updated weights to reflect changing network conditions such as node energy depletion or increasing coverage gaps.

3.4 CO-EVOLUTIONARY OPTIMIZATION

The Co-Evolutionary Optimization phase is responsible for evolving multiple sub-populations in parallel to increase diversity and avoid premature convergence. Each sub-population evolves independently using selection, crossover, and mutation. Periodically, high-performing individuals are shared (migrated) across sub-populations to introduce beneficial traits.

For example, Table.5 presents three sub-populations (SP1, SP2, SP3) with selected individuals and their fitness scores.

Table.5. Sub-Populations with Fitness Scores

Sub-Population	Individual	Binary CH Encoding	Fitness Score
SP1	I1	[0 1 0 1 0 0 1 0 0 0]	0.790
SP2	I2	[1 0 0 0 1 0 0 0 1 0]	0.802
SP3	I3	[0 0 1 0 0 1 0 1 0 0]	0.779

During each generation, individuals with higher fitness are selected for crossover. For example, crossover between I2 from SP2 and I1 from SP1 produces a new individual: [1 1 0 1 1 0 1 0 1 0]. Mutation randomly flips bits to maintain diversity (e.g., flipping bit 6 results in: [1 1 0 1 1 1 1 0 1 0]). Migration occurs after every k generations, where the top individual from each sub-population is exchanged. This enhances global search and combines strengths from different subspaces.

3.5 ADAPTIVE WEIGHT ADJUSTMENT

To ensure adaptability, the algorithm dynamically adjusts the weights used in the fitness function based on real-time network metrics. If the average residual energy in the network decreases, more importance is given to the energy component of the fitness. Conversely, if cluster distances become large, the intra-cluster distance weight is increased.

Let the weight update rule be governed by a normalized adjustment factor Δw_i, computed as:

$$\Delta w_i = \frac{P_i}{\sum_{j=1}^{n} P_j} \qquad (1)$$

The Table.6 shows the observed network metrics and their corresponding priority scores in a given generation.

Table.6. Network Status and Factor Priorities

Factor	Observed Value	Priority Score (P_1–P_4)
Residual Energy (RE)	Low (0.48 avg)	0.40
Intra-Cluster Distance (ICD)	High (25 m)	0.35
Node Degree (ND)	Medium	0.15
Coverage Redundancy (CR)	High	0.10

Using the equation, updated normalized weights are calculated as shown in Table.7.

Table.7. Updated Weights Based on Priority Scores

Fitness Factor	Updated Weight (wiw_iwi)
RE	0.40 / 1.0 = 0.40
ICD	0.35 / 1.0 = 0.35
ND	0.15 / 1.0 = 0.15
CR	0.10 / 1.0 = 0.10

These weights are then plugged into the fitness function for the next generation, making the algorithm responsive to the network is changing needs.

3.6 LOCAL SEARCH REFINEMENT

After the co-evolutionary process converges (or reaches near-optimal solutions), Local Search Refinement is applied to the top-ranked individuals to fine-tune CH placement. This process explores the local neighborhood of selected CHs by slightly modifying the current solution (e.g., swapping one CH with a nearby high-energy node). For instance, if node N5 is currently a CH, the algorithm checks nearby nodes (within 2-hop distance) like N4 or N6 to see if replacing N5 improves the overall fitness. Table 8 shows a local search applied to individual I1, where each variation is evaluated for fitness.

Table.8. Local Search Variations for Best Individual

Variation	CH Configuration (Modified)	Fitness Score
Original	[0 1 0 1 0 0 1 0 0 0] (N2, N4, N7)	0.790
Var-1	[0 1 0 0 1 0 1 0 0 0] (N2, N5, N7)	0.812
Var-2	[0 1 0 0 0 1 1 0 0 0] (N2, N6, N7)	0.805

In this example, Var-1 outperforms the original solution, so it replaces the original CH configuration in the population. This refinement step ensures the final solution is not just globally promising, but also locally optimized for the given topology.

3.7 FINAL CLUSTER HEAD SELECTION

The termination criterion in AMCE-LS is based on either a fixed number of generations (e.g., 100 iterations) or the convergence of the population (i.e., no significant improvement in the fitness score over multiple generations). Once the termination condition is met, the algorithm evaluates the final population and selects the best individual—the one with the highest fitness score—as the optimal CH configuration. For example, after 100 generations, the top five individuals in the final population are shown in Table 9.

Table.9. Final Generation – Top Individuals Ranked by Fitness

Individual ID	CH Encoding (Binary)	CHs (Node IDs)	Final Fitness Score
I93	[0 1 0 1 0 0 1 0 0 0]	N2, N4, N7	0.831
I56	[1 0 0 1 0 1 0 0 0 0]	N1, N4, N6	0.825
I78	[0 0 1 0 1 0 1 0 0 0]	N3, N5, N7	0.822
I88	[0 1 0 1 1 0 0 0 0 0]	N2, N4, N5	0.818
I45	[0 1 1 0 0 0 1 0 0 0]	N2, N3, N7	0.815

The individual with the highest fitness (I93) is chosen as the final CH configuration. This CH set will then be used to form clusters in the network.

