I. J. Computer Network and Information Security, **2025, 5, 91-113**
Published Online on October 8, 2025 by MECS Press (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2025.05.07

Enhanced Wireless Sensor Network Lifetime using Modified SFLA with Improved Fitness Function

Abdulhameed Pathan
Technical Consultant, Adobe India Private Limited, Bangalore, India
E-mail: abdulpathan1994@gmail.com
ORCID iD: https://orcid.org/0000-0002-2741-7442

Amol C. Adamuthe*
Department of Information Technology, Kasegaon Education Society's Rajarambapu Institute of Technology, affiliated to Shivaji University, Sakharale, MS – 415414, India
E-mail: amol.admuthe@gmail.com
ORCID iD: https://orcid.org/0000-0003-2483-7051
*Corresponding author

Received: 15 May 2025; Revised: 09 July 2025; Accepted: 21 August 2025; Published: 08 October 2025

Abstract: In the pursuit of enhancing Wireless Sensor Networks (WSNs), this study introduces a novel amalgamation of the Enhanced Shuffled Frog Leaping Algorithm (ESFLA) with a multi-solution evolution paradigm. By intricately examining diverse algorithmic facets, including partitioning strategies, fitness functions, and convergence mechanisms, the research endeavors to elevate the efficiency, robustness, and longevity of WSNs. Rigorous experimentation across 15 input datasets, meticulously categorized based on network density, unveils profound insights into the algorithm's performance. Significantly, the proposed ESFLA-MSU achieves exceptional outcomes, eclipsing traditional methods. A pioneering fitness function optimally redistributes workloads, culminating in extended network lifespans, a striking reduction in energy consumption by up to 28.5%, and remarkable load balancing improvements of up to 35.7%. Comparative analyses of partitioning strategies underscore ESFLA's adaptability, while multi-solution evolution integration accelerates convergence, with an expedited rate of up to 46.3%.

Index Terms: Shuffled Frog Leaping Algorithm, Load Balancing of Gateways, Wireless Sensor Networks.

1. Introduction

Wireless Sensor Networks (WSNs) are networks of small autonomous devices called sensor nodes that collect data from the environment and communicate wirelessly. They find applications in diverse domains such as environmental monitoring, industrial automation, healthcare, smart cities, agriculture, disaster management, military and defense, wildlife tracking, home automation and structural health monitoring. Wireless Sensor Networks (WSNs) encounter challenges due to limited energy, scalability, data aggregation, security, node failures and more. Ensuring energy efficiency while performing tasks, managing large networks, maintaining data quality, and addressing security threats are major concerns. Additionally, adapting to dynamic topologies, reducing communication overhead, and managing resource constraints are significant challenges. Achieving accurate localization, maintaining quality of service, and distributing tasks effectively among heterogeneous nodes further complicate WSNs. Solving these challenges requires innovative approaches in areas such as energy management, communication protocols, security mechanisms, and optimization techniques.

Figure 1 illustrates the communication process in cluster-based Wireless Sensor Networks (WSNs). Within these WSNs, a gateway assumes the role of gathering, processing, and transmitting data received from sensor nodes located within the cluster area. Each of these operations necessitates the utilization of a certain amount of energy. The gateways are equipped with batteries, which have limited energy reserves. In figure 2, three distinct gateways labeled as “g1”, “g2” and “g3” come into view. These gateways are each associated with a cluster that comprises six, four and two sensor nodes, respectively. What becomes evident is that gateway “g1” bears a significantly greater workload when contrasted with the other gateways. This situation has the potential to result in the premature depletion of the battery in gateway "g1", posing a challenge to its sustained functionality.

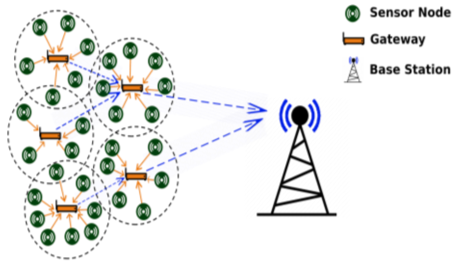


Fig.1. Communication of WSNs (Taken from [1])

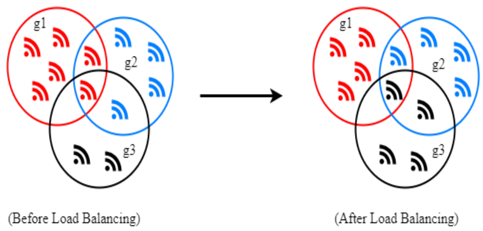


Fig.2. Example of gateways load balancing

WSNs enable data collection from remote and inaccessible areas, contributing to scientific research, efficient resource management, improved decision-making, and enhanced safety. They have become integral for efficient data transfer in various domains. The progress in technology has led to the development of compact sensor nodes that seamlessly integrate into WSNs, performing vital functions like data sensing and transmission. At the core of WSNs, sensor nodes play a pivotal role in sensing data and transmitting it to base stations or fellow sensor nodes. However, this essential operation demands energy, a constrained resource in WSNs [2,3]. Sensor nodes exclusively rely on battery power, and given their non-replaceable or non-rechargeable nature [4], the pursuit of energy-efficient data transmission strategies becomes imperative. The improper distribution of the workload onto a gateway can result in premature battery depletion, leading to an early failure of the gateway. This, in turn, may escalate the energy burden on nearby gateways, potentially initiating a chain reaction of gateway failures. Hence, it becomes imperative to implement effective load-balancing strategies for gateways to mitigate such issues [5]. Addressing this issue necessitates the implementation of a load-balancing strategy among the gateways, as exemplified in figure 2. This practice ensures an equitable distribution of tasks among the gateways, thereby averting the potential early exhaustion of g1's energy reserves, thereby promoting sustained network functionality.

The Shuffled Frog Leaping Algorithm (SFLA) is a nature-inspired optimization technique used to solve complex optimization problems. It mimics the behavior of frogs and works through iterations to find the best solutions [6]. Initially, a population of "frogs" is created, each representing a potential solution. These frogs are evaluated based on their fitness. Through shuffling, local search, and memetic operations, the algorithm collaborates and competes to improve solutions over iterations. It continues until a stopping criterion is met, delivering the best-found solution. SFLA is valuable for a wide range of optimization tasks across various domains. SFLA amalgamates the advantageous characteristics found in memetic algorithms and particle swarm optimization [7]. It's noteworthy that in SFLA, memes propagate more swiftly than the genes in a genetic algorithm (GA), which results in an enhanced convergence rate. Furthermore, SFLA's inherent self-evolutionary mechanism renders it particularly well-suited for addressing combinatorial problems when compared to the particle swarm optimization (PSO) algorithm [6]. As a testament to its versatility and effectiveness, SFLA has been extensively examined in the context of resolving a myriad of intricate optimization challenges. These encompass diverse problem domains, including but not limited to pick-and-place sequencing optimization [7], the 0/1 knapsack problem [8], the traveling salesperson problem [9], project scheduling quandaries [10], as well as applications in flow shop scheduling [11,12], cloud computing optimization [13], social network optimization [14], vehicle routing problems [15], and economic load dispatch issues [16].

