Hindawi
Wireless Communications and Mobile Computing
Volume 2022, Article ID 8911651, 21 pages
https://doi.org/10.1155/2022/8911651

Research Article

Metaheuristic Load-Balancing-Based Clustering Technique in Wireless Sensor Networks

Sandip K. Chaurasiya,[1] Arindam Biswas,[2] Prasit Kumar Bandyopadhyay,[3] Amit Banerjee,[4] and Rajib Banerjee[5]

[1]*University of Petroleum & Energy Studies (UPES), Department of Cybernetics, School of Computer Science, Energy Acres Building Bidholi, Dehard un-248007, Uttarakhand, India*
[2]*School of Mines, Kazi Nazrul University, Asansol, West Bengal, India*
[3]*Department of Electronics & Communication Engineering, School of Engineering, Sister Nivedita University, DG 1/2, New Town, Action Area 1, Kolkata, West Bengal, India*
[4]*Physics Department, Bidhan Chandra College, Asansol, 713303 West Bengal, India*
[5]*Department of Electronics & Communication Engineering, Dr. B. C. Roy Engineering College, Durgapur, West Bengal, India*

Correspondence should be addressed to Arindam Biswas; mailarindambiswas@yahoo.co.in
and Amit Banerjee; amitbanerjee.nus@gmail.com

Received 11 August 2021; Accepted 6 December 2021; Published 21 January 2022

Academic Editor: Rajesh Kaluri

The resource-constrained nature of wireless sensor networks engenders the development of energy-efficient network operations. To mitigate the prime concern of developing an energy-efficient network, clustering of the nodes has emerged as a very effective tool. If executed intelligently, clustering can not only help in obtaining even load distribution among the network nodes but also help in having the enhanced network lifetime and scalability. In this work, a Metaheuristic Load-Balancing-Based Clustering Technique (MLBCT) in wireless sensor networks has been proposed which formulates the energy-balanced clusters based on the differential evolution technique to improve the network lifetime. To ensure the formation of balanced clusters, several metrics like nodes' proximity, nodes' distribution, and energy distribution across the sensing field have been considered. Moreover, to facilitate the even load distribution among the cluster members, a randomized rotation of cluster head is implemented. The supremacy of the proposed scheme is confirmed through an extensive set of simulations against the state-of-art schemes. Simulation results reflect an average gain of 51.85% in network lifetime under the variable network configurations in an ideal environment. Moreover, a thorough statistical analysis is performed to prove the efficacy of the proposed fitness function by obtaining confidence intervals under two different network scenarios with variable node counts.

1. Introduction

A wireless sensor network (WSN) comprises a large number of tiny devices capable of sensing the surrounding, processing the collected data as per the application, and communicating the processed field information to the centralized base station (BS) [1]. However, the sensor nodes deployed (either randomly or deterministically) in the sensing field suffer from several constraints. They are limited in processing abilities, storage abilities, power, and other allied restrictions [2]. Among all these restrictions, limited power is the most severe one as the node drained of all the energy and frequent recharging and replacement cannot be facilitated, especially in remote applications of WSN like habitat monitoring, environmental monitoring, industrial monitoring, and military surveillance systems [3, 4].

Typically, transmission and route allocation consume most of the nodes' energy and are very much responsible for the power drainage of the sensor nodes. Thus, to solve this issue, energy-efficient network layer operations have been targeted by researchers for many years. Routing is the main functionality of the network layer, and hence, designing an energy-efficient routing protocol is consistently captivating the attention of the community. To the aforementioned, clustering has evolved as a very significant tool that not only eases the task of routing and distributes the load evenly within a cluster but also, through the use of data aggregation, results in substantial saving of nodes' energy to be consumed in other significant network operations.

Clustering has been defined as the grouping of nodes based on some common attributes. In a clustering-based architecture, the network nodes are partitioned into some groups termed clusters. Within the cluster, a node is designated as cluster head (CH) which carries out more energy heavy tasks such as data aggregation and long-distance communication to the sink on behalf of the entire cluster. The rest of the nodes, called cluster members, perform the basic task of sensing and short-distance communication to the CH [5]. To effectively improve the WSN performance, balancing the clusters is a prerequisite. Thus, the formation of clusters in the WSN can be seen as an optimization problem involving multiple variables to be brought into consideration like nodes' proximity, nodes' residual energy, and size of the tentative clusters. The optimization problems can be classified into two major categories—heuristic and metaheuristic.

The primary motivation behind this work is to pursue the problem of clustering through metaheuristic algorithms. As mentioned above, since the formation of balanced clusters leading to the energy-efficient network operation requires the adequate consideration of various parameters such as nodes' proximity and cluster size, optimization techniques can help a lot in having a suitable solution. With the obtainment of balanced clusters and rotation of cluster head's role among the nodes over the network rounds, the foremost goal of network lifetime improvement can be achieved effectively. In this paper, a novel energy-efficient clustering protocol, Metaheuristic Load-Balancing-Based Clustering Technique (MLBCT), is proposed for the wireless sensor networks based on the idea of differential evolution, a metaheuristic technique. The proposed scheme defines a suitable fitness function to formulate the balanced network partitioning. Once the clusters are finalized, the scheme freezes those and enables the CH-role rotation among the cluster members. To prove the scheme's efficacy, an extensive set of simulations demonstrate the showcasing of the improved network lifetime and network energy consumption.

1.1. Major Contributions and Organization of the Paper. The major contributions of the proposed MLBCT are as follows:

(i) Design of an appropriate fitness function leading to

  (a) balanced cluster formation

  (b) reduced intracluster communication cost

(ii) Development of a differential evolution-based energy-efficient clustering scheme on the basis of the devised fitness function

(iii) Performance analysis of the proposed scheme, MLBCT

  (a) under varying network configurations to showcase its adaptability and scalability

  (b) with comparison to the state-of-art schemes in terms of network performance

  (c) with statistically justified results

The rest of the paper is organized into five descriptive sections. Section 2 outlines the literature review of the existing works in the same context to identify the technical gaps. Section 3 presents the adopted network model, an introductory discussion on differential evolution, and the terminology to be used throughout the work. Section 4 describes the proposed scheme detailing each of its constituent phases. Section 5 discusses the performance in detail to confirm the supremacy and efficacy of the MLBCT, and finally, Section 6 concludes the work by mentioning the future scope for the same.

2. Literature Review

As mentioned in Section 1, the optimization techniques can be majorly categorized as heuristic and metaheuristic schemes. Heuristic techniques utilize the complete set of particulars of a given problem and, being greedy in nature, generate solutions that might get trapped into local maxima/minima instead of producing the global maxima/minima.

On the other hand, metaheuristic techniques, also termed guided random search algorithms, are problem-independent, providing the optimal solution without getting stuck into the local maxima/minima. Metaheuristic algorithms compute the optimal solution by thoroughly exploring and exploiting the available search space in multiple iterations. The general working of the metaheuristic techniques is summarized in Figure 1.

The metaheuristic scheme starts working with a randomly selected set of solution vectors that improve over the iteration. Once the application-specific parameters such as scaling factor and crossover rate are defined, the fitness of the current solution set is evaluated through a carefully designed fitness function. Then, the counter which keeps track of the iterations is initialized. Afterward, a selection from the population chosen is made, and the selected vectors undergo a variation phase (mutation/crossover). Thus, updated vectors are again evaluated for their current fitness, and through a survivor function, a greedy selection strategy, the population for the next generation is finalized. The process of updating the set of solutions is repeated for a predefined number of iteration, and at last, the most recent population is selected as the final solution. An intelligently and carefully designed fitness function plays the most significant role in obtaining further improved offspring in metaheuristic techniques.

Here, we present a brief review of such schemes based on the approaches known as heuristic and metaheuristic.

2.1. Heuristic Schemes. In one work [6], the authors proposed the most popular clustering-based routing protocol, Low-Energy Adaptive Clustering Hierarchy (LEACH), for the wireless sensor networks, which features a probabilistic

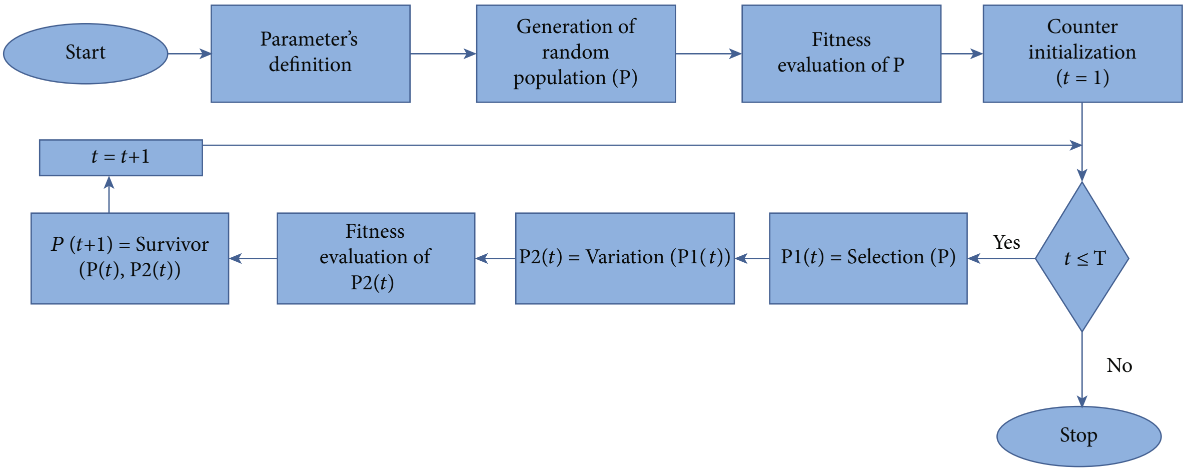

Start
Parameter's definition
Generation of random population (P)
Fitness evaluation of P
Counter initialization (t = 1)
$t = t+1$
$t \leq$ T
Yes
No
Stop
P1(t) = Selection (P)
P2(t) = Variation (P1(t))
Fitness evaluation of P2(t)
P (t+1) = Survivor (P(t), P2(t))

Figure 1: General scheme of metaheuristic techniques.

selection of cluster heads. It implements the localized coordination for various network operations and randomized rotation of the role of the cluster heads for load balancing among the nodes. However, since the selection of cluster heads does not count the residual energy of the nodes, nodes with low residual energy might suffer from early death if frequently selected as cluster heads.

In another work [7], the authors of the LEACH proposed an extension of the [6] requiring the nodes to send their location and energy status to the base station for the selection of cluster heads in a centralized manner and the formation of appropriate clusters via the application of simulated annealing algorithm.

The authors proposed a chain-based scheme in which, instead of forming multiple clusters [8], the nodes were provisioned to develop chains in a way that each sensor could exchange data with the neighbor nodes. At last, the chain leader concludes the entire data flow and forwards it to the base station. However, the scheme proved to be more energy-efficient than LEACH, but the significant delay in the delivery and dynamic topological adjustments appeared as the major issues of the scheme.

In [9], the authors proposed a static clustering scheme that eradicated the energy costing of the dynamic cluster formation in every round of the network operation as in LEACH, etc. In this scheme, distance-based clustering is executed via the base station. Once the clusters are decided, two important parameters—residual energy of the nodes and the nodes' spatial distribution—are considered to select cluster heads. However, the scheme only targeted energy consumption minimization.

In one scheme [10], the authors proposed a centralized scheme that treated coverage in the sensing field as equally important as the energy efficiency. The scheme starts with the distance-based clustering as in the [9]. It selects the cluster heads based on the weighted mean of the contribution factor of the nodes, where the contribution factor is defined as the ratio of the node's residual energy to that of the native grid in the sensing field. The main objective of the scheme is to assure network-wide coverage for the maximum network operation time.

In [11], the authors proposed a LEACH-based clustering protocol that mainly targets the energy efficiency and the fault tolerance in the network. To improve the network lifetime, the network nodes are provisioned to send their data to their respective cluster heads only when the current data is distinct from the previous data. At the end of every network round, noncluster head nodes forward their current energy status to the respective cluster heads to get classified as faulty (nodes with lower residual energy level) and live nodes (nodes with sufficient residual energy). The identification of faulty nodes facilitates the fault tolerance in the network.