To ensure energy fairness and minimize delay, the final CH selection score is calculated using a tie-breaker equation when two individuals have similar fitness scores:

$$\text{CH\_Score}=\alpha \cdot \text{Avg(RE)}+\beta \cdot \frac{1}{\text{Avg(ICD)}} \tag{2}$$

where,

α, β are weighting constants (e.g., 0.6 and 0.4)

Avg(RE) = Average residual energy of selected CHs

Avg(ICD) = Average intra-cluster distance

This ensures CHs not only achieve high fitness but are also energy-resilient and well-positioned.

3.8 CLUSTER FORMATION AND DATA TRANSMISSION

Once the final CHs are selected and broadcasted, the cluster formation phase begins. Each non-CH node evaluates the signal strength or Euclidean distance to nearby CHs and joins the one with the lowest communication cost (typically shortest distance). The Table.10 displays a distance matrix between normal nodes and selected CHs (N2, N4, N7).

Table.10. Distance Matrix (Node to CH)

Node ID	Distance to N2 (m)	Distance to N4 (m)	Distance to N7 (m)
N1	18.0	12.5	25.0
N3	10.3	19.8	15.2
N5	21.2	10.1	16.7
N6	22.0	11.3	9.0
N8	14.7	20.3	13.5

Each node chooses the CH with the minimum distance, as shown in Table.11.

Table.11. Final Cluster Assignment Based on Nearest CH

Node ID	Closest CH	Assigned Cluster
N1	N4	Cluster 2
N3	N2	Cluster 1
N5	N4	Cluster 2
N6	N7	Cluster 3
N8	N7	Cluster 3

After clustering, the data transmission begins. Each normal node sends its sensed data to the CH. The CH aggregates the data (to reduce redundancy) and forwards it to the base station (BS), either directly or through multi-hop communication if needed. To evaluate the performance, Table.12 summarizes cluster characteristics and load distribution.

Table.12. Cluster Statistics and CH Workload

Cluster	CH Node	Member Nodes	Avg. Member Distance (m)	CH Residual Energy (J)
1	N2	1	10.3	1.6
2	N4	2	11.3	1.5
3	N7	2	11.2	1.7

This phase continues in rounds, and the CHs are re-selected periodically using the same AMCE-LS process to adapt to energy depletion and ensure network longevity.

4. RESULTS AND DISCUSSION

The proposed AMCE-LS was implemented and evaluated using the MATLAB R2022b simulation environment, a widely used tool for WSN simulation due to its strong numerical capabilities and visualization support. The experiments were conducted on a system with the following specifications: Intel Core i7 11th Gen processor, 16 GB RAM, and Windows 11 OS. MATLAB is Communication and Wireless Sensor Network Toolboxes were employed for modeling node behavior and communication energy. To validate the effectiveness of AMCE-LS, it was compared against three benchmark algorithms:

- LEACH (Low Energy Adaptive Clustering Hierarchy) – A widely used probabilistic CH selection protocol.
- PSO-CH (Particle Swarm Optimization for Cluster Head selection) – A metaheuristic-based method using particle dynamics.
- EECDA (Energy-Efficient Clustering and Data Aggregation) – A residual-energy-based clustering protocol.

These methods were evaluated using consistent parameters and simulation scenarios. The performance was assessed in terms of energy efficiency, network lifetime, stability, and clustering quality.

Table.13. Simulation Parameters for AMCE-LS and Comparative Methods

Parameter	Value
Number of sensor nodes (N)	100
Simulation area	100 m × 100 m
Base Station location	(50, 150)
Initial energy per node	2 Joules
Data packet size	4000 bits
Control packet size	200 bits
Transmission energy (E_tx)	50 nJ/bit
Receiving energy (E_rx)	50 nJ/bit
Data aggregation energy (E_DA)	5 nJ/bit/signal
Amplifier energy (ε_fs, ε_mp)	10 pJ/bit/m², 0.0013 pJ/bit/m⁴
Number of simulation rounds	500
CH update interval	Every 20 rounds
Number of sub-populations	3
Population size	30 individuals per sub-population

4.1 CH DISTRIBUTION BALANCE

This metric assesses how evenly the CHs are distributed spatially in each round. A balanced distribution prevents cluster overlap and avoids node isolation. AMCE-LS is multifactor encoding ensures well-dispersed CHs throughout the field.