The evolution of the Shuffled Frog Leaping Algorithm (SFLA) within the literature has witnessed the incorporation of several enhancements aimed at augmenting its efficiency and effectiveness. For instance, Sharma et al. [17] introduced the Centroid Mutation Embedded Shuffled Frog Leaping Algorithm, a modification that integrates a geometric centroid mutation into the algorithm. This addition was designed to expedite the convergence process and improve its overall performance. Another advancement, namely the Binomial Crossover Embedded SFLA [18], was developed with the goal of not only accelerating convergence but also enhancing the algorithm's ability to exploit optimal solutions. In the pursuit of solutions that encompass a broader spectrum of possibilities, an Elitism-based SFLA

[19] was introduced. This particular variant focused on fostering diversification within the algorithm's solution space. By incorporating elitism, the algorithm ensures that the most promising solutions are retained across iterations, thereby maintaining a balance between exploration and exploitation. Further innovations have led to hybridized iterations of the SFLA. Wang et al. [20] presented the Shuffled Frog Leaping Differential Evolution (SFDE), a novel hybrid approach that combines the SFLA with Differential Evolution techniques. This integration aimed to enhance radio time perception and elevate system throughput, showcasing the versatility of the algorithm and its adaptability to specific optimization challenges. The continuous evolution and refinement of the SFLA through these various modifications underscore its adaptability and continued relevance in tackling optimization problems across different domains.

Within the context of Wireless Sensor Networks (WSNs), the research conducted by Edla et al. [21] introduced an Improved Shuffled Frog Leaping Algorithm aimed at addressing the challenge of load balancing among gateways. The main focus of their work was to optimize the distribution of loads across gateways, a critical concern in WSNs. To evaluate the performance of ISFLA, Edla et al. [21] conducted a comprehensive comparative study against several existing load balancing techniques, including NLDLB [22], SBLB [23], SGA [24] and NGA [25]. Through their experimental analysis, the researchers demonstrated the distinct advantages of SFLA over these well-established load balancing methods. The results underscored the algorithm's ability to effectively balance the loads among gateways, thereby contributing to enhanced network performance in WSN scenarios. This study not only highlights the significance of load balancing in WSNs but also showcases the potential of SFLA as a promising solution to this critical challenge.

Primary objectives of the paper are summarized below.

- Formulate and implement a unique fitness function that incorporates novel parameters to optimize load balancing and energy consumption within Wireless Sensor Networks (WSNs). This fitness function is designed to intelligently allocate resources and distribute workloads, thereby enhancing the overall efficiency and sustainability of WSNs.
- The objective is to harness the power of multi-solution exchange, facilitating accelerated convergence rates, diversified exploration trajectories, and the generation of higher-quality solutions.
- Investigate and propose innovative partitioning approaches to strategically organize the population of frogs within the ESFLA framework. These strategies aim to optimize the allocation of frogs across memeplexes and sub-memeplexes, consequently shaping the algorithm's search dynamics and enhancing its performance.

This article is organized as follows. Literature review and problem definition is given in section 2 and 3 respectively. Section 4 is about proposed methodology. Section 5 presents input scenarios, obtained results and discussion. Finally, Section 6 is the conclusion.

2. Literature Review

In the realm of Wireless Sensor Networks (WSNs), the literature has explored various algorithms aimed at optimizing clustering and load balancing. Gupta and Jha [26] undertook the task of resolving the energy equilibrium predicament in node clustering and routing within WSNs. Their approach introduced an upgraded clustering protocol founded on the cuckoo search algorithm, optimizing the allocation of cluster heads with a fresh objective function. In addition, they introduced a more refined routing protocol utilizing the harmony search method for the transmission of data packets. Through comprehensive evaluations encompassing average energy usage, node status, and network longevity, their amalgamated Cuckoo-Harmony Search protocol displayed substantial enhancements compared to established methodologies. Edla et al. [27] propose a shuffled complex evolutionary (SCE) approach addressing load balancing, refining the fitness function and introducing a new allocation constraint. This constraint guides the allocation of sensors to gateways, particularly those farthest from the base station. Collectively, these approaches provide a comprehensive exploration of clustering and load balancing in WSNs, contributing valuable insights and solutions to a complex problem space. Wang et al. [1] presented an inventive energy-efficient clustering algorithm geared towards WSNs, incorporating an enhanced artificial bee colony (ABC) algorithm. This enhanced ABC strategy seamlessly incorporated essential elements including cluster head energy, density, and spatial positioning to effectively tackle the prevailing clustering issues within WSNs. Furthermore, their protocol introduced an energy-efficient routing algorithm through an enhanced ant colony optimization technique. Simulation outcomes revealed superior performance compared to recent analogous protocols. Archana [28] centred their efforts on energy-efficient load balancing and extending network lifespan by means of a fitness function-based algorithm. Their proposed techniques were rigorously researched and subjected to comparison with existing protocols using simulation parameters. By broadening load distribution and diminishing energy consumption, their strategy demonstrated promising strides in advancing network performance metrics.