In [12], the authors proposed a Fault-Tolerant Clustering-based Multipath algorithm (FTCM) to address the problems of energy efficiency and fault tolerance in the wireless sensor networks. The scheme calls the hybrid energy-efficient distributed clustering (HEED) [13] scheme to partition the network into an appropriate number of clusters. It also appoints a backup CH (BCH) for a cluster head to improve the fault tolerance. The BCH consistently monitors the performance of CH and keeps a copy of CH's data until delivered to the base station. In case of any mishap at the CH end, the BCH can instantly transmit data to the base station without asking the member nodes to send their data again. In addition to the regular responsibilities of CH, the CH is also responsible for the removal of the majority of faulty nodes via hypothesis testing and majority voting. The proposed scheme enables three paths to transfer data from the source node to the base station based on the parameters—residual energy of the nodes, number of hops, propagation speed, and path reliability.

In [14], the authors proposed a clustering-based Hierarchical Fault-Management Framework (HFMF) to address energy management and fault management jointly. For the minimization of energy consumption, the sleep/active method is used. For the management of faults, that is, faults' detection and recovery, backup CH (BCH) is appointed along with every CH to take care of acting CH in the event of its malfunctioning or failure. Later by measuring the data correlation among the cluster members, nodes are grouped virtually to further achieve the energy and fault management. The authors have successfully demonstrated that the proposed scheme not only manages the transient faults, intermittent faults, and permanent hardware faults but also the link faults are detected.

2.2. Metaheuristic Schemes. A wide variety of metaheuristic techniques such as genetic algorithm (GA), genetic programming (GP), evolutionary programming (EP), evolution strategies (ES), differential evolution (DE), particle swarm optimization (PSO), ant colony optimization (ACO), and teaching-learning-based optimization (TLBO) exist in the literature. Such metaheuristic techniques with the virtue of being problem-independent have already imparted a lot in almost every field of engineering like [15]. In the context of wireless sensor networks, some contributions are noticed especially for the selection of cluster heads and the effective formulation of the clusters like in [16–25].

Due to its simplicity, robustness, and fast convergence, differential evolution has proved its worth over the algorithms like GA and PSO [26]. Several contributions have already been proposed based on this outstanding differential evolution technique in search of suitable clusters of the nodes in WSN. This subsection discusses some of the prime contributions in this regard as follows:

In one work [27], a differential evolution-based routing scheme, DE-LEACH, is proposed for environmental monitoring wireless sensor networks. DE-LEACH applies the fast and straightforward converging search technique of differential evolution to produce the clusters by considering the nodes' residual energy status and spatial distribution. The scheme consists of four phases: partitioning initial clusters, collecting status information of the nodes within the clusters through the auxiliary cluster heads, determining optimized cluster heads with differential evolution, and forming optimized clusters. The phases are to be executed in every round of the network operation. The scheme outperforms the traditional LEACH, and LEACH-C [7]. However, the nodes are burdened with heavy computational responsibilities.

In another work [28], a differential evolution-based clustering algorithm (DECA) is proposed, which provisions specialized nodes enriched with the additional amount of initial energy to act as cluster heads. These specialized nodes are called relay nodes or gateways. In DECA, besides providing a suitable fitness function (to measure the health of the tentative clusters), a new local improvement phase has also been proposed that carefully prevents early death of the gateways. DECA utilizes the DE/best/1/bin scheme for the differential evolution. In addition to a novel scheme for the vector representation, a fitness function is designed by considering the standard deviation of the lifetime of gateways and average cluster distance. The scheme outperforms the [29–31] traditional differential evolution and genetic algorithm-based scheme in terms of network lifetime; however, the scheme gives only a little attention to the cluster balancing via its local improvement phase.

A hybrid differential evolution and simulated annealing (DESA) scheme for the improvement of network lifetime in wireless sensor networks is proposed in [32]. The scheme utilizes a hybrid of differential evolution and simulated annealing for local and global optimal solutions, respectively. There are four phases in the scheme—population vector initialization, mutation, crossover, and selection as in the traditional differential evolution. However, instead of using a random selection of population vectors, a more effective, "opposite point method" [33] technique is used for the initialization of population vectors. The mutation scheme is decided randomly at run time based on a chosen threshold value (here, it is 0.5) in such a way that a random number belonging to (0, 1) is observed, and if it is below the threshold, the mutation scheme is DE/rand/1; otherwise, it is DE/target – to – best/1. The fitness function is designed by considering the ratio of nodes' energy to that of the respective clusters. And for crossover, a blending rate based on Gaussian distribution is used. The scheme outperforms the traditional differential evolution scheme in terms of network lifetime, energy consumption, throughput, etc.; however, it converges slowly.

In [34], the authors proposed Multiobjective Load-Balancing Clustering (MLBC) which is a multiobjective optimization technique that addresses two significant problems in WSN—energy efficiency and reliability. It utilizes the Multiobjective Particle Swarm Optimization (MOPSO). MLBC targets energy efficiency by appropriately considering the average residual energy of the cluster heads and reliability by reducing the intercluster communication cost among the nodes in a cluster. It also provisions the load balancing via shuffling the roles of the next-hop node and CH in every iteration. However, it considers only the average residual energy of cluster heads in formulating the objective function for energy efficiency.

In a scheme [35], efficient energy consumption in wireless sensor networks using an improved differential evolution algorithm is highlighted. The scheme is an improvement of [28], in which the mutation strategy has been updated to accommodate the target vector along with the prior best and two random population vectors. Also, the fitness function has been upgraded to accommodate the total energy of the gateways and nodes in addition to the existing network lifetime standard deviation component. However, nothing has been mentioned concerning the load balancing among the clusters.

In one work [36], the authors proposed a hybrid metaheuristic clustering algorithm that exploits the best of Artificial Bee Colony and differential evolution optimization techniques. In their proposed Artificial Bee Colony (ABC) with differential evolution (DE) scheme, known as ABC-DE-based clustering scheme, the objective function is designed by taking into account the three network parameters—average intracluster distance, average energy of cluster heads, and data transmission delay to ensure the load-balanced cluster heads. In addition to this, an ABC-based metaheuristic algorithm has also been proposed to facilitate the dynamic repositioning of the mobile sink within the cluster-based network to achieve further energy efficiency.

In [37], the authors have addressed the problem of energy optimization in an Internet-of-Thing-based WSN (IoT-based WSN). In pursuance of the problem, as mentioned earlier, a hybrid of the Whale Optimization

Algorithm (WOA) and simulated annealing (SA) metaheuristic algorithms have been employed to select the most suitable cluster heads in their respective clusters. For choosing the most appropriate cluster heads, the fitness function of the proposed scheme considers a set of five node-specific parameters: residual energy, load, delay, distance, and temperature. The fitness function ensures that the node with the highest residual energy but the least load, delay, distance, and temperature is selected as the cluster heads in every network round.

In one work [38], the authors proposed an Artificial Intelligence- (AI-) based quorum system to address the issue of energy conservation in the wireless sensor networks. The primary motivation behind the proposed AI-based was to fasten the neighbor discovery process in order to minimize the network latency. Moreover, the scheme facilitates a quorum-based grid system that allows a substantial increase in the number of nodes in the quorum without mandating the increase in the number of quorums to reduce the effective network delay. In addition to the aforesaid, the feature of weighted load balancing reduces the network energy consumption to improve the network lifetime. Through the various experimentation, the authors have established the outperformance of their proposed scheme over the state-of-the-art quorum algorithms in terms of latency, improved coverage, energy efficiency, and network lifetime.

In [39], the authors proposed a genetic algorithm- (GA-) inspired clustering-based approach to address the problem of node's localization in wireless sensor networks. To find the accurate position of unknown nodes with respect to the anchors or known nodes, the authors used the Euclidean distance objective function in their proposed scheme. Through various simulation results, the supremacy of the GA-based localization scheme with an extended clustering approach has been established over the state-of-the-art schemes like Centroid and Distance Vector-Hop (DV-Hop) in terms of improved location accuracy.

In a scheme [40], the author proposed a genetic algorithm-based energy-efficient clustering scheme which addressed the localization problems in wireless sensor networks. The authors utilized parameters like node's residual energy, distance estimation, and coverage connection in the formulation of fitness function for their proposed scheme, Energy-Efficient Clustering in Genetic Algorithm Localization (EECGL). Through various experimentation, the authors have shown that EECGL approximates the unknown node's location with the least localization error and extends the effective network lifetime by minimizing the overall network energy consumption.

In a work [41], the authors proposed a metaheuristic energy-efficient clustering technique which is inspired by the Brain Storm Optimization (BSO). The BSO is a swarm-based metaheuristic technique exploiting the human brainstorming process in search of the best possible solutions. In their proposed scheme, Energy-Efficient Clustering-Brain Storm Optimization (EEC-BSO), the authors have focused on deciding energy-efficient clusters in a way that nonparticipating nodes in the information transmission process are sent to sleep mode minimizing the overall network consumption. In the formulation of such clusters, the fitness function is designed by considering the parameters like node's residual energy, coverage, and packet data rate. Moreover, the outperformance of EEC-BSO has been established over the state-of-the-art schemes such as LEACH, LEACH-Centralized, Energy-Efficient Clustering Scheme (EECS), and LEACH-BSO in terms of reduced energy consumption, improved coverage, and data packet rate.

In a proposed scheme [42], a differential evolution-based clustering routing protocol (DEBCRP) for wireless sensor networks. DEBCRP is a base station-dependent scheme that applies DE/best/1/bin scheme for the network partitioning into some clusters. The fitness function devised by the authors considers the nodes' residual energies with respect to the probable cluster heads and the distance between the nodes and the cluster heads for the formulation of clusters. At last, to communicate the data from the sensing field to the base station, a PEGASIS [8] like a chain of the cluster heads is formed. The scheme DEBCRP is reported to outperform the S-DE [43] in terms of network lifetime. However, no adequate consideration is given for the formulation of load-balanced clusters, which is the most prime key to network lifetime improvement. Also, PEGASIS like chain of the cluster heads suffers from similar problems as in [8], for example, delayed communication, and since data from one CH is to be aggregated with that of the others in the direction to the sink, there might be introduced some inaccuracy in the information being sent to the base station.

From the aforementioned analysis, it can be easily concluded that despite being the most important factor for the formulation of clusters in the network, cluster balancing has been addressed the least. Thus, the work being presented here serves the following objectives:

(i) Balanced cluster formulation to contribute effectively towards the enhancement of network lifetime

(ii) Adaptable clustering solution to perform consistently well in any network configuration

3. Preliminaries

This section describes the network model for the scheme. In addition to this, it also discusses the basics of the differential evolution metaheuristic technique and the entire set of notations used throughout the work.

3.1. Network Model. MLBCT assumes the wireless sensor network with the following characteristics:

(1) All the sensor nodes are deployed randomly across the sensing field and are static. More illustratively, nodes once deployed cannot change their location

(2) The sensor nodes are homogeneous and equipped with a definite amount of initial energy

(3) The sensor nodes are facilitated with the power control features to introduce variations in the transmission power as and when needed

(4) The base station is also static and can be placed at any point in the network accordingly

(5) The continuous data flow model is used here to define the working mode of the sensor nodes

3.2. Differential Evolution: An Overview. The differential evolution has evolved as a prevalent stochastic metaheuristic multimodal optimization technique over the continuous search space. Similar to the general scheme of metaheuristic techniques as discussed in Section 1, it starts with the definition of the initial parameters where the values of scaling factor and crossover rate are defined along with the randomized set of initial solutions (initial population) and the number of iterations. Here, each solution vector (equivalently known as chromosome or genome) termed as a target vector undergoes the mutation phase followed by the recombination. This mutation followed by the recombination is nothing but the variation phase of Figure 1. As depicted in Figure 2, the target vector, once it passes through the mutation phase, becomes the donor/mutant vector. After the recombination or crossover phase, the donor vector is known as the trial vector.

In the differential evolution scheme, obtainment of the next-generation solutions is performed only after the generation of all trail vectors when compared to particle swarm optimization, and teaching-learning-based optimization [44, 45]. In other words, the greedy selection towards the next-generation solution is performed between the pair of target and trial vectors once all the target vectors have been converted into trial vectors. A variety of mutation strategies exist, such as random, best, and target-to-best, along with the two types of crossover techniques—binomial and exponential crossovers. The binomial and exponential crossover can be defined as follows:

3.2.1. Binomial Crossover.

$$u_j = \begin{cases} v_j & \text{if } r \leq C_{\mathrm{P}} \text{ OR } j = \delta, \\ x_j & \text{if } r > C_{\mathrm{P}} \text{ AND } j \neq \delta, \end{cases} \quad (1)$$

where C_{P} is the crossover probability, δ is the randomly selected variable location from the set $\{1, 2, 3, \cdots, |\text{decision variable}|\}$, r is the random number between 0 and 1, u_j refers to the j^{th} variable of the trial vector, v_j refers to the j^{th} variable of donor/mutant vector, and x_j refers to the j^{th} variable of the target vector.