Table.14. Network Lifetime Comparison (in Rounds)

Number of Nodes	LEACH	PSO-CH	EECDA	Proposed AMCE-LS
20	450	520	540	**610**
40	630	710	740	**820**
60	780	860	900	**1025**
80	870	970	1010	**1160**
100	920	1015	1070	**1290**

Table.15. Stability Period (Rounds until First Node Dies)

Number of Nodes	LEACH	PSO-CH	EECDA	Proposed AMCE-LS
20	120	160	180	**210**
40	190	230	260	**300**
60	240	290	330	**390**
80	270	320	370	**445**
100	310	360	410	**500**

Table.16. Average Residual Energy (Joules at Round 500)

Number of Nodes	LEACH	PSO-CH	EECDA	Proposed AMCE-LS
20	0.18	0.24	0.27	**0.34**
40	0.21	0.28	0.31	**0.40**
60	0.23	0.31	0.36	**0.46**
80	0.25	0.33	0.39	**0.52**
100	0.27	0.36	0.42	**0.58**

Table.17. Throughput (Total Packets Delivered to BS)

Number of Nodes	LEACH	PSO-CH	EECDA	Proposed AMCE-LS
20	2100	2600	2900	**3250**
40	4100	4800	5100	**5850**
60	5900	6700	7100	**8050**
80	7400	8200	8800	**10120**
100	8900	9800	10400	**12050**

Table.18. CH Distribution Balance (Standard Deviation of Cluster Sizes) (*Lower values indicate better balance*)

Number of Nodes	LEACH	PSO-CH	EECDA	Proposed AMCE-LS

20	2.8	2.3	1.9	**1.3**
40	3.2	2.7	2.2	**1.6**
60	3.5	3.0	2.5	**1.8**
80	3.9	3.4	2.8	**2.0**
100	4.3	3.7	3.1	**2.2**

The experimental results clearly demonstrate the superior performance of the proposed Adaptive Multifactor Co-Evolutionary Algorithm with Local Search (AMCE-LS) over existing methods—LEACH, PSO-CH, and EECDA—across multiple performance metrics and network sizes.

One of the most significant improvements observed is in the network lifetime. AMCE-LS achieved an average lifetime of 1290 rounds for 100 nodes, compared to 1070 rounds for EECDA, 1015 rounds for PSO-CH, and 920 rounds for LEACH. This translates to a 20.6% improvement over EECDA, 27.1% over PSO-CH, and a remarkable 40.2% increase over LEACH. This is primarily due to the algorithm is adaptive weight tuning, which ensures CHs are selected based on residual energy, intra-cluster distance, and load balancing.

In terms of stability period (first node death), AMCE-LS extended the time to FND to 500 rounds for 100 nodes, while EECDA, PSO-CH, and LEACH recorded 410, 360, and 310 rounds respectively. This marks a 22% increase over EECDA and 61.3% over LEACH. The stability gain implies more reliable early-stage data collection and longer sustainable sensing coverage.

The average residual energy retained at round 500 in AMCE-LS was 0.58 J, compared to 0.42 J (EECDA), 0.36 J (PSO-CH), and 0.27 J (LEACH). This showcases a 38% energy efficiency gain over EECDA and more than 114% over LEACH, attributed to energy-aware encoding and refined local search that minimizes redundant transmissions and distributes energy loads more effectively.

For throughput, AMCE-LS achieved 12,050 packets delivered at round 500 for 100 nodes, surpassing EECDA (10,400), PSO-CH (9,800), and LEACH (8,900). This equates to a 15.9% increase over EECDA and 35.4% over LEACH, indicating that AMCE-LS ensures sustained and high-quality data delivery to the base station through optimized CH placement and consistent aggregation policies.

Lastly, in CH distribution balance, the standard deviation of cluster sizes for AMCE-LS was 2.2 at 100 nodes, compared to 3.1 (EECDA), 3.7 (PSO-CH), and 4.3 (LEACH). This reflects a 29% improvement over EECDA and nearly 49% better balance over LEACH. Balanced clustering reduces communication bottlenecks and prevents CH overload, leading to prolonged node lifetimes.

Thus, the multifactor co-evolutionary optimization coupled with adaptive weight adjustment and local search refinement in AMCE-LS results in significant and consistent improvements across all key performance metrics, especially in dynamic WSN environments with energy and topology constraints.

5. CONCLUSION

AMCE-LS method is an easy way to pick a CH in WSN. Using multifactor co-evolutionary optimization, adaptive weight adjustment, local search enhancement, and other similar ideas, the system solves large problems including uneven energy use, early node death, and incomplete CH distribution. AMCE-LS always exceeds earlier methods like LEACH, PSO-CH, and EECDA in all the critical areas, such as network lifetime, stability period, energy retention, throughput, and CH balance. This is true no matter how big the network is or what factors are being looked at. AMCE-LS is a superb design for long-term and mission-critical networks because it has a 20% longer network lifetime, better residual energy, and data transmission rates that keep the same throughout the simulation. It seems like AMCE-LS could be a nice place to start when making sensor networks that use less energy and can grow.