Huamei et al. [29] shifted their focus towards the enhancement of energy efficiency and the equitable distribution of energy consumption within Wireless Sensor Networks (WSNs). Their approach centered around the development of an innovative energy-efficient clustering and routing protocol designed to address these critical issues. Their strategy encompassed the implementation of a non-uniform clustering network partition; a novel concept aimed at mitigating energy depletion challenges within the network. This partitioning approach was coupled with the utilization of an improved shuffled frog leaping algorithm, which played a pivotal role in optimizing the selection of cluster heads. Through rigorous simulation experiments, the research outcomes came to light. These experiments unveiled a substantial 20% improvement in energy efficiency within the WSN, a noteworthy achievement. Additionally, the proposed protocol contributed to the extension of the network's lifespan, marking a significant advancement in the quest for sustainable and efficient WSN architectures. Lipare et al. [30] introduced a novel method involving a modified particle swarm optimization (PSO) for fuzzy rule generation and clustering within WSNs. Their technique harnessed the Fuzzy C-means (FCM) approach for cluster formation, coupled with the Sugeno fuzzy system for cluster head selection. The introduced SF-MPSO algorithm outperformed established algorithms in terms of energy consumption and network longevity, thus showcasing the effectiveness of this modified PSO approach. Adamuthe and Pathan [31] proposed an enhanced version of the shuffled frog leaping algorithm (SFLA) with a heightened local exploration and energy-biased load reduction phase. Their strategy was formulated to enhance load balancing specifically within WSN gateways. By infusing variations into the SFLA algorithm to bolster local search capabilities, they achieved improved network energy consumption, load balancing, and overall network lifespan. In their research, Wang et al. [32] unveiled an energy-efficient cluster-based routing protocol for WSNs. Their method encompassed the utilization of the Self-Organizing Map neural network for initial node clustering. Further optimization of network clustering was achieved through the application of the firefly algorithm (FA), which factored in elements such as cluster head energy, average distance, and cluster quality. The introduction of decision domains further enhanced cluster dispersion. The subsequent inter-cluster routing phase involved an enhanced ant colony optimization (ACO) mechanism, integrating node attributes such as angle, distance, and energy to facilitate precise next-hop selection. The inclusion of a polling control mechanism during intra-cluster communication stages further uplifted network throughput. Simulation outcomes effectively showcased the prowess of the proposed EECRAIFA protocol in realizing balanced energy consumption, extended network lifespan, and improved throughput across diverse application scenarios. The research by Benelhouri et al. [33] contributed an enhanced gateway-based energy-aware multi-hop routing protocol customized for heterogeneous WSNs. This study expanded the applicability of the MGEAR protocol, originally devised for homogenous networks, to the realm of heterogeneous scenarios. Through an intelligent cluster-head selection process grounded in residual energy and regional averages, this strategy effectively balanced load distribution and extended the lifespan of sensor nodes. Simulation findings indicated substantial improvements in throughput and network longevity, particularly in the context of heterogeneous environments. The work by Moinyak et al. [34] tackled the challenges of routing and clustering within WSNs by harnessing an innovative approach centered on Spotted Hyena Optimization. Their study demonstrated significant potential in extending network longevity through the effective distribution of load among sensor nodes. The proposed strategy not only offered a robust solution to existing predicaments but also laid the groundwork for a more sustainable and efficient WSN architecture. The paper underscored the importance of groundbreaking methodologies in addressing intricate issues within the WSN realm. Chaurasia [35] unveiled the MOCRAW algorithm, a fusion of meta-heuristic optimization techniques designed to address energy consumption and data transmission challenges in WSNs. This algorithm harnessed dynamic cluster head selection for optimal routing, incorporating the Dragonfly Algorithm (DA) alongside both local and global search optimizations. By factoring in elements such as energy levels and distances, MOCRAW ensured efficient data transmission while simultaneously mitigating issues of isolated nodes and network overhead. Simulation-based evaluations convincingly demonstrated the superior energy efficiency and overall network performance exhibited by MOCRAW. The study conducted by Vijayvergia and Modani [36] focused on energy-efficient load balancing and routing through a multi-objective-based algorithm tailored for WSNs. The research addressed the challenge of energy consumption imbalance around gateways, introducing a methodology employing beetle swarm optimization (BSO) to achieve optimal clustering and routing. The proposed algorithm effectively curbed power consumption and surpassed existing methods in key metrics including energy consumption, delivery ratio, throughput, and network longevity, thus showcasing its potential to elevate WSN performance.

In the realm of modern research, there is a burgeoning interest in nature-inspired algorithms, which derive their inspiration from diverse natural phenomena and employ these concepts to tackle a wide spectrum of problems. In recent years, these algorithms have surged in popularity, gaining recognition as powerful and versatile problem-solving tools. Among the notable algorithms in this category, several have stood out for their effectiveness in addressing complex challenges. These algorithms include ray optimization, gravitational search algorithms, harmony search algorithms, flower pollination algorithms, cuckoo search, and bat algorithms. Each of these techniques draws analogies from the natural world, leveraging insights from various aspects of nature to create innovative problem-solving approaches. These nature-inspired algorithms have exhibited significant success in a variety of problem domains, highlighting their potential to provide novel solutions to intricate challenges. Their capacity to mimic and adapt natural processes offers researchers and practitioners a valuable toolbox for approaching complex problems in diverse fields.

3. Problem Definition

3.1. Network Energy Model

Wireless Sensor Networks (WSNs) utilize radio signals to establish communication among devices, with the size of these signals' contingent upon the volume of data being conveyed. The process of generating, transmitting, and receiving these signals necessitates energy consumption. To assess the energy consumption linked with these operations, researchers commonly adopt the energy model introduced by Heinzelman et al. [37]. This energy model takes into account the separation between the transmitting and receiving units and then selects the most suitable communication channel accordingly.

In situations where the distance between the units is shorter than a predefined threshold, denoted as 'δ', the model employs a free space channel. Conversely, if the distance surpasses 'δ', the model switches to a multi-path fading channel for communication. The calculation of the energy required for transmitting a message of length 'm' across a distance 'd' is accomplished using Equation (1).

$$\begin{aligned} E_T(m,d) &= m * E_{elec} + m * E_{fs} * d2, \text{if } d < \delta \\ (m,d) &= m * E_{elec} + m * E_{mp} * d4, if\ d \geq \delta \end{aligned} \tag{1}$$

In this context, E_{elec} represents the energy consumed by the electronic circuitry for each bit processed, E_{fs} denotes the energy expended per bit in a free space communication model and E_{mp} signifies the energy utilized per bit when utilizing a communication channel that experiences multipath fading.

Equation (2) concisely demonstrates the energy usage linked to a node while gathering a set of 'm' data bits within its effective communication range.

$$E_R(m) = m * E_{elec} \tag{2}$$

Equation (3) defines the energy required by gateway to aggregate m messages of x-bit each. In this equation E_{Aggr} is per-bit energy consumption to aggregate data in circuit.

$$E_A(m) = m * E_{Aggr} \tag{3}$$

3.2. Fitness Function Proposed by Edla et al. [21]

In the study by Edla et al. [21], they introduced a fitness function designed to assess the fitness level of each individual frog. This fitness function, as outlined in Equation (4), was formulated with the objective of reducing the workload on the gateways.

$$fitness = \left(1 - \frac{\mu_{load}}{G_{max}}\right) + \frac{\text{\#heavyandunderloadedgateways}}{\text{Total\#gateways}} \tag{4}$$

Here, μ_{load} denotes the average load distributed across all gateways. The calculation of μ_{load} is performed using Equation (5).

$$\mu_{load} = \sum_{i=1}^{i=m} Load(G_i)/\#gateways \tag{5}$$

The energy required by gateways to execute various processes is considered when calculating load. Equation (6) may be used to compute the load on each gateway.

$$Load(G_i) = m * \frac{E_{remain}(G_i)}{E_{initial}(G_i)} \tag{6}$$

The quantity of bits designated for transmission by the i[th] gateway is denoted as 'm'. Additionally, E_{remain} stands for the remaining energy of gateway G_i, while $E_{initial}$ represents its initial energy level. To compute the value of E_{remain}, Equation (7) is used. To determine $E_T(m,d)$ and $E_R(m)$, equations (1) and (2) are used respectively.

$$E_{remain} = E_{initial} - E_R(m) - E_T(m,d) \tag{7}$$

The tally of gateways positioned outside the range of the maximum (T_{max}) and minimum (T_{min}) thresholds is termed as the count of heavy and underloaded gateways. Gateways with a load exceeding T_{max} are classified as heavy loaded, while those with a load falling below T_{min} are categorized as underloaded. The determination of the threshold values T_{max} and T_{min} can be achieved through the utilization of Equation (8) and (9) respectively.

$$T_{max} = \mu_{load} + \gamma \tag{8}$$

$$T_{min} = \mu_{load} - \gamma \tag{9}$$

Equation (10) used to determine γ, taking into account the disparity between the load of the boundary gateway (R) and the overall count of gateways. The computation of 'R' is accomplished by applying Equation (11), where G_{max} denotes the maximum load of a gateway and G_{min} signifies the minimum load of a gateway.

$$\gamma = R \,/\, Total\#gateways \tag{10}$$

$$R = G_{max} - G_{min} \tag{11}$$

The problem is formulated as a task of minimizing, where the designation of the best frog is granted to the one with the lowest fitness value. The basis for this fitness measurement is drawn from energy consumption, thus prohibiting the possibility of attaining a value of zero or entering negative ranges.