3.2.2. Exponential Crossover. In the exponential crossover, at very first, the n^{th} variable from the donor vector is copied into the trial vector. Afterward, every subsequent variable from the donor vector is copied into the trial vector as long as the $r \leq C_{\mathrm{P}}$. Once $r > C_{\mathrm{P}}$, variables from the target vector are copied into the trial vector.

Based on the adapted mutation strategy and crossover type, various schemes have been proposed for differential evolution, and to discriminate among them, a standard notation, **DE/x/y/z**, is used. Here, **DE** refers to the differential evolution, **x** denotes the mutation strategy, **y** denotes the number of difference vectors to be used in the mutation operation, and **z** refers to the crossover scheme selected. Some of the variants of the DE schemes are listed here in Table 1.

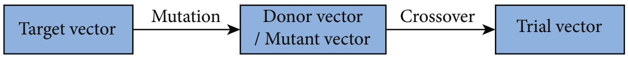


Figure 2: Vector transformation in differential evolution.

Here, in Table 1, V is the donor vector, F is the scaling factor such that $F \in (0, 2)$, X_{best} is the target vector with best fitness value, X_i is the i^{th} target vector, and X_{r_j} is the j^{th} target vector chosen randomly where $j \in [1, N]$, N being the number of target vectors in the population. Once the trail vectors are generated for all the target vectors of current generation, say G, offsprings are chosen based on the fitness value of the corresponding pairs of target and trial vectors, i.e., $< X_{i,G}, U_{i,G} >$ for $i \in [1, N]$ as follows:

$$X_{i,G+1} = \begin{cases} U_i & \text{if fitness}(U_{i,G}) \geq \text{fitness}(X_{i,G}), \\ X_i & \text{otherwise}. \end{cases} \quad (2)$$

3.3. Terminology. The notations used throughout the work have been listed as follows:

(i) S denotes the set of sensor nodes such that $S = \{s_1, s_2, s_3, \cdots, s_N\}$ where N is the number of nodes deployed in the sensing field

(ii) Θ denotes the set of cluster heads such that $\Theta = \{\mathrm{CH}_1, \mathrm{CH}_2, \mathrm{CH}_3, \cdots, \mathrm{CH}_k\}$ where k is the number of cluster heads

(iii) RE_i denotes the residual energy of the i^{th} node in the network

(iv) $\mathrm{Cluster}^i_{\mathrm{RE}}$ denotes the residual energy of the i^{th} cluster such that $\mathrm{Cluster}^i_{\mathrm{RE}} = \sum_{j=1}^{m} \mathrm{RE}_j$ where m refers to the cluster size

(v) CS_i denotes the cluster size of the i^{th} cluster

(vi) AvgCS refers to the average cluster size, i.e., average number of nodes in a cluster

(vii) ACE refers to the average cluster energy such that $\mathrm{ACE} = \sum_{i=1}^{N} \mathrm{RE}_i / k$

(viii) $d(i, j)$ denotes the Euclidean distance between the i^{th} and j^{th} nodes in the network

(ix) $\mathrm{dist}_m(i, j)$ denotes the Euclidean distance between the i^{th} and j^{th} members of the m^{th} cluster. This parameter is basically used to measure the nodes' proximity

(x) R_{C} denotes the communication range of the nodes

(xi) $\mathrm{ComCH}(s_i)$ refers to the set of cluster heads within the communication range of the node s_i, i.e., $\mathrm{ComCH}(s_i) = \{\mathrm{CH}_j \mid d(s_i, \mathrm{CH}_j) \leq R_{\mathrm{C}}\}$

Table 1: Differential evolution schemes.

DE scheme	Mutation strategy	Mutation expression	Crossover type
DE/rand/1/bin	Random	$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right)$	Binomial
DE/rand/2/exp	Random	$V = X_{r_1} + F\left(X_{r_2} - X_{r_3}\right) + F\left(X_{r_4} - X_{r_5}\right)$	Exponential
DE/best/1/bin	Best	$V = X_{\text{best}} + F\left(X_{r_1} - X_{r_2}\right)$	Binomial
DE/best/2/bin	Best	$V = X_{\text{best}} + F\left(X_{r_1} - X_{r_2}\right) + F\left(X_{r_3} - X_{r_4}\right)$	Binomial
DE/target-to-best/1/exp	Target-to-best	$V = X_i + F(X_{\text{best}} - X_i) + F\left(X_{r_1} - X_{r_2}\right)$	Exponential
DE/target-to-best/2/exp	Target-to-best	$V = X_i + F(X_{\text{best}} - X_i) + F\left(X_{r_1} - X_{r_2}\right) + F\left(X_{r_3} - X_{r_4}\right)$	Exponential

The main objective of the present work is to formulate the balanced clusters within the network for the even distribution of load among the nodes. To ensure this, it is attempted that the clusters are equipped with an almost similar count of member nodes situated close to one another. Also, the clusters are left with an approximately equal amount of residual energy at the end of every network round.

4. Proposed Scheme: Metaheuristic Load-Balancing-Based Clustering Technique (MLBCT)

This section describes the proposed scheme, Metaheuristic Load-Balancing-Based Clustering Technique (MLBCT) in wireless sensor network. The MLBCT is a base station-(BS-) assisted scheme which calls the BS for the differential evolution-based cluster formation. Once the optimized and balanced clusters come into existence, it hands over the responsibility of further network operations to the network nodes.

The scheme starts with a bootstrapping phase in which all the nodes are assigned unique IDs, which in turn communicate their IDs and location information to the BS. The BS then applies the differential evolution with a well-established fitness function (detailed below) and formulates the balanced clusters. The selected cluster heads are then informed of their specific roles and their members' information by the base station. Thus, selected cluster heads then provide their IDs to the respective members along with the TDMA schedules. Afterward, the overall network operation is divided into rounds where each round consists of the steady-state phase and the responsible node selection phase. In the steady-state phase, cluster members send their data to their respective cluster heads, which aggregate the received data and forward it to the base station. In the responsible node selection phase, the current cluster head in a cluster, select a node randomly to act as head for the next round and broadcast into the concerned cluster. The entire workflow is summarized in Figure 3 and has been detailed into the subsequent subsections and algorithm as follows:

4.1. Bootstrapping. In bootstrapping, differential evolution is applied by the base station to divide the entire network into k number of balanced clusters where k is a user-defined parameter. It starts with the sharing of node-specific information such as identity, residual energy, and location information to the base station by the nodes deployed. Based on the information received, BS performs the following to determine the required partitioning.

4.1.1. Generation of the Random Population. The population vectors are generated as per the [28]. Each population vector is chosen in such a way that it indicates the assignment of every network node to one of the cluster heads. The notation adopted to represent the i^{th} population vector of the G^{th} generation is as follows:

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \cdots, x_{N,i,G}], \tag{3}$$

where $x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \cdots, x_{N,i,G}$ are the random numbers between 0 and 1. $x_{j,i,G}$ denotes the assignment of the node s_j to one of cluster heads, say k, as follows:

$$l = \text{ceiling}\left(x_{j,i,G} * \left|\text{ComCH}\left(s_j\right)\right|\right), \tag{4}$$

$$\text{C}H_k = \text{index}\left(\text{ComCH}\left(s_j\right), l\right). \tag{5}$$

Here, the length of the population vectors is definite and determined by the number of nodes deployed in the field.

Thus, corresponding to every population vector, say $\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \cdots, x_{N,i,G}]$, we have another vector, say $\vec{Y}_{i,G} = [y_{1,i,G}, y_{2,i,G}, y_{3,i,G}, \cdots, y_{N,i,G}]$ such that

$$f\left(x_{j,i,G}\right) = y_{k,i,G}, \tag{6}$$

where $y_k \in \Theta$ is assigned to the node x_j in the i^{th} vector of G^{th} generation as per equations (4) and (5).

4.1.2. Fitness Function. It can be easily intuited that if the clusters are balanced in the clustered network architecture, they might have an almost similar level of residual energy and a similar count of member nodes. With this conception, to meet our primary objective of network partitioning into some balanced clusters, nodes' residual energy and cluster size have been taken as the decision parameters. In addition to this, nodes' proximity has also been taken into account, ensuring the reduced energy consumption in intracluster communication.

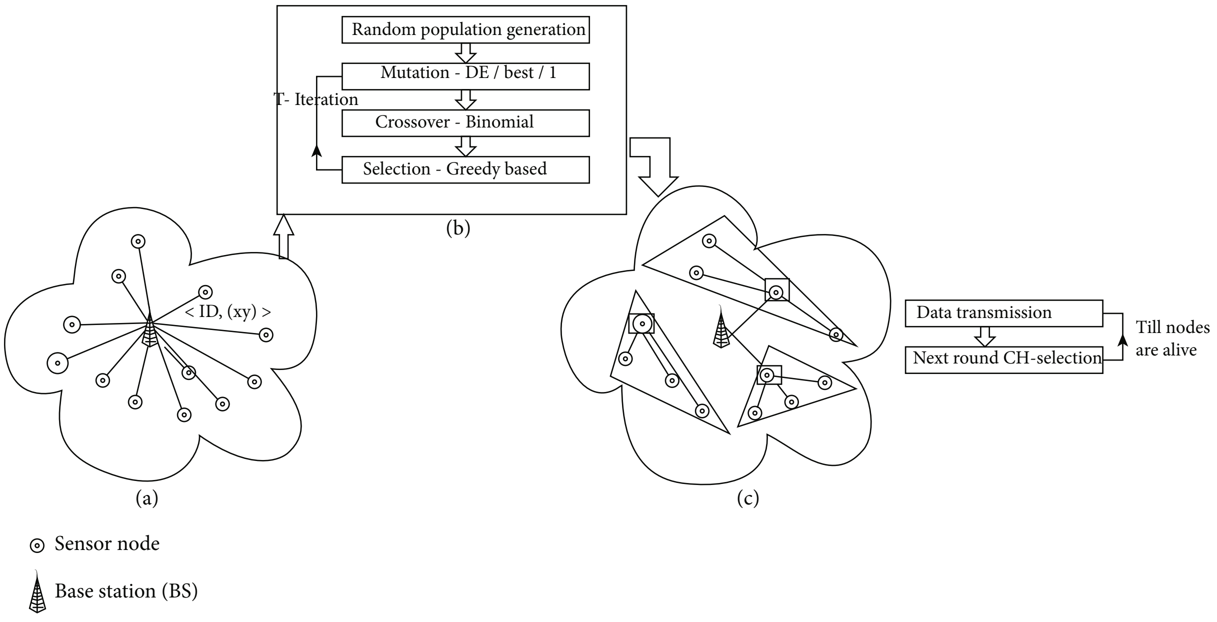


Figure 3: MLBCT: operation.

A suitable fitness function always contributes the most to the differential evolution to converge. Thus, the fitness function has been derived in such a way that it characterizes all the aforementioned requirements as follows:

(i) Standard deviation of average cluster energy

If the clusters have been formed in an optimized way, ensuring the entire network energy is distributed evenly across the clusters formed in the network, each cluster is supposed to have an almost similar level of residual energy. In other words, it can be said that in terms of average cluster energy (ACE), each cluster should have the approximately same amount of energy, and hence, the standard deviation accords to the following:

$$\sigma_{\mathrm{CE}} = \sqrt{\frac{1}{k} * \sum_{i=1}^{k} \left(\mathrm{ACE} - \mathrm{Cluster}_{\mathrm{RE}}^{i}\right)^2}, \tag{7}$$

where k is the number of clusters. It is quite obvious that the lower the value of σ_{CE}, the higher the value of fitness, i.e.,

$$\text{Fitness Value} \propto \frac{1}{\sigma_{\mathrm{CE}}}. \tag{8}$$

(ii) Standard deviation of average cluster size

The balanced clusters must have an approximately equal number of members. In other words, it can be said that the average cluster size (AvgCS) of each cluster should have the almost same count of cluster members.