REFERENCES

[1] Z. Cui, T. Zhao, L. Wu, A.K. Qin and J. Li, "Multi-Objective Cloud Task Scheduling Optimization based on Evolutionary Multi-Factor Algorithm", *IEEE Transactions on Cloud Computing*, Vol. 11, No. 4, pp. 3685-3699, 2023.

[2] T. Zhao, L. Wu, D. Wu, J. Li and Z. Cui, "Multi-Factor Evolution for Large-Scale Multi-Objective Cloud Task Scheduling", *KSII Transactions on Internet and Information Systems*, Vol. 17, No. 4, pp. 1100-1122, 2023.

[3] D. Zuo and Y. Lu, "Design of Amorphous Alloy Composition and Optimization of Vacuum Die Casting Process Parameters based on Co-evolutionary Algorithm", *IEEE Access*, Vol. 12, pp. 123883-123896, 2024.

[4] A.J. Wilson, W.S. Kiran, A.S. Radhamani and A.P. Bharathi, "Optimizing Energy-Efficient Cluster Head Selection in Wireless Sensor Networks using a Binarized Spiking Neural Network and Honey Badger Algorithm", *Knowledge-based Systems*, Vol. 299, pp. 1-11, 2024.

[5] R. Somula, Y. Cho and B.K. Mohanta, "SWARAM: Osprey Optimization Algorithm-based Energy-Efficient Cluster Head Selection for Wireless Sensor Network-based Internet of Things", *Sensors*, Vol. 24, No. 2, pp. 1-19, 2024.

[6] A. Jalili, M. Gheisari, J.A. Alzubi, C. Fernández-Campusano, F. Kamalov and S. Moussa, "A Novel Model for Efficient Cluster Head Selection in Mobile WSNs using Residual Energy and Neural Networks", *Measurement Sensors*, Vol. 33, pp. 1-6, 2024.

[7] N.R. Saadallah, S.A. Alabady and F. Al-Turjman, "Energy-Efficient Cluster Head Selection Via Genetic Algorithm", *Al-Rafidain Engineering Journal*, Vol. 29, No. 1, pp. 12-25, 2024.

[8] V. Prakash, D. Singh, S. Pandey, S. Singh and P.K. Singh, "Energy-Optimization Route and Cluster Head Selection using M-PSO and GA in Wireless Sensor Networks", *Wireless Personal Communications*, pp. 1-26, 2024.

[9] Y. Patidar, M. Jain and A.K. Vyas, "Optimal Stable Cluster Head Selection Method for Maximal Throughput and Lifetime of Homogeneous Wireless Sensor Network", *SN Computer Science*, Vol. 5, No. 2, pp. 1-7, 2024.

[10] S.K. Chandrasekaran and V.A. Rajasekaran, "Energy-Efficient Cluster Head using Modified Fuzzy Logic with WOA and Path Selection using Enhanced CSO in IoT-Enabled Smart Agriculture Systems", *The Journal of Supercomputing*, Vol. 80, No. 8, pp. 11149-11190, 2024.

[11] D. Gupta, J.V.N. Ramesh, M.K. Kumar, F.Y. Alghayadh, S.B. Dodda, T.A. Ahanger and S.R. Karumuri, "Optimizing

Cluster Head Selection for E-Commerce-Enabled Wireless Sensor Networks", *IEEE Transactions on Consumer Electronics*, Vol. 70, No. 1, pp. 1640-1647, 2024.

[12] S. El Khediri, A. Selmi, R.U. Khan, T. Moulahi and P. Lorenz, "Energy Efficient Cluster Routing Protocol for Wireless Sensor Networks using Hybrid Metaheuristic Approache's", *Ad Hoc Networks*, Vol. 158, pp. 1-5, 2024.

[13] S.D. Mishra and D. Verma, "Energy-Efficient and Reliable Clustering with Optimized Scheduling and Routing for Wireless Sensor Networks", *Multimedia Tools and Applications*, Vol. 83, No. 26, pp. 68107-68133, 2024.

[14] M.K. Roberts, P. Ramasamy and F. Dahan, "An Innovative Approach for Cluster Head Selection and Energy Optimization in Wireless Sensor Networks using Zebra Fish and Sea Horse Optimization Techniques", *Journal of Industrial Information Integration*, Vol. 41, pp. 1-9, 2024.

[15] M.S. Nidhya, E. Gurumoorthi, M. Babu, A.C. Charumathi, M. Naved and B. Maram, "Cluster Head Selection Algorithm using Machine Learning", *Proceedings of International Conference on Sentiment Analysis and Deep Learning*, pp. 1452-1455, 2025.