3.3. Proposed Fitness Function

The objective function introduced by Edla et al. [21] was formulated to minimize the maximum load on gateways. Nonetheless, this approach doesn't provide significant benefits for gateways with the least remaining energy. Our emphasis was on safeguarding gateways from early exhaustion, aiming to extend their operational lifespan. The suggested energy-aware fitness function was devised to enhance the longevity of individual gateways, thus contributing to an overall improvement in the network's lifespan.

Furthermore, the proposed fitness function takes into account the energy consumption associated with gateway data aggregation. Equation 12 outlines the energy-aware fitness function designed for achieving load balancing among gateways in Wireless Sensor Networks (WSNs).

$$fitness = \sqrt{\frac{\sum_{i=1}^{i=N} (RE_i - \mu)^2}{N}} \tag{12}$$

Here, *'N'* is the total number of gateways in the given network. The mean residual energy of gateways (μ) is calculated from Equation 13.

$$\mu = \frac{\sum_{i=1}^{i=N} RE_i}{N} \tag{13}$$

In the above equation, RE_i is the residual energy of gateway i. The residual energy is calculated from Equation 14.

$$RE_i = AE_i - CE_i \tag{14}$$

Here,

- AE_i is the available energy in the gateway 'i' before any allocation
- CE_i is consumed energy by the gateway 'i'.

Diverse tasks engage gateways, encompassing activities such as gathering data from sensors, consolidating data, and transmitting data to the base station. Each of these undertakings entails a certain level of energy consumption. The energy usage of gateway 'i' is calculated using Equation 15.

$$CE_i = E_R(m) + E_T(m,d) + E_A(m) \tag{15}$$

Energy values for $E_R(m)$, $E_T(m,d)$ and $E_A(m)$ are computed using Equations 1, 2 and 3 respectively. The proposed fitness function should be used in minimization means a solution with minimum fitness value is considered the best solution.

4. Methodology

This section presents the methodology of SFLA, ISFLA and the proposed SFLA approach with different partitioning approaches.

4.1. Shuffled Frog Leaping Algorithm

The approach of the shuffled frog leaping algorithm (SFLA), as documented by [6,38], is outlined within this subsection. SFLA has demonstrated its effectiveness in solving multiple real-world problems, consistently yielding optimal solutions. Within SFLA, each individual solution is referred to as a "Frog," and these frogs interact with one another to enhance their respective fitness levels. The flowchart of the SFLA is illustrated in figure 3. The sequential stages of the SFLA process are elaborated upon below.

- Population Initialization: An assortment of frogs is generated through random processes. This collection is represented as the population denoted by P = {P1, P2, … Px}, where 'x' signifies the population size.
- Evaluation of Fitness and Frog Sequencing: During this stage, a numerical fitness value is calculated for each

individual frog within P, utilizing a fitness function tailored to the specific problem. These frogs are then arranged in descending order based on their computed fitness values.

- Formation of Memeplexes and Sub-memeplexes: In this step, the ordered population of frogs in P is divided into 'm' subsets (memeplexes), each containing 'n' frogs, resulting in P = m x n. The arrangement is structured so that the frog with the highest fitness value occupies the initial subset, followed by the subsequent frog in the second subset, and so on. Moreover, each memeplex is subdivided into a predetermined quantity of sub-memeplexes.
- Local Evolution: As depicted in figure 4, each sub-memeplex undergoes independent evolution. During this evolutionary process, information is exchanged between the best and worst frogs within each sub-memeplex.
- Evaluation of Convergence: The evaluation of convergence criteria takes place at the end of each iteration. If these criteria are met, the execution is stopped; otherwise, the algorithms proceed to the next iteration. Convergence conditions could encompass reaching a predefined maximum iteration count or achieving a designated threshold fitness value.

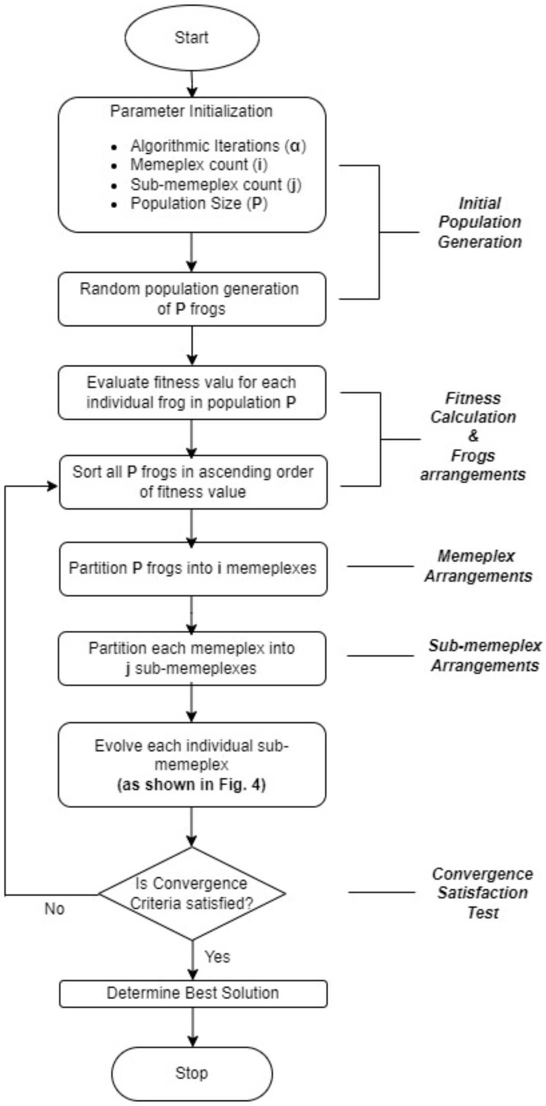


Fig.3. Enhanced shuffled frog leaping algorithm (ESFLA) flowchart

Edla et al. [21], adopted and presented an improved version of SFLA to accommodate its use in load balancing of WSN's. Edla et al. [21] tweaked the algorithm in a couple of stages.

- Initialization of Population: Instead of allocating any random gateway to sensor nodes, Edla et al. [21], implemented the validation that any gateway only within its contact range be allocated. This assures that all the generated solutions are valid.
- Modification in Offspring Generation Stage: Within the context of SFLA, the phase involving the creation of offspring involves generating two offspring. Occasionally, these offspring might not exhibit better fitness values compared to their parents. To address this scenario, the Improved SFLA (ISFLA) introduces a

refinement wherein a single offspring is produced. This is achieved by exchanging information between the best and worst solutions, accomplished through a single-point crossover. This adjustment serves to maintain the integrity of the best solution.