With this, the standard deviation and the fitness value accord to equations (9) and (10), respectively.

$$\sigma_{\mathrm{CS}} = \sqrt{\frac{1}{k} * \sum_{i=1}^{k} \left(\mathrm{AvgCS} - \mathrm{CS}_i\right)^2}, \tag{9}$$

where k is the number of clusters. It can be intuited again that the lower the value of σ_{CS}, the higher the value of fitness, i.e.,

$$\text{Fitness Value} \propto \frac{1}{\sigma_{\mathrm{CS}}}. \tag{10}$$

(iii) Nodes' proximity within the cluster

This is the metric that ensures that when there comes to decide on the nodes to be a part of a cluster, the one who is located at a shorter distance from the other members gets priority. The central idea behind having this metric is to reduce the cost of communication within the cluster. The lower the value of this metric, the higher the value of fitness. More illustratively,

$$\text{Fitness Value} \propto \frac{1}{\sum_{m=1}^{k} \mathrm{dist}_m(i, j)}. \tag{11}$$

From equations (8), (10), and (11), we can have the following:

$$\text{Fitness Value} \propto \frac{1}{\sigma_{\mathrm{CE}}} * \frac{1}{\sigma_{\mathrm{CS}}} * \frac{1}{\sum_{m=1}^{k} \mathrm{dist}_m(i, j)}, \tag{12}$$

i.e.,

$$\text{Fitness Value} = \frac{K}{\sigma_{\text{CE}} * \sigma_{\text{CS}} * \sum_{m=1}^{k} \text{dist}_m(i, j)}, \tag{13}$$

where "K" is proportionality constant which can be set as $K = 1$ without loss of generality.

And, hence,

$$\text{Fitness Value} = \frac{1}{\sqrt{1/k * \sum_{i=1}^{k} \left(\text{ACE} - \text{Cluster}_{\text{RE}}^{i}\right)^2} * \sqrt{1/k * \sum_{i=1}^{k} (\text{AvgCS} - \text{CS}_i)^2} * \sum_{m=1}^{k} \text{dist}_m(i, j)} \tag{14}$$

or

$$\text{Fitness Value} = \frac{k}{\sqrt{\sum_{i=1}^{k} \left(\text{ACE} - \text{Cluster}_{\text{RE}}^{i}\right)^2} * \sqrt{\sum_{i=1}^{k} (\text{AvgCS} - \text{CS}_i)^2} * \sum_{m=1}^{k} \text{dist}_m(i, j)}. \tag{15}$$

4.1.3. Mutation Strategy. Like in [28, 42], **DE/best/1/bin** scheme is adapted here in this work which refers to the application of the **DE/best/1** mutation strategy. As depicted in Figure 2, each target vector of the population (say, of the size P) will go through this scheme to get transformed into a donor vector. From Table 1, the mutation expression for the selected strategy is

$$\vec{V}_{i,G} = \vec{X}_{\text{best},G} + F\left(\vec{X}_{r_1,G} - \vec{X}_{r_2,G}\right), \tag{16}$$

where $\vec{X}_{\text{best},G}$ and $\vec{X}_{r_1,G}$, $\vec{X}_{r_2,G}$ refer to the best vector, and any two randomly selected vectors from the G^{th} generation of the population such that i, r_1, and r_2 are the three random integers $\in [1, P]$ and $i \neq r_1 \neq r_2$, respectively. F is the scaling factor that may have any value between $(0, 2)$.

From equation (3), it is quite obvious that the components of the vectors in equation (16)—$\vec{X}_{\text{best},G}$, $\vec{X}_{r_1,G}$, and $\vec{X}_{r_2,G}$—are the random values $\in (0, 1)$. In order to ensure that the components of the vector $\vec{V}_{i,G}$ are also the values $\in (0, 1)$, a few amendments are being introduced as in [28].

Let

$$\vec{D}_{i,G} = \vec{X}_{r_1,G} - \vec{X}_{r_2,G}, \tag{17}$$

then,

$$d_{j,i,G} = \begin{cases} 1 + \left(x_{j,r_1,G} - x_{j,r_2,G}, & \text{if} \quad \left(x_{j,r_1,G} - x_{j,r_2,G}\right) \leq 0, \\ \left(x_{j,r_1,G} - x_{j,r_2,G}\right), & \text{otherwise.} \end{cases} \tag{18}$$

Also, for the computation of $v_{j,i,G}$ contributing to $\vec{V}_{i,G}$, the following can be referred to

$$v_{j,i,G} = \begin{cases} \left(x_{j,\text{best},G} + F * d_{j,i,G}\right) - 1, & \text{if} \quad \left(x_{j,\text{best},G} + F * d_{j,i,G}\right) > 1, \\ x_{j,\text{best},G} + F * d_{j,i,G}, & \text{otherwise.} \end{cases} \tag{19}$$

4.1.4. Crossover Scheme. The crossover schemes in terms of the binomial and exponential crossover are already described in Section 3. A binomial crossover scheme is used in this work to convert the donor vector into the trial vector.

4.1.5. Selection or Offspring Generation. Once all the trial vectors are generated following the above-mentioned steps, the next generation can be obtained on basis of the comparison of fitness values of the corresponding pair of target and trial vectors as given in

$$\vec{X}_{i,\,G+1} = \begin{cases} \vec{U}_{i,\,G} & \text{if} \quad \text{fitness}\left(\vec{U}_{i,\,G}\right) \geq \text{fitness}\left(\vec{X}_{i,\,G}\right), \\ \vec{X}_{i,\,G}, & \text{otherwise.} \end{cases} \tag{20}$$

4.1.6. Complexity Analysis. Throughout the proposed scheme, fitness function would be evaluated for $N_P + N_P * T$ times where N_P refers to the size of population and T refers to the number of iterations known a priori.

Moreover, exploiting solution space in search of the most optimal solution is a continuous process in the metaheuristic scheme. For this reason, even in the best case, the complexity of the fitness function will be $O(n^2)$ as each newly generated solution has to be compared with its predecessor in terms of its fitness value. Similarly, complexity of the fitness function in the worst case will be $O(n^2)$ due to successive fitness value computation and comparison. Thus, the average-case complexity for the fitness function can be concluded as $O(n^2)$.

As explained at the beginning of this section, once the clusters are formed, and members are notified of their respective initial heads, further network operations can be divided into two rounds—the steady-state phase and the responsible node selection phase.

4.2. Steady-State Phase. This phase refers to the data transmission in which cluster members send their data to their respective cluster heads in the designated time slots. After receiving the data from its members, cluster heads aggregate the collected data and forward it to the base station on behalf of their entire cluster.

4.3. Responsible Node Selection Phase. After executing the steady-state phase, a cluster head in its respective cluster selects a node randomly as the head for the next round and communicates the same to its members. The members note the same and communicate their data to that newly selected cluster head in the upcoming round accordingly. The process is carried out in each of the clusters in the network.

5. Performance Analysis

This section deals with the various experimental processes conducted throughout the work and analyses the obtained results thoroughly.

5.1. Experimental Environment. In conducting the experiments, different network configurations with varying node densities have been examined. More illustratively, experiments have been performed with the different number of nodes, say 50, 100, 150, and 200 in an area of $100 \times 100\,\text{m}^2$ with two different sink placements—one at the center of the sensing field (50 m, 50 m) and another beyond the network precisely at (50 m, 150 m). An instance of clustering with 50 nodes and 5 and 10 cluster heads, respectively, is demonstrated in Figure 4. The base station is situated at (50 m, 150 m) in this exemplary instance.

An extensive set of experiments have been performed for the proposed scheme using MATLAB.

Mainly, the experiments have been performed to

(1) Prove the efficacy of the proposed fitness function

In this set of experiments, the proposed fitness function as in equation (15) has been tested for the quality of clusters being produced. It has been verified that the proposed fitness function yields balanced clusters in terms of cluster size. The clusters generated as per equation (15) have been compared with the clusters produced by the fitness function given in [42] under two different clustering scenarios. The network is divided into 5 clusters and 10 clusters, respectively.

(2) Prove the supremacy of the proposed scheme, MLBCT in terms of network lifetime and network stability

In the second set of experiments, the performance of MLBCT is compared to that of DEBCRP [42] and improved differential evolution-LEACH (ImDE-LEACH) [46], majorly in terms of network lifetime and network stability with respect to the number of alive nodes in the network, network energy consumption, average residual energy per network nodes over the network rounds, and data packets delivered to the base station under the variable network configurations. Moreover, for the sake of experimentation, the performance of the LEACH [6] has also been recorded into the same context as that of MLBCT, DEBCRP, and ImDE-LEACH.

5.2. Simulation Parameters. To compare the performance of the proposed scheme, MLBCT, with that of DEBCRP and ImDE-LEACH, simulation parameters have been adopted here as listed in Table 2. However, to prove the scalability and adaptability of the proposed scheme, the performance has also been tested under variable network configurations.

In addition to the parameters listed in Table 2, the following performance criteria have been used for the evaluation of schemes:

(i) Network lifetime: the network lifetime is generally measured as the time when the first node dies, or when the last node dies in the network [28–31, 42]. In this work, both definitions have been considered to demonstrate the supremacy of the MLBCT over DEBCRP, and ImDE-LEACH

(ii) Network stability: network stability refers to how smoothly the network operations are going on. It can be measured in terms of the rate of the network energy consumption and the average residual energy per network node. The lower the rate of energy consumption, the more stable the network is, resulting in improved network lifetime. Similarly, the higher the value of average residual energy per network node, the more stable and durable the network is

To further compare the performance of the schemes—MLBCT, DEBCRP, and ImDE-LEACH, packet delivery at the base station can also be considered as a criterion.

The success in this regard can be judged by the higher number of successfully delivered packets to the base station.

To find the energy consumption by the nodes in the network operation, the widely adopted first-order radio model [13, 28, 42, 46–52] has been used here in this work.

5.3. Results and Discussion. As stated in point 1 of Section 5.1, the suitability of the proposed fitness function equation (15) is manifested in the first set of experiments. Since the scheme is a metaheuristic one, a suitable fitness function might contribute a lot to decide the best possible clusters. The main objective of this work is to formulate the clusters which are balanced in the sense that the clusters are having an almost similar count of member nodes and the member nodes are located close to one another to have minimized intracluster communication.

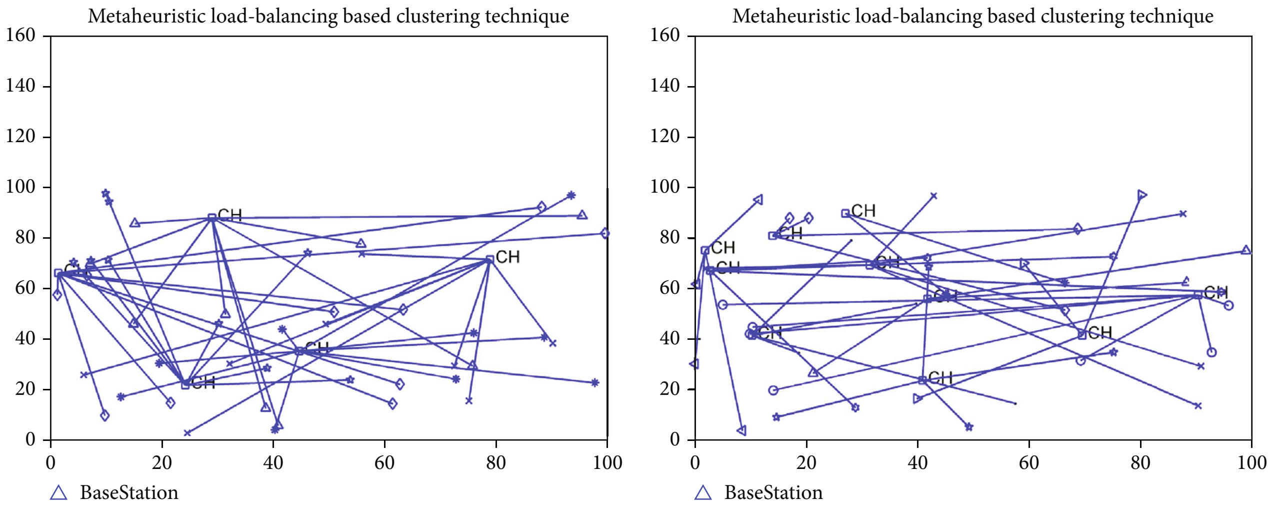
In this experimentation, variable node counts as in Table 2 have been considered for two instances of clustering such as 5 clusters and 10 clusters as shown in Figure 5.

The success of the fitness proposal mentioned above is evident in Figure 5. When implemented in the scheme DEBCRP, the proposed fitness function has been found more effective in having more balanced clusters. In other

```
Input:
* N: No. of randomly deployed sensor nodes.
*$ff^n()$: Fitness function.
* F: Mutation/Scaling factor.
* T: No. of iteration
* k: No. of user-specified clusters
*$C_r$: Crossover rate
◊   BEGIN
                        %% BOOTSTRAPPING PHASE %%.
◊ for i ⟵1: N
◊ Status Transmission($Node_i \longrightarrow$ BS)
◊ end for
◊ Random population generation (P) where each vector ($X_i$) refers to the complete assignment of all the nodes to the $k$ cluster heads
◊ for i ⟵1: size(P)
◊     $ff^n(X_i)$
◊ end for
◊ for i ⟵1: T
◊     for j ⟵1: size(P)
◊       $V_j = X_{best} + F(X_{r_1} - X_{r_2})$
                        %% $V_j$ is the $j^{th}$ donor vector
◊       $U_j = [u_j^l]$ where $u_j^l$ is the $l^{th}$ component of $U_j$ defined as follows:
                        $u_j^l = \begin{cases} v_j^l & \text{if } r \le C_r \text{ OR } l = \delta \\ x_j^l & \text{if } r > C_r \text{ AND } l \ne \delta \end{cases}$
                        %% $U_j$ is the $j^{th}$ trial vector
◊     end for      (inner loop)
◊     for j ⟵1: size(P)
◊       $ff^n(U_j)$
                        %% i.e. fitness function evaluation of the $j^{th}$ trial vector
◊       if $(ff^n(U_j) > ff^n(X_j))$
◊         Update P
                        %% Greedy approach for the update of population vector
◊       end if
◊     end for (inner loop)
◊ end for      (outer loop)
                        %% STEADY STATE PHASE %%
◊ while(nodes are alive)
◊     for i ⟵1:
                        %% i.e. for every cluster
◊       for j ⟵1:
                        %% m ⟶ no. of members in the $i^{th}$ cluster
◊         DataTransmission($Node_j^i \longrightarrow CH^i$)
◊       end for (inner loop)
◊       DataTransmission($CH^i \longrightarrow BS$)
                        %% here, aggregated data by the $CH^i$ to the base station
◊     end for (outer loop)
                        %% RESPONSIBLE NODE SELECTION PHASE %%
◊     for i ⟵1: k
◊         Random selection of $CH_{next}^i$ from within the $i^{th}$ cluster by $CH^{i.}$
◊         New CH's Information dissemination by the $CH^i$
◊ end for
◊ end while
◊ END
```

Algorithm 1: MLBCT.

words, clusters are obtained with an approximately similar count of member nodes, leading to the even distribution of load throughout the network nodes. In Figure 5(a), the efficacy of the proposed scheme is demonstrated with five clusters being formed in the network, whereas Figure 5(b) presents the same while partitioning the network into 10 clusters. It can be easily observed from the figure that the members recorded in the clusters do not vary to the extent



(a) Clustering instance with 5 clusters

(b) Clustering instance with 10 clusters

Figure 4: Simulation interface for network operation.

Table 2: Parameters used in the simulation.

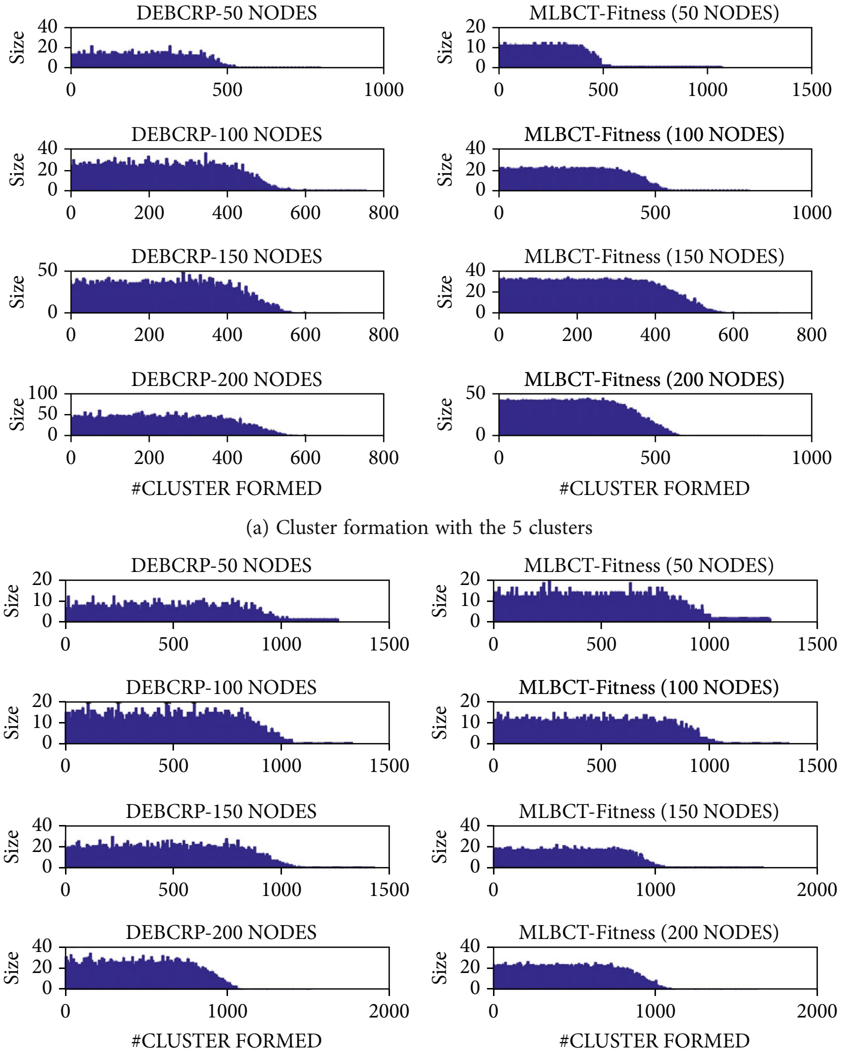
Parameter	Parameter's value
Network area	$100 \times 100\,\mathrm{m}^2$
Base station's position	$\{(50\,\mathrm{m}, 50\,\mathrm{m}), (50\,\mathrm{m}, 150\,\mathrm{m})\}$
Node deployment strategy	Random deployment
Number of nodes deployed in the network	{50,100,150,200}
Initial energy of the normal nodes	0.1 J
Size of data packet	4000 bits
Size of data packet header	200 bits
Energy consumed in data aggregation $(\varepsilon_{\mathrm{da}})$	5 nJ/bits/signal
Energy consumed in the transceivers' circuitry (E_{elec})	50 nJ/bit
Amplification factor in free space model $(\varepsilon_{\mathrm{fs}})$	$10\,\mathrm{pJ/bit/m}^2$
Amplification factor in multipath fading model $(\varepsilon_{\mathrm{mp}})$	$0.0013\,\mathrm{pJ/bit/m}^4$
Population size	10
Mutation factor	0.5
Crossover rate	0.7

as it is there in DEBCRP over the network rounds. Also, it has been verified that the scheme for the fitness evaluation of the clusters works invariably well irrespective of node density present in the network.

5.3.1. Statistical Analysis. Statistical analysis is performed to further explain the efficacy of the proposed fitness function (MLBCT-fitness) as in equation (15) in producing the balanced clusters. This is done by finding out the standard deviation of average cluster size, σ_{CS} following equation (9) along with the confidence interval. Standard deviation is defined as the measurement of how the clusters being produced deviate from the ideal distribution of the nodes among the specified number of clusters. The ideal distribution refers to the clusters with (N/k) nodes if N nodes are to be distributed among k clusters.

For this very purpose, as explained above, the proposed fitness function is fitted into the scheme of DEBCRP, and the performance of such a modified scheme is compared with that of DEBCRP with respect to the formation of clusters. This is achieved by recording the clusters' length in both cases until the first node dies. Afterward, standard deviations of the average cluster size are measured in both of the cases—with its own fitness function ($\sigma_{D\text{-Fitness}}$) and MLBCT-fitness function ($\sigma_{M\text{-Fitness}}$).

Figures 6(a) and 6(b) demonstrate the standard deviations of the average cluster size for the different network deployments with 50, 100, 150, and 200 nodes with the requirements mentioned above of having 5 clusters and 10 clusters, respectively. It can be explicitly observed that the standard deviations and the MLBCT-fitness function are quite low compared to the standard deviations obtained via



(a) Cluster formation with the 5 clusters

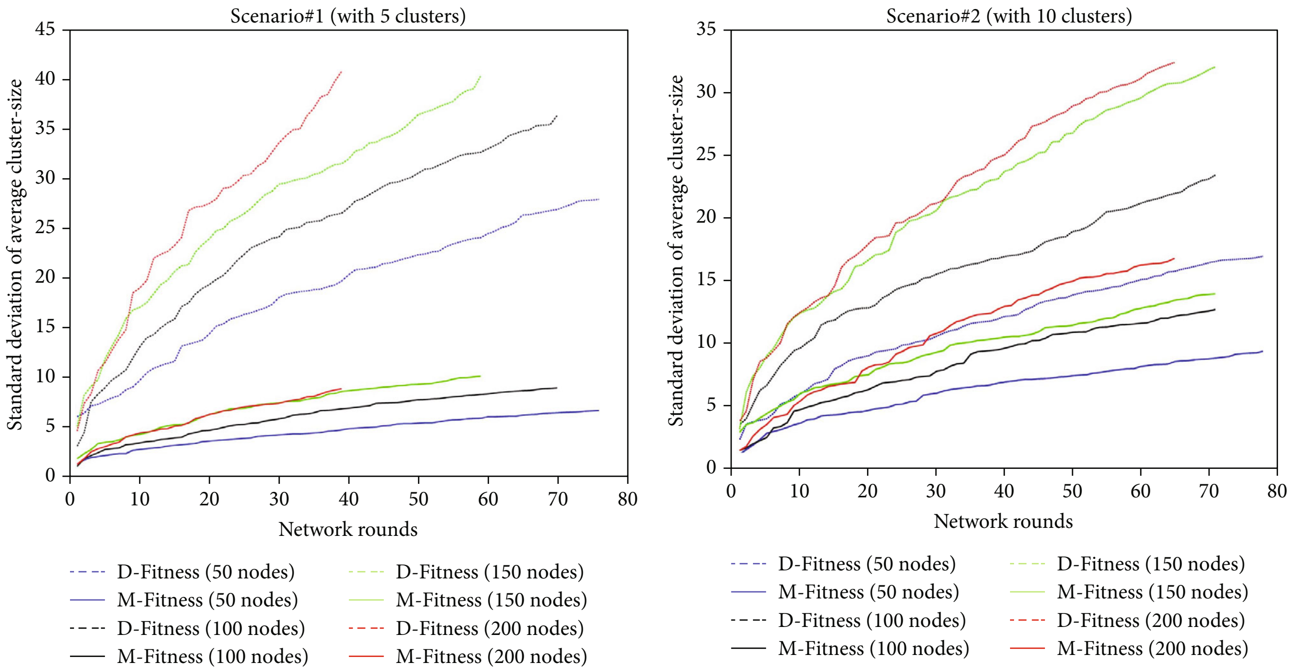
(b) Cluster formation with the 10 clusters

Figure 5: Efficacy of the proposed fitness function.

the application of the DEBCRP-fitness function for all the node deployments under both the specified requirements of 5 clusters and 10 clusters. This also justifies the efficacy of the scheme.

Another statistical analysis known as confidence interval justifies the probability of the deployment of the nodes within a range of the values of the cluster. In this case, the confidence intervals with the confidence levels 95% and 99%, respectively, are measured for both cases of the clustering scenarios with variable node counts. Table 3 clearly explains the efficacy of the MLBCT-fitness function over the fitness function used in DEBCRP in every possible network configuration. For example, when 100 nodes are deployed to be distributed among 5 clusters, ideally, each cluster should have 20 nodes. Here, the proposed fitness function ensures that each cluster has a node count in the range [18.8245, 21.1755] with 95% confidence and in the range [18.4526, 21.5474] with 99% confidence, whereas the fitness function of DEBCRP finds the same as in the ranges [15.2210, 24.7790] and [13.7093, 26.2907] with 95% and 99% confidences, respectively. It can be easily intuited that the node count in each cluster is much closer to the ideal node count (20 here) with the MLBCT-fitness function when compared to that with the DEBCRP-fitness function. The consistency of the MLBCT-fitness function in terms of balanced clusters' formation can be seen in Table 3.

5.3.2. Experimental Analysis. In this second set of experiments, as stated in point 2 of Section 5.1, MLBCT is compared to DEBCRP, ImDE-LEACH, and LEACH concerning the metrics—network lifetime, network energy consumption rate, and average residual energy per network node under two different network configurations, say WSN#1 and WSN#2. In WSN#1, the sink has been placed at the center of the $100 \times 100\,\text{m}^2$ sensing field, precisely at (50 m, 50 m) whereas, in WSN#2, the sink is located outside the sensing field at (50 m, 150 m). Moreover, to validate the adaptability of the scheme, simulations have been conducted with variable node deployments, say with 50 nodes, 100 nodes, 150 nodes, and 200 nodes.