- Introduction of an Innovative Transfer Phase: A new phase, termed the 'transfer phase,' has been introduced. In this phase, a gateway experiencing a high load is selected, and its load is eased by relocating the farthest sensor node covered by that gateway to another gateway situated within the range of that distant sensor node. This strategy aids in balancing the loads of gateways and subsequently reduces the overall energy consumption of gateways. This phase bears resemblance to the mutation phase observed in Genetic Algorithms (GAs).

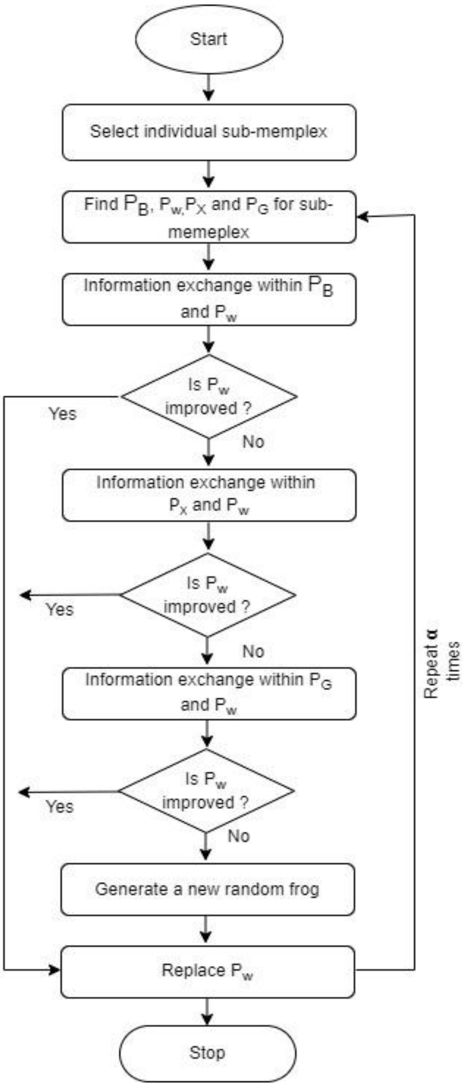


Fig.4. Local evolution phase of enhanced shuffled frog leaping algorithm

Adamuthe & Pathan [31] presented an advanced iteration of the Shuffled Frog Leaping Algorithm, denoted as ESFLA, building upon the foundation established by the Improved Shuffled Frog Leaping Algorithm (ISFLA) introduced by Edla et al. [21]. This enhanced algorithm is specifically tailored for enhancing load balancing in Wireless Sensor Networks (WSNs). Fig. 3 and 4 presents ESFLA algorithm and local evolution phase respectively. ESFLA introduces several notable enhancements that contribute to its performance compared to ISFLA. Three variations in Local evolution phase are as follows.

- ESFLA-GG: In this variant, termed ESFLA-GG, the approach involves substituting the underperforming frogs within subsequent sub-memeplexes with the globally best-performing frog.
- ESFLA-SS: This iteration, referred to as ESFLA-SS, integrates a stochastic sampling operator driven by probabilities to select both source and target frogs in the information exchange process.
- ESFLA-RW: The ESFLA-RW version employs a probabilistic roulette wheel operator to determine the source frog. Meanwhile, the least performing frog within a sub-memeplex is the default selection for the target frog. This strategy fosters a more diverse and wide-ranging exchange of information.

- The research article introduces an original phase specifically designed to alleviate the load of gateways characterized by the lowest remaining energy levels. This novel strategy targets the enhancement of load distribution equilibrium and the extension of Wireless Sensor Network (WSN) gateway lifespan.

4.2. Proposed Hybrid Enhanced SFLA with Multi Solution Evolution Phenomenon (ESFLA-MSE)

The multi-solution evolution paradigm (MSEP) is a concept in evolutionary computation and optimization that focuses on evolving a population of solutions to address complex problems. Unlike traditional single-solution approaches, which aim to find a single optimal solution, MSEP emphasizes generating and maintaining multiple diverse solutions. ESFLA operates within the multi-solution evolution paradigm by maintaining a diverse set of solutions throughout its process. The Evolution phenomenon in the Shuffled Frog Leaping Algorithm (SFLA) refers to the crucial step where information is shared from the best-performing frogs to the least-performing frogs within each sub-memeplex. This process helps in transferring valuable knowledge from the most successful solutions to those that need improvement. However, a limitation of this phenomenon is that, in each iteration, only the weakest frog within each sub-memeplex is improved, potentially slowing down the overall optimization process. To enhance the efficiency of this information exchange mechanism, proposed a new approach called the Enhanced SFLA with Multi-Solution Evolution (ESFLA-MSE).

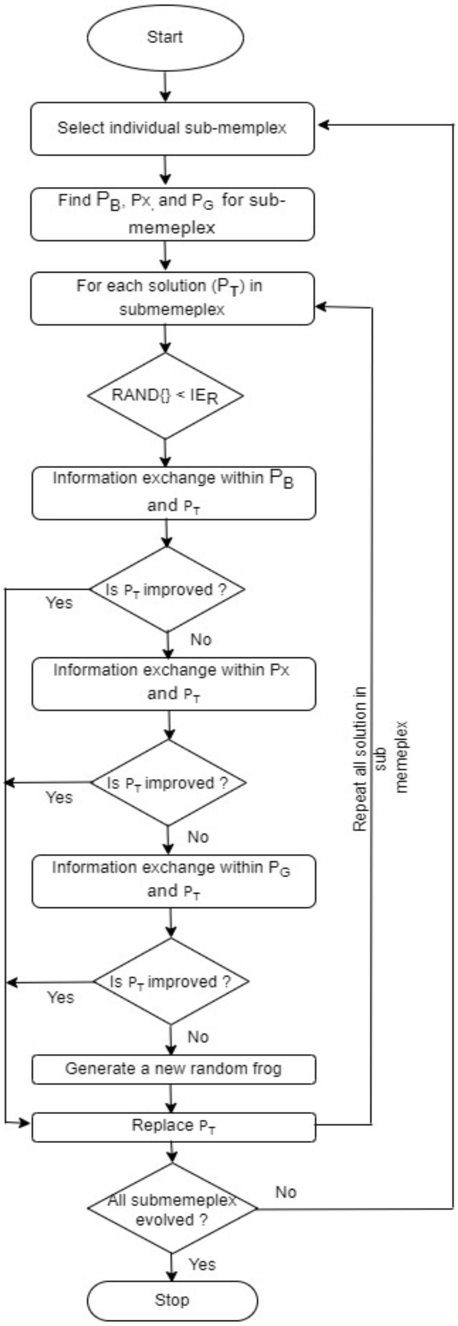


Fig.5. Enhanced SFLA with multi-solution evolution (ESFLA-MSE) flowchart

This hybrid method combines concepts from other optimization algorithms, specifically the Jaya and Teaching-Learning-Based Optimization (TLBO) algorithms. These algorithms are known for their ability to motivate the improvement of all solutions in the population, rather than just the weakest ones. The integration of Jaya and TLBO concepts involves incorporating strategies from these algorithms into the SFLA framework. These strategies encourage all solutions within a sub-memeplex to progress, rather than just focusing on the weakest solution. This is a departure from the traditional SFLA approach where only the weakest frog is improved. Drawing inspiration from Genetic Algorithms, introduced the concept of a crossover rate, which termed as the Information Exchange Rate (IER). This rate controls the likelihood of a solution participating in the information exchange process. Solutions are assigned random values between 0 and 1, and their engagement in the information exchange phase depends on whether their generated random value is within the specified IER range. The proposed approach's evolution process is illustrated through a flowchart in figure 5. This flowchart outlines how the multi-solution evolution mechanism works within the ESFLA-MSE framework. It provides a visual representation of how solutions are selected for information exchange based on the IER and how this process differs from the traditional SFLA.