(a) Scenario#1-5 clusters

(b) Scenario#2-10 clusters

Figure 6: Standard deviation of average cluster size for the clusters formed over the network rounds.

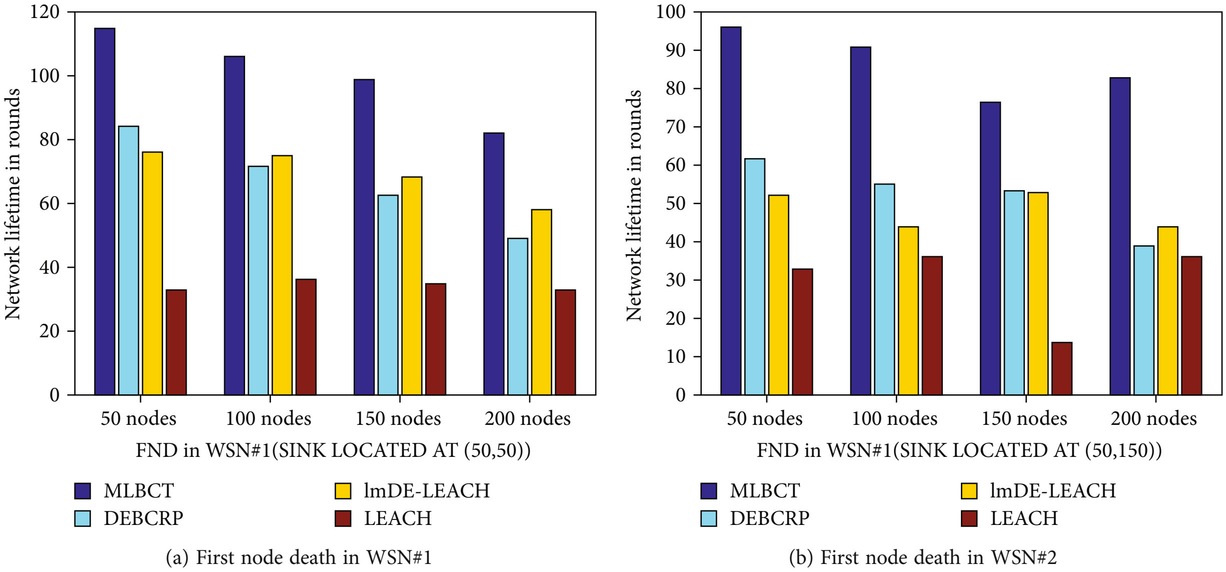
Table 3: Mean of standard deviations and confidence intervals for the clusters generated.

Clustering scenario	#nodes	$\text{Mean}_{\sigma D\text{−Fitness}}$	$\text{Mean}_{\sigma M\text{−Fitness}}$	Interval estimate with 95% confidence		Interval estimate with 99% confidence	
				With *D*-fitness	With *M*-fitness	With *D*-fitness	With *M*-fitness
Scenario#1 (with 5 clusters)	50	18.9254	4.4628	[4.7541, 15.2459]	[8.7630, 11.2370]	[3.0947, 16.9053]	[8.3717, 11.6283]
	100	24.3826	5.9975	[15.2210, 24.7790]	[18.8245, 21.1755]	[13.7093, 26.2907]	[18.4526, 21.5474]
	150	27.1672	6.9578	[25.6523,34.3477]	[28.8865, 31.1135]	[24.2771, 35.7229]	[28.5343, 31.4657]
	200	25.9534	5.7280	[36.4030, 43.5970]	[39.2061, 40.79939]	[35.2652, 44.7348]	[38.9550, 41.0450]
Scenario#2 (with 10 clusters)	50	11.6163	6.3429	[1.7801, 8.2199]	[3.2419, 6.7581]	[0.7616, 9.2384]	[2.6857, 7.3143]
	100	15.8541	8.4076	[6.8926, 13.1074]	[8.3521, 11.6479]	[5.9096, 14.0904]	[7.8308, 12.1692]
	150	21.6239	9.5944	[11.5395, 18.4605]	[13.4646, 16.5354]	[10.4448, 19.5552]	[12.9789, 17.0211]
	200	21.7987	10.7536	[16.9789, 23.0211]	[18.5096, 21.4904]	[16.0232, 23.9768]	[18.0382, 21.9618]

(1) Network Lifetime. As mentioned earlier in this section that the network lifetime can be defined as the time when the first node dies in the network or the time when the last node dies in the network. In Figures 7 and 8, both strategies have been followed separately.

Figures 7(a) and 7(b) describe the death of the first node that is FND (first node death) in the schemes MLBCT, DEBCRP, ImDE-LEACH, and LEACH under the network scenarios WSN#1 and WSN#2.

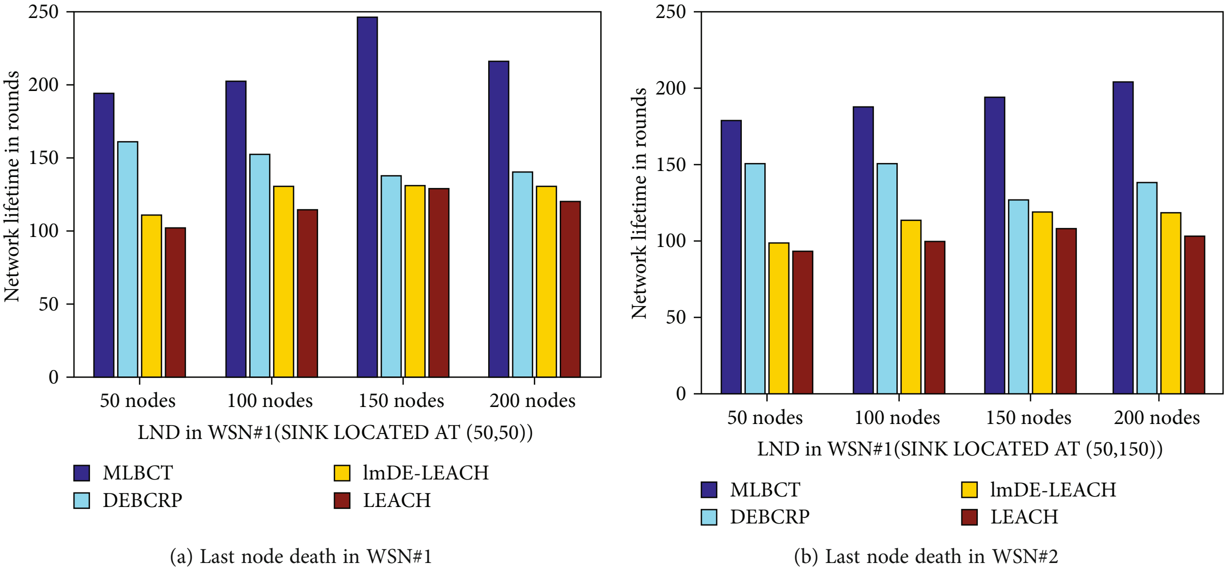
In WSN#1 (Figure 7(a)), when the number of nodes deployed are 50, 100, 150, and 200, the events of the first node's death (FND) occur at the round no. 115, 106, 99, and 82 in the proposed scheme; at 84, 72, 63, and 49 in DEBCRP; at 76, 75, 68, and 58 in ImDE-LEACH; and 33, 36, 35, and 33 in LEACH, respectively. Similarly, in



(a) First node death in WSN#1

(b) First node death in WSN#2

Figure 7: Network lifetime comparison in terms of FND.



(a) Last node death in WSN#1

(b) Last node death in WSN#2

Figure 8: Network lifetime comparison in terms of LND.

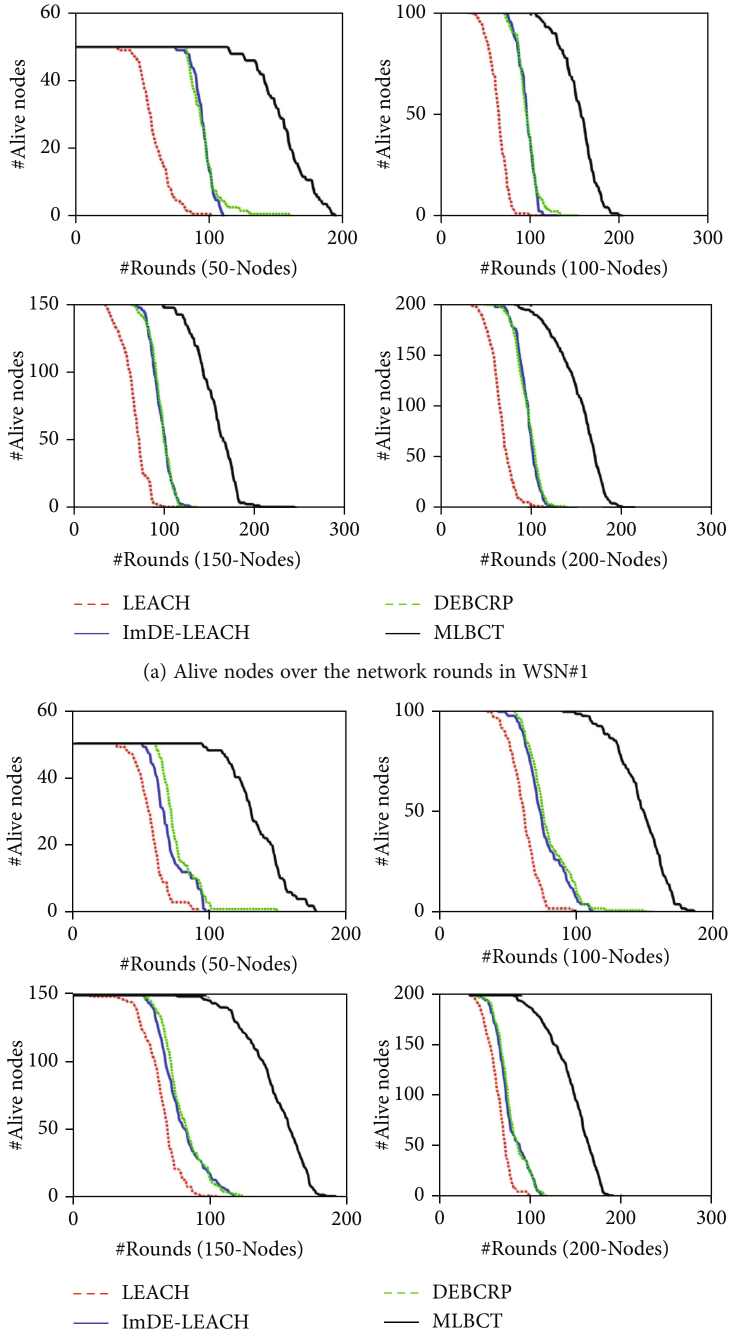
WSN#2 (Figure 7(b)), FNDs occur at round no. 96, 91, 77, and 83 in MLBCT; at 62, 55, 53, and 59 in DEBCRP; at 52, 44, 53, and 44 in ImDE-LEACH; and at 33, 36, 14, and 36 in LEACH, respectively, for the aforementioned nodes' count.

On the other hand, if the network lifetime is taken as the time when the last node dies that is LND (last node death) in the network, Figures 7(a) and 7(b) describe the outcomes of experiments conducted in this regard with the variable number of nodes as above, say 50, 100, 150, and 200, respectively.

In WSN#1 (Figure 8(a)), the last node's death events occur at round no. 194, 202, 246, and 216 in the MLBCT; at 161, 152, 138, and 141 in DEBCRP; at 111, 131, 131, and 130 in ImDE-LEACH; and at 102, 114, 129, and 119 in LEACH, respectively. Likewise, in WSN#2 (Figure 8(b)), LNDs occur at round no. 178, 187, 193, and 203 in MLBCT; at 150, 151, 126, and 138 in DEBCRP; at 98, 113, 119, and 118; and at 93, 99, 108, and 103 in LEACH, respectively, for the aforementioned nodes' count. The appreciable results due to FND and LND calculation state the supremacy of using the proposed MLBCT over other schemes.

Moreover, the comparative performance of the schemes MLBCT, DEBCRP, ImDE-LEACH, and LEACH with respect to the nodes' death rate can also be observed from Figure 9.

Figure 9(a) describes the performance of the MLBCT against that of DEBCRP, ImDE-LEACH, and LEACH in



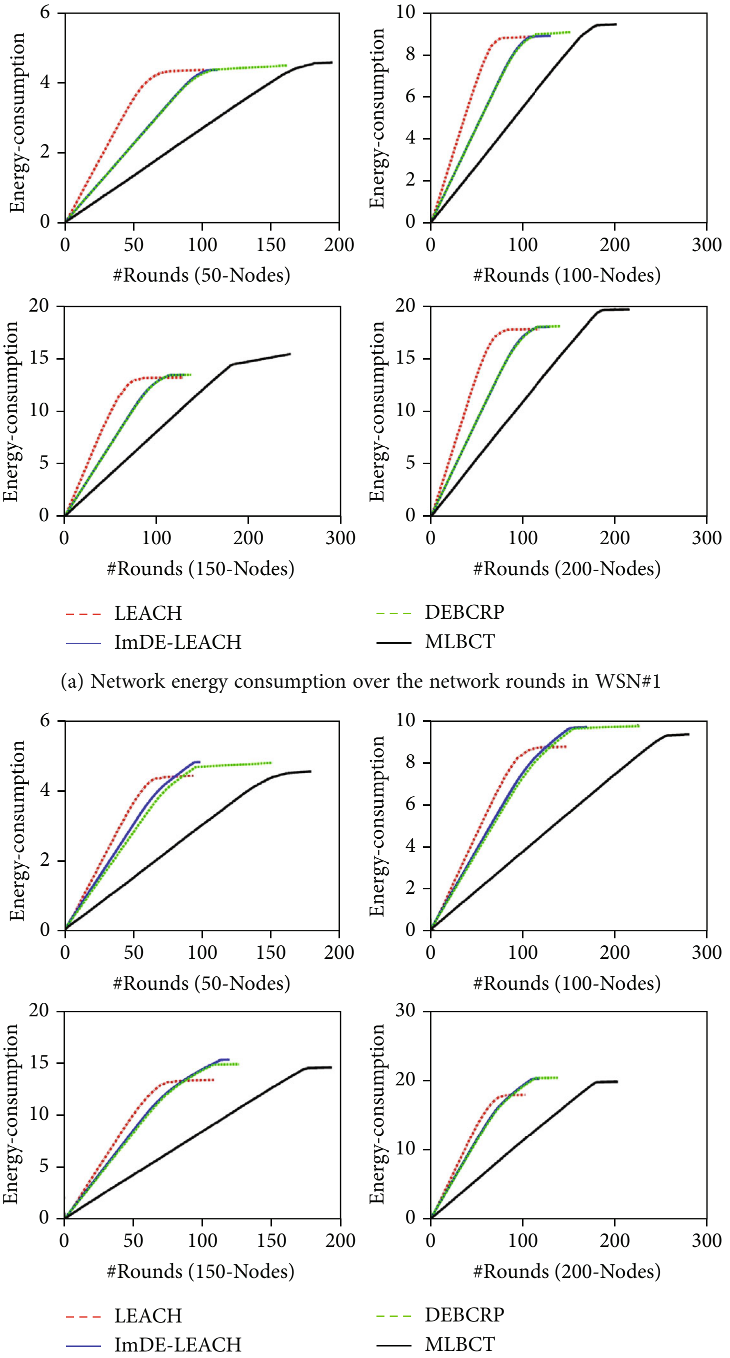
(a) Alive nodes over the network rounds in WSN#1

(b) Alive nodes over the network rounds in WSN#2

Figure 9: Network lifetime comparison in terms of alive nodes/round.

variable node population under the first network scenario WSN#1. Similarly, Figure 9(b) describes the same but for WSN#2. It is evident from Figure 9 that irrespective of the network configuration and nodes' population in the sensing field, MLBCT performs consistently well as the nodes' death rate is low in MLBCT, and hence, the number of alive nodes is high at any point of network operation in MLBCT when compared to DEBCRP, ImDE-LEACH, and LEACH. Thus, it can be concluded here that the MLBCT outperforms DEBCRP, ImDE-LEACH, and LEACH in terms of the first performance criterion—network lifetime.

(2) Network Energy Consumption. From Figure 10, it can be concluded that at any point of the network operation, the energy consumption in MLBCT is less than that in DEBCRP, ImDE-LEACH, and LEACH in both of the scenarios implemented that is in WSN#1 (Figure 10(a)) and



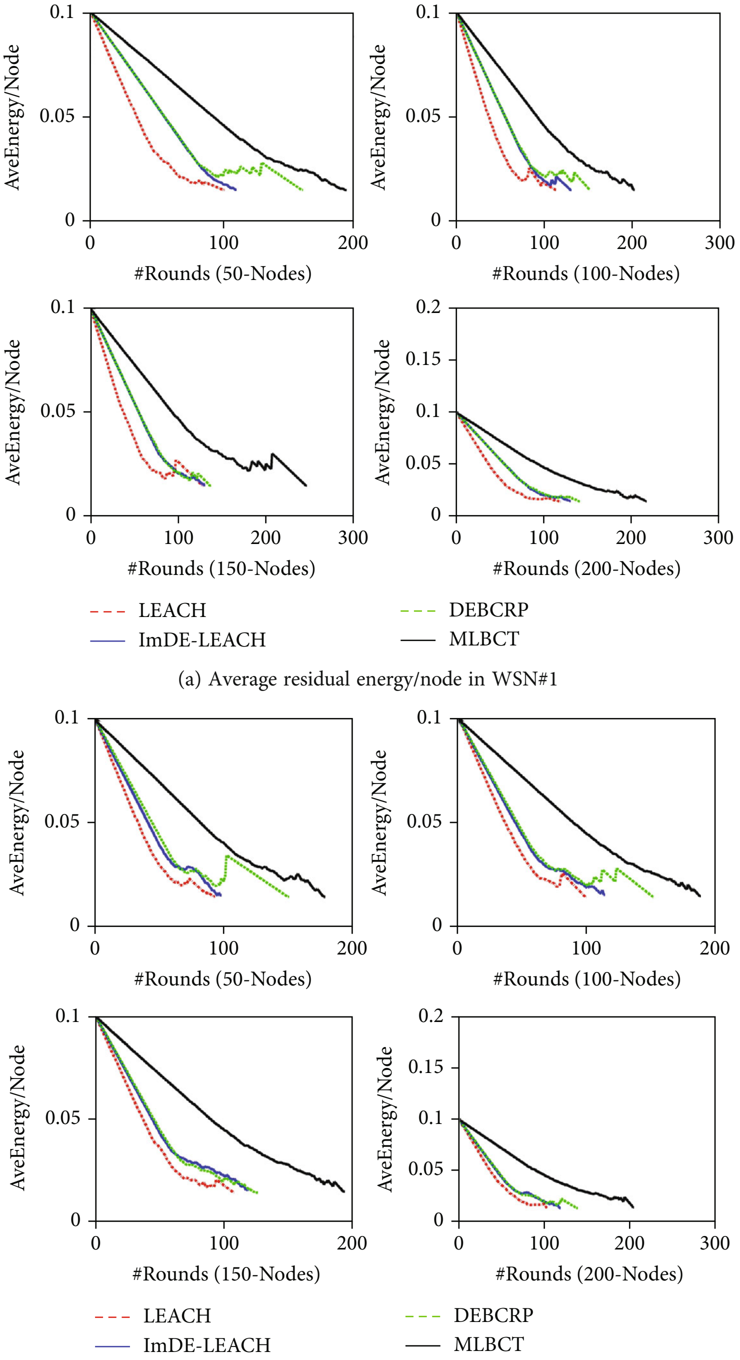
(a) Network energy consumption over the network rounds in WSN#1

(b) Network energy consumption over the network rounds in WSN#2

Figure 10: Comparison of network energy consumption over the network rounds.

WSN#2 (Figure 10(b)). Moreover, to demonstrate the consistency in the performance, variable counts of sensor nodes have been deployed here too.

(3) Average Residual Energy/Node. In this next set of experiments, the performance of MLBCT is measured in terms of the average residual energy that a network node has at any point in the network operation for the schemes DEBCRP, ImDE-LEACH, and LEACH. It can be explicitly observed that the nodes are always equipped with a larger amount of residual energy if being operated with MLBCT in comparison to DEBCRP, ImDE-LEACH, and LEACH (Figure 11). It is noticed not only in WSN#1 (Figure 11(a)) but also in WSN#2 (Figure 11(b)); average residual energy for a network node is higher at any point in network operation if implemented with MLBCT.



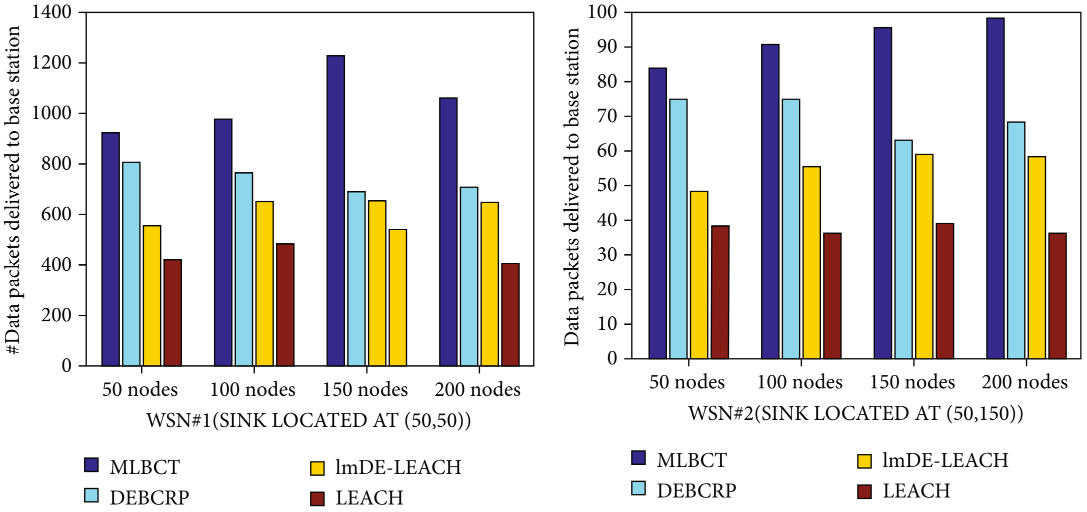
(a) Average residual energy/node in WSN#1

(b) Average residual energy/node in WSN#2

Figure 11: Comparison of average energy/node over the network rounds.

This depicts that a network utilizing MLBCT saves energy and keeps its resource intact for future usage, which is the desired criteria for sensor networks.

(4) Data Packet Delivery at Base Station. In the final set of experiments, the performance of MLBCT against the DEBCRP, ImDE-LEACH, and LEACH with respect to the number of data packets delivered to the base station is compared. The predominance of the proposed scheme, MLBCT, can be read for both the network scenarios WSN#1 and WSN#2 in Figures 12(a) and 12(b), respectively. For the 50, 100, 150, and 200 nodes, MLBCT enriches the base station with 915, 969, 1221, and 1054 data packets, respectively. However, DEBCRP results into 800, 755, 685, and 700 data packets, ImDE-LEACH results into 550, 650, 650, and 645 data packets, and LEACH results into 416, 477, 533, and



(a) Data packet delivery at base station in WSN#1

(b) First node death in WSN#2

Figure 12: Data packet delivery at base station.

401 data packets, respectively, for the aforesaid network nodes into WSN#1. Similarly, for the WSN#2, in comparison to 745, 750, 625, and 685 data packets due to DEBCRP, 98, 113, 119, and 118 data packets through ImDE-LEACH, and 382, 360, 388, and 372 data packets via LEACH, MLBCT results into 838, 903, 950, and 983 data packets at the base station, respectively, for the node deployment mentioned above. This suggests that MLBCT successfully transmits more packets depicting its dominance in terms of successful transmission.

Based on the outcomes of the various simulations conducted so far, it can be concluded that the MLBCT outperforms the DEBCRP, ImDE-LEACH, and LEACH in terms of the chosen criteria of network lifetime, network stability, average residual energy, and data packet delivery.

6. Conclusion and Future Works

In this work, a Metaheuristic Load-Balancing-Based Clustering Technique has been proposed for wireless sensor networks. To achieve the prime objective of load-balanced clusters, a fitness function has been proposed that offers balanced clusters in terms of their size and energy and ensures the members to be in close proximity to one another reducing the cost of intracluster communication. Through an extensive set of simulations and experimentation, the supremacy of the proposed scheme MLBCT has been proved over the existing ones DEBCRP, and ImDE-LEACH in terms of improved network lifetime and network stability, average residual energy, and data packet delivery.

Statistical analysis also justifies and supports the feasibility of the scheme. Moreover, the scheme's adaptability and scalability have also been established by varying the network configuration with the different number of nodes and different placement of the base station.