4.3. Explored Partitioning Approaches

In SFLA, there exists a phase wherein several frogs are divided into memeplexes, and subsequently, these memeplexes are subdivided into a couple of sub-memeplexes, allowing independent evolution of each sub-memeplex with its assigned frogs. In order to assess the influence of partitioning logic on the algorithm's performance, various memeplex partitioning logics were investigated. Furthermore, the ramifications of these partitioning logics were quantified and the results were compared to confirm the precise effects of these logics.

A. Approach of Original SFLA

The partitioning method was introduced in SFLA by Eusuff et al. [6]. This method involves allocating frogs in such a way that the initial frog is positioned within the first memeplex, the second frog within the second memeplex, the i^{th} frog within the i^{th} memeplex, and the $(i+1)^{th}$ frog once again within the first memeplex. An illustration of the original partitioning approach can be observed in figure 6. Algorithm 1 provides the pseudo code outlining this approach.

Algorithm 1: Original SFLA Partitioning Approach

Input: Set of Frogs(*F*), No. of memeplexes to be formed (*m*)
Results: Set of memeplexes created by partitioning the set of frogs
Procedure partitionFrogs(F, m):
1. N := number of frogs := count(F);
   // Initialize empty memeplexes array
2. Mem ← Empty array of *m* positions
3. **for** i := 0 **to** N **do**
     j := i % m
     Assign i^{th} frog (F_i) to j^{th} memeplex ($Mem_j \leftarrow F_i$)
4. **end for**
5. **return** Mem

B. Enhanced Shuffled Frog Leaping Algorithm - Reverse Partitioning (ESFLA-IF-RV)

In this method, the distribution of Frogs follows a pattern: the initial frog is assigned to the first memeplex, the subsequent frog is assigned to the second memeplex, and the i^{th} frog is designated to the i^{th} memeplex. Upon entering the second round of allocation, the $(i+1)^{th}$ frog occupies the ith memeplex, the $(i+2)^{th}$ frog occupies the $(i-2)^{th}$ memeplex, and so forth. Algorithm 2 outlines the pseudo code for this procedure.

Algorithm 2: ESFLA-IF-RV Partitioning Approach

Input: Set of Frogs(*F*), No. of memeplexes to be formed (*m*)
Result: Set of memeplexes created by partitioning the set of frogs
Procedure partitionFrogs(F, m):
1. N := number of frogs := count(F);
2. **for** i := 0 **to** N **do**
batch = floor(i / m)
if batch % 2 == 0:
  j = i % m
else:
  j = m - (i % m) - 1
Assign i^{th} frog (F_i) to j^{th} memeplex ($Mem_j \leftarrow F_i$)
3. **End for**
4. **return** Mem

C. Enhanced Shuffled Frog Leaping Algorithm - Random Partitioning (ESFLA-IF-RD)

In this approach, Frogs are distributed randomly in any memeplexes. We made it sure the number of frogs in all memeplexes were nearly equal. Algorithm 3 presents the pseudo code for the approach.

Algorithm 3: ESFLA-IF-RD Partitioning Approach

Input: Set of Frogs(*F*), No. of memeplexes to be formed (*m*)
Result: Set of memeplexes created by partitioning the set of frogs
Procedure partitionFrogs(F, m):
1. **while** F is not empty **do**
2. **for** j := 0 **to** m **do**
// Get a random frog number from F
i := rand(F)
Assign i^{th} frog (F_i) to j^{th} memeplex ($Mem_j \leftarrow F_i$)
3. **endfor**
4. **end while**
5. **return** Mem

D. Enhanced Shuffled Frog Leaping Algorithm - Roulette Wheel Partitioning (ESFLA-IF-RW)

In this approach, Frogs for every memeplex is chosen using the Roulette wheel selection. Once the frog is assigned to a memeplex, it is removed from the frogs list so that a fresh roulette wheel could be generated with the updated frog list. We made it sure that duplication of frogs is avoided in this approach. Algorithm 4 presents the pseudo code for the approach.

Algorithm 4: ESFLA-IF-RW Partitioning Approach

Input: Set of Frogs(F), Fitness of all frogs in F (Fitness), No. of memeplexes to be formed (m)
Results: Set of memeplexes created by partitioning the set of frogs
Procedure partitionFrogs(F, Fitness, m):
1. **while** F is not empty **do:**
for j:=0 to m **do**
i := getRouletteFrog(F, Fitness)
Assign i^{th} frog (F_i) to j^{th} memeplex ($Mem_j \leftarrow F_i$)
Remove F_i from F
end for
2. **end while**
3. **return** Mem

E. Enhanced Shuffled Frog Leaping Algorithm - Roulette Wheel with Duplication Partitioning (ESFLA-IF-RWD)

In this approach, Frogs for every memeplex is chosen using the Roulette wheel selection. Once the frog is assigned to a memeplex, it is not removed from the frogs list so that a duplication possibility is maintained. Algorithm 5 presents the pseudo code for the approach.

Algorithm 5: ESFLA-IF-RWD Partitioning Approach

Input: Set of Frogs(F), Fitness of all frogs in F (Fitness), No. of memeplexes to be formed (m)
Results: Set of memeplexes created by partitioning the set of frogs
Procedure partitionFrogs(F, Fitness, m):
1. count := count(F)
2. **for** i := 0 **to** count
For j:=0 to m
I := getRouletteFrog(F, Fitness)
Assign i^{th} frog (F_i) to j^{th} memeplex ($Mem_j \leftarrow F_i$)
end for
3. **End for**
4. **Return Mem**