As a future extension of this work, a heterogeneous wireless sensor network (HWSN) would be investigated to device a clustering-based scheme induced by metaheuristic techniques to consistently contribute to the network operations without being affected by the heterogeneity present in the network.

Data Availability

Extensive analysis, method, and result data has been fully provided.

Conflicts of Interest

The authors declare that they have no competing interests.

Acknowledgments

This work is partially supported by DST/TDT/DDP-38/2021, Device Development Programme (DDP), by the Department of Science & Technology (DST), Ministry of Science and Technology, Government of India.

References

[1] I. F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communication Magazine*, vol. 40, no. 8, pp. 102–114, 2002.

[2] C. O. Iwendi and A. R. Allen, "Enhanced security technique for wireless sensor network nodes," in *IET Conference on Wireless Sensor Systems (WSS 2012)*, pp. 1–5, London, 2012.

[3] A. Dumka, S. K. Chaurasiya, A. Biswas, and H. L. Mandoria, *A Complete Guide to Wireless Sensor Networks: From Inception to Current Trends*, CRC Press, Boca Raton, Florida, USA, 1st edition, 2019.

[4] S. J. Ramson and D. J. Moni, "Applications of wireless sensor networks –survey," in *2017 International Conference on Innovations in Electrical, Electronics, Instrumentation and Media Technology (ICEEIMT)*, pp. 325–329, Coimbatore, India, 2017.

[5] S. Ghiasi, A. Srivastava, X. Yang, and M. Sarrafzadeh, “Optimal energy aware clustering in sensor networks,” *Sensors Journal*, vol. 2, no. 7, pp. 258–269, 2002.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balkrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *Proceedings of 33rd Hawaii international conference on system science*, pp. 1–10, Maui, HI, USA, 2000.

[7] W. R. Heinzelman, A. Chandrakasan, and H. Balkrishnan, “An application -specific protocol architecture for wireless microsensor networks,” *IEEE Transactions on wireless communications*, vol. 1, no. 4, pp. 660–670, 2002.

[8] S. Lindsey and C. S. Raghavendra, “PEGASIS: power-efficient gathering in sensor information systems,” in *Proceedings, IEEE Aerospace Conference*, pp. 1125–1130, Big Sky, MT, USA, 2002.

[9] S. K. Chaurasiya, T. Pal, and S. D. Bit, “An enhanced energy-efficient protocol with static clustering for WSN,” in *The International Conference on Information Networking 2011 (ICOIN2011)*, pp. 58–63, Kuala Lumpur, Malaysia, 2011.

[10] S. K. Chaurasiya, J. Sen, S. Chatterjee, and S. D. Bit, “EBLEC: an energy-balanced lifetime enhancing clustering for WSN,” in *2012 14th International Conference on Advanced Communication Technology (ICACT)*, pp. 189–194, PyeongChang, Korea (South), 2012.

[11] M. N. Cheraghlou and M. Haghparast, “A novel fault-tolerant leach clustering protocol for wireless sensor networks,” *Journal of Circuits, Systems and Computers*, vol. 23, no. 3, article 1450041, 2014.

[12] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. Jafarali Jassbi, “Novel fault-tolerant clustering-based multipath algorithm (FTCM) for wireless sensor networks,” *Telecommunication Systems*, vol. 74, no. 4, pp. 411–424, 2020.

[13] O. Younis and S. Fahmy, “HEED: a hybrid energy-efficient distributed clustering approach for ad hoc sensor networks,” *IEEE Transaction on Mobile Computing*, vol. 3, no. 4, pp. 366–379, 2004.

[14] E. Moridi, M. Haghparast, M. Hosseinzadeh, and S. Jafarali Jassbi, “A novel hierarchical fault management framework for wireless sensor networks: HFMF,” *Peer-to-Peer Networking and Applications*, vol. 85, 2021.

[15] T. R. Gadekallu, M. Alazab, R. Kaluri et al., “Hand gesture classification using a novel CNN-crow search algorithm,” *Complex & Intelligent Systems*, vol. 7, no. 4, pp. 1855–1868, 2021.

[16] T. Shankar, S. Shanmugavel, A. Karthikeyan, A. M. Gupte, and S. Sarkar, “Load balancing and optimization of network lifetime by use of double cluster head clustering algorithm and its comparison with various extended LEACH versions,” *International Review on Computers and Software*, vol. 8, no. 3, pp. 795–803, 2013.

[17] T. Shankar and S. Shanmugavel, “Energy optimization in cluster based wireless sensor networks,” *Journal of Engineering Science and Technology*, vol. 9, no. 2, pp. 246–260, 2014.

[18] A. M. Zungeru, L. M. Ang, and K. P. Seng, “Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison,” *Journal of Network and Computer Applications*, vol. 35, no. 5, pp. 1508–1536, 2012.

[19] M. Saleem, A. Gianni, and M. F. Di Caro, “Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions,” *Information Sciences*, vol. 181, no. 20, pp. 4597–4624, 2011.

[20] R. V. Kulkarni and G. K. Venayagamoorthy, “Particle swarm optimization in wireless-sensor networks: a brief survey,” *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 2, pp. 262–267, 2011.

[21] R. Mukherjee, S. Debchoudhury, R. Kundu, S. Das, and P. Suganthan, “Adaptive differential evolution with locality based crossover for dynamic optimization,” in *2013 IEEE Congress on Evolutionary Computation*, pp. 63–70, Cancun, Mexico, 2013.

[22] S. Das, A. Konar, and U. K. Chakraborty, “Annealed differential evolution,” in *2007 IEEE Congress on Evolutionary Computation*, pp. 1926–1933, Singapore, 2007.

[23] P. Kuila, S. K. Gupta, and P. K. Jana, “A novel evolutionary approach for load balanced clustering problem for wireless sensor networks,” *Evolutionary Computation*, vol. 12, pp. 48–56, 2013.

[24] P. Kuila and P. K. Jana, “Energy efficient clustering and routing algorithms for wireless sensor networks: particle swarm optimization approach,” *Engineering Applications of Artificial Intelligence*, vol. 33, pp. 127–140, 2014.

[25] D. Kim, S. Song, and B. Y. Choi, “Energy-efficient adaptive geosource multicast routing for wireless sensor networks,” *Journal of Sensors*, vol. 2013, 14 pages, 2013.

[26] J. Vesterstrom and R. Thomsen, “A comparative study of differential evolution, particle swarm optimization, and evolutionary algorithms on numerical benchmark problems,” in *Proceedings of the 2004 Congress on Evolutionary Computation (IEEE Cat. No.04TH8753)*, pp. 1980–1987, Portland, OR, USA, 2004.

[27] X. Li, L. Xu, H. Wang, J. Song, and S. X. Yang, “A differential evolution-based routing algorithm for environmental monitoring wireless sensor networks,” *Sensors*, vol. 10, no. 6, pp. 5425–5442, 2010.

[28] P. Kuila and P. K. Jana, “A novel differential evolution based clustering algorithm for wireless sensor networks,” *Applied Soft Computing*, vol. 25, pp. 414–425, 2014.

[29] C. P. Low, C. Fang, J. M. Ng, and Y. H. Ang, “Efficient load-balanced clustering algorithms for wireless sensor networks,” *Computer Communications*, vol. 31, no. 4, pp. 750–759, 2008.

[30] G. Gupta and M. Younis, “Load-balanced clustering of wireless sensor networks,” in *IEEE International Conference on Communications, 2003. ICC '03*, vol. 3, pp. 1848–1852, Anchorage, AK, USA, 2003.

[31] K. Pratyay and P. K. Jana, “Energy efficient load-balanced clustering algorithm for wireless sensor networks,” *Procedia Technology*, vol. 6, pp. 771–777, 2012.

[32] T. Sweta Potthuri and A. R. Shankar, “Lifetime improvement in wireless sensor networks using hybrid differential evolution and simulated annealing (DESA),” *Ain Shams Engineering Journal*, vol. 9, no. 4, pp. 655–663, 2018.

[33] J. Brest, S. Greiner, B. Boskovic, M. Mernik, and V. Zumer, “Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems,” *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

[34] S. Randhawa and S. Jain, “MLBC: multi-objective load balancing clustering technique in wireless sensor networks,” *Applied Soft Computing*, vol. 74, pp. 66–89, 2019.

[35] M. Ghahramani and A. Laakdashti, “Efficient energy consumption in wireless sensor networks using an improved differential evolution algorithm,” in *2020 10th International Conference on Computer and Knowledge Engineering (ICCKE)*, pp. 18–23, Mashhad, Iran, 2020.

[36] G. P. Gupta and B. Saha, “Load balanced clustering scheme using hybrid metaheuristic technique for mobile sink based wireless sensor networks,” *Journal of Ambient Intelligence and Humanized Computing*, vol. 2020, 2020.

[37] C. Iwendi, P. K. R. Maddikunta, T. R. Gadekallu, K. Lakshmanna, A. K. Bashir, and M. J. Piran, “A metaheuristic optimization approach for energy efficiency in the IoT networks,” *Software: Practice and Experience*, vol. 51, pp. 1–14, 2021.

[38] S. Ponnan, A. K. Saravanan, C. Iwendi, E. Ibeke, and G. Srivastava, “An artificial intelligence-based quorum system for the improvement of the lifespan of sensor networks,” *IEEE Sensors Journal*, vol. 21, no. 15, pp. 17373–17385, 2021.

[39] S. H. Sackey, J. Chen, A. J. Henry, and X. Zhang, “A clustering approach based on genetic algorithm for wireless sensor network localization,” in *2019 15th International Conference on Computational Intelligence and Security (CIS)*, pp. 54–58, Macao, China, 2019.

[40] J. Chen, S. H. Sackey, J. H. Anajemba, X. Zhang, and Y. He, “Energy-efficient clustering and localization technique using genetic algorithm in wireless sensor networks,” *Complexity*, vol. 2021, 12 pages, 2021.

[41] S. H. Sackey, J. A. Ansere, J. H. Anajemba, M. Kamal, and C. Iwendi, “Energy efficient clustering based routing technique in WSN using brain storm optimization,” in *2019 15th International Conference on Emerging Technologies (ICET)*, Peshawar, Pakistan, 2019.

[42] M. Kaddi, Z. Khalili, and M. Bouchra, “A differential evolution based clustering and routing protocol for WSN,” in *2020 International conference on mathematics and information technology*, pp. 190–195, Adrar, Ageria, 2020.

[43] A. Gaur and T. Kumar, “Switching-differential evolution (S-DE) for cluster head election in wireless sensor network,” *IJARIIE*, vol. 2, no. 5, pp. 2395–4396, 2016.

[44] R. V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems,” *Computer-Aided Design*, vol. 43, no. 3, pp. 303–315, 2011.

[45] R. V. Rao, V. J. Savsani, and D. P. Vakharia, “Teaching-learning-based optimization: an optimization method for continuous non-linear large scale problems,” *Information Sciences*, vol. 183, no. 1, pp. 1–15, 2012.

[46] M. Ramadas and A. Abraham, “Clustering wireless sensor networks using ImDE algorithm with LEACH protocol,” in *2019 2nd IEEE Middle East and North Africa COMMunications Conference (MENACOMM)*, pp. 1–4, Manama, Bahrain, 2019.

[47] S. K. Chaurasiya, J. Mondal, and S. Datta, “Field-of-view based hierarchical clustering to prolong network lifetime of WMSN with obstacles,” in *2014 International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, pp. 72–77, Hosur, India, 2014.

[48] S. K. Singh, P. Kumar, and J. P. Singh, “A survey on successors of LEACH protocol,” *IEEE Access*, vol. 5, pp. 4298–4328, 2017.

[49] A. Mehmood, S. Khan, B. Shams, and J. Lloret, “Energy-efficient multi-level and distance-aware clustering mechanism for WSNs,” *International Journal of Communication Systems*, vol. 28, no. 5, pp. 972–989, 2015.

[50] R. Banerjee, S. Chatterjee, and S. Das Bit, “An energy saving audio compression scheme for wireless multimedia sensor networks using spatio-temporal partial discrete wavelet transform,” *Computers and Electrical Engineering*, vol. 48, pp. 389–404, 2015.

[51] R. Banerjee and S. Das Bit, “Low-overhead video compression combining partial discrete cosine transform and compressed sensing in WMSNs,” *Wireless Networks*, vol. 25, no. 8, pp. 5113–5135, 2019.

[52] R. Banerjee and S. Das Bit, “An energy efficient image compression scheme for wireless multimedia sensor network using curve fitting technique,” *Wireless Networks*, vol. 25, no. 1, pp. 167–183, 2019.