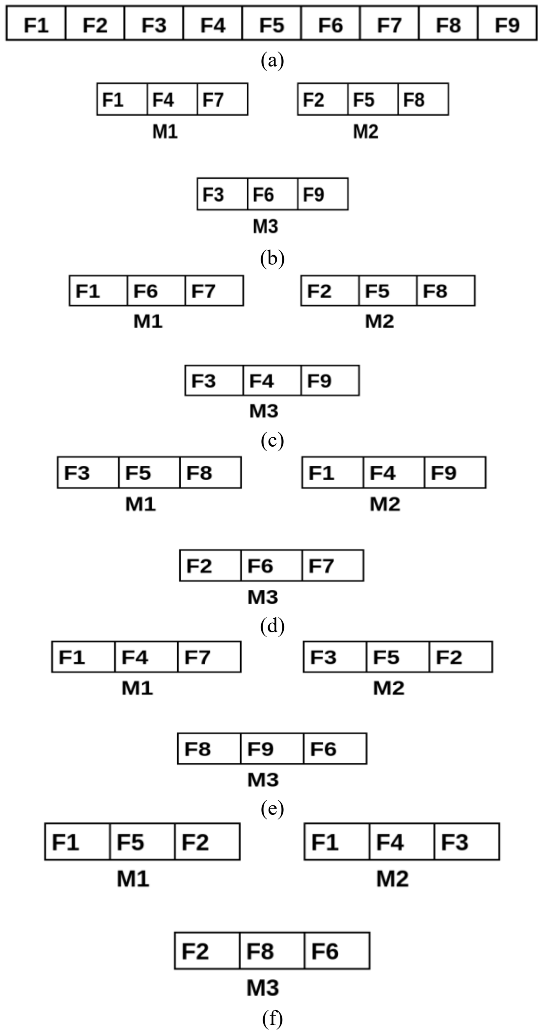


Fig.6. Example of proposed partitioning approaches (a) Frogs before applying partitioning (b) After partitioning using original SFLA approach (c) After partitioning using ESFLA-IF-RV approach (d) After partitioning using ESFLA-IF-RD approach (e) After partitioning using ESFLA-IF-RW approach (f) After partitioning using ESFLA-IF-RWD approach

4.4. ESFLA Steps and Operations

A. Representation and Initialization of a Frog

In the given context, the term “frog” symbolizes the distribution of gateways assigned to individual sensors. The frog's length corresponds to the total number of sensors within the Wireless Sensor Network (WSN). Each sensor is associated with a designated gateway located within its communication radius. To visualize this solution concept, please refer to figure 7, which offers a descriptive example. In this illustration, sensors s4 and s5 are paired with gateway g1, whereas sensors s1, s6, and s10 are connected to gateway g2.

s	1	2	3	4	5	6	7	8	9	10
g	2	3	4	1	1	2	3	4	4	2

Fig.7. Representation of a sample frog

During the initialization phase, a predetermined count of frogs is generated in a random manner. Each individual frog is crafted by randomly assigning a gateway to every sensor node, while considering the applicable interaction range. This systematic approach ensures the feasibility of each generated frog in the initialization phase, as gateways are only assigned to sensor nodes within their interaction range.

To exemplify, let's consider a Wireless Sensor Network (WSN) consisting of 10 sensor nodes labeled as S (s1 to s10) and 4 gateways denoted as G (g1, g2, g3, g4). The arrangement of sensor nodes and gateways within their interaction range is outlined in table 1. According to the information in table 1, sensor node s1 can be paired with either g2 or g3, while sensor node s2 has the potential to be linked to g1, g2, or g4. To illustrate this scenario, figure 8 provides a visual representation of a viable frog.

Table 1. Sensor nodes and gateways details

Sensor node	Gateways within range	Sensor node	Gateways within range
s_1	g_2, g_3	s_6	g_3, g_4
s_2	g_1, g_2, g_4	s_7	g_1, g_2
s_3	g_2, g_4	s_8	g_2, g_3, g_4
s_4	g_1, g_3, g_4	s_9	g_4
s_5	g_3, g_4	s_{10}	g_1, g_4

s	1	2	3	4	5	6	7	8	9	10
g	2	1	4	3	3	4	2	3	4	1

Fig.8. A valid frog for example 1

B. Fitness Evaluation

The determination of the fitness for each individual frog is achieved by applying a fitness function as detailed in section 3.

C. Memeplex Arrangements

The frogs generated during the initialization stage are organized into a predetermined number of memeplexes. This organization is achieved by initially ordering all frogs in increasing order, with their fitness values as the guiding factor. Following this arrangement, the distribution to memeplexes follows a specific pattern outlined in Equation 16. According to this distribution approach, the initial frog is assigned to the initial memeplex, the ith frog is allocated to the ith memeplex, and the (i+1)th frog is subsequently positioned in the first memeplex again.

$$memeplex(f) = pos(f) \bmod m \tag{16}$$

To provide an example, consider a situation where 8 frogs are generated during the initial phase and then sorted in ascending order according to their fitness values, labeled as f1 to f8, as demonstrated in figure 9. The allocation of these 8 sorted frogs into 2 memeplexes, following the specified distribution pattern, is illustrated in figure 10.

1	2	3	4	5	6	7	8	9	10
3	2	4	1	3	4	2	2	4	1

(f1)

1	2	3	4	5	6	7	8	9	10
2	4	2	3	4	3	1	4	4	1

(f2)

1	2	3	4	5	6	7	8	9	10
2	4	2	1	3	4	1	3	4	4

(f3)

1	2	3	4	5	6	7	8	9	10
3	1	4	3	4	3	2	4	4	1

(f4)

1	2	3	4	5	6	7	8	9	10
2	4	2	3	4	3	1	2	4	1

(f5)

1	2	3	4	5	6	7	8	9	10
3	2	4	1	3	4	2	3	4	4

(f6)

1	2	3	4	5	6	7	8	9	10
3	1	4	3	4	4	2	2	4	1

(f7)

1	2	3	4	5	6	7	8	9	10
2	2	4	1	3	3	1	4	4	1

(f8)

Fig.9. Sample 8 frogs generated in the initial population

Memeplex 1	f1	f3	f5	f7
Memeplex 2	f2	f4	f6	f8

Fig.10. Sample memeplex arrangement

D. Sub-memeplex Arrangements

Following the initial arrangement of frogs into memeplexes, an additional level of subdivision is introduced. In this phase, each memeplex is divided into j sub-memeplexes, where the specific value of 'j' is determined randomly from the factors associated with the size of the memeplex itself. Here, the size of a memeplex signifies the total count of frogs it contains.

To illustrate this process, consider the example presented in figure 11, where the memeplex size is determined to be 4. The factors of 4 encompass 1, 2, and 4. Consequently, the value of j can be selected from the set {1, 2, 4}, reflecting the available factors. This selection of 'j' plays a role in further dividing the memeplex into its constituent sub-memeplexes. The visual representation depicted in figure 11 provides a clear demonstration of how each memeplex from the previously shown figure 10 is partitioned into 2 sub-memeplexes. In this specific instance, a total of 4 sub-memeplexes are generated, identified as (i1, j1), (i1, j2), (i2, j1), and (i2, j2). This process of subdivision adds an additional layer of organization to the frog population, enhancing the diversity of solutions explored during the optimization process.

	Submemeplex (j_1)		Submemeplex (j_2)	
Memeplex (i_1)	f1	f3	f5	f7
Memeplex (i_2)	f2	f4	f6	f8

Fig.11. Partitioning of memeplexes into 2 sub-memeplexes

E. Convergence Satisfaction Test

After each iteration, the algorithm evaluates convergence criteria. If these criteria are fulfilled, the execution is terminated. Conversely, if the criteria are not met, the algorithm advances to the subsequent iteration. Convergence criteria usually encompass conditions like attaining a predetermined threshold fitness value or reaching the maximum permissible iteration count.

5. Experimental Details, Results and Discussion

The following segment introduces the experimental setup, outlines the datasets employed, presents the result comparison, and engages in a comprehensive discussion.

5.1. Input Scenario

A total of 15 randomized input datasets has been generated to facilitate algorithm experimentation, encompassing a spectrum ranging from networks with low density to those exhibiting high density. These datasets have been systematically categorized into three distinct groups: small, medium, and large. These categorizations are based on several factors, including the area of deployment, the quantity of sensors involved, and the count of gateways employed. Specifically, input datasets 1 to 6 fall within the small category, input datasets 7 to 10 correspond to the medium category, and input datasets 11 to 15 are attributed to the large category. The detailed specifications for each input dataset are succinctly outlined in table 2.

Table 2. Description of generated input datasets

Input	Max threshold distance (m)	Deployment Area (mxm)	Location of Base Station (x, y)	No. of sensor nodes	No. of gateway
1	10	50x50	(48, 25)	50	10
2	10	50x50	(48, 25)	100	20
3	10	50x50	(48, 25)	200	30
4	10	100x100	(96, 50)	500	50
5	10	100x100	(96, 50)	750	65
6	10	100x100	(96, 50)	1000	80
7	15	200x200	(196, 100)	1200	100
8	15	200x200	(196, 100)	1400	120
9	15	200x200	(196, 100)	1600	140
10	15	200x200	(196, 100)	1800	160
11	15	300x300	(296, 150)	2000	150
12	15	300x300	(296, 150)	2250	175
13	15	300x300	(296, 150)	2500	200
14	15	300x300	(296, 150)	2750	225
15	15	300x300	(296, 150)	3000	250

Table 3 depicts simulation parameters used throughout the experimentations. Figure 12 is a visual representation of deployment for input 3 which consists of 200 sensors and 30 gateways. In figure 12, blue dots represent sensors, red dots represent gateways and green dots are base stations. The 'x' and 'y' coordinates of these components are recorded in the input dataset for further calculations.

Table 3. Simulation parameters

Description	Variable Name	Value
Default Initial Gateway Energy	$E_{initial}$	10000000 nJ
Sensor Data Length	l	8 bits
Energy Required by Circuit	E_{elec}	50 nJ/bit
Energy Required to Aggregate	E_{Aggr}	5 nJ/bit/msg
Energy Required by Free Space	E_{fs}	10 pJ/bit/m2
Energy Required by Multi-Path	E_{mp}	0.001 pJ/bit/m4
Frog Population Size	*PopSize*	100
Max Convergence Iterations	α	50

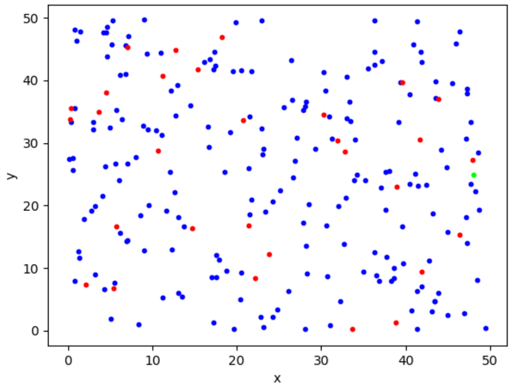


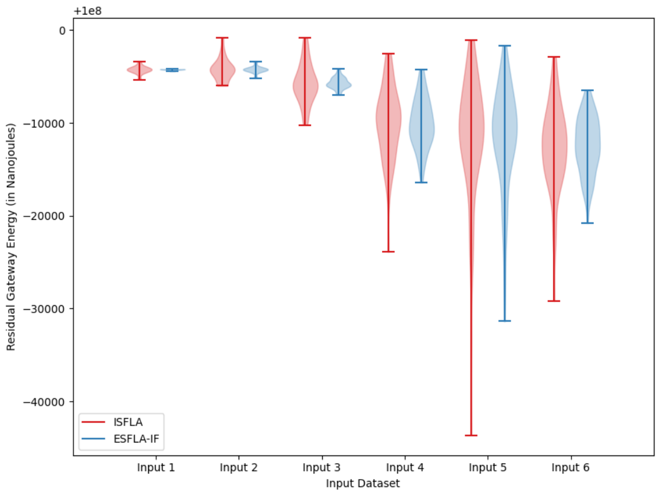
Fig.12. Visual deployment of 200 sensors & 30 gateways in 50x50 m^2

5.2. Results of Proposed Fitness Function

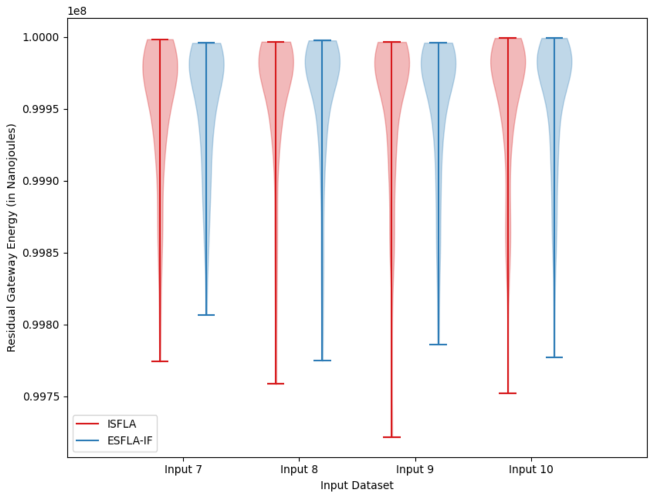
Performance of proposed fitness function is compared with fitness function presented in [21]. We have implemented the ISFLA algorithm for optimizing the gateway load balancing in WSN. The ISFLA algorithm is implemented with a new fitness function. This variation is termed as Enhanced Shuffled Frog Leaping Algorithm – Improved Fitness (ESFLA-IF). The variant ESFLA-IF is compared with ISFLA in terms of residual energy of gateways, network energy consumption and network lifetime.

A. Results of Residual Gateways Energy

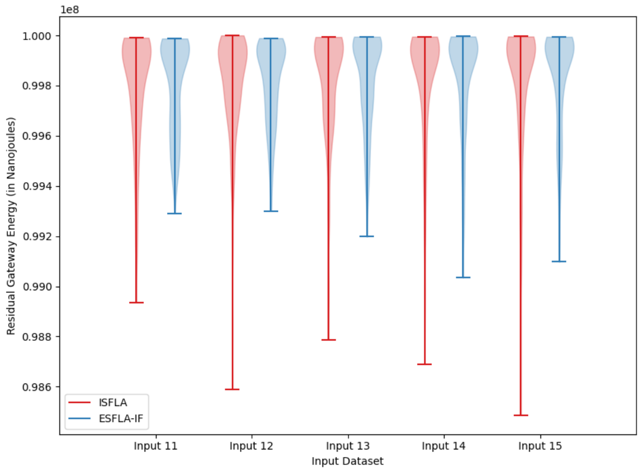
To gauge the effectiveness of load balancing, a comparison was conducted between the residual energy levels of gateways resulting from the best solutions obtained through ISFLA and ESFLA-IF.



(a)



(b)



(c)

Fig.13. Comparison of algorithms using residual gateway energy (a) small instances (b) medium instances (c) large instances

For each algorithm, the optimal solution attained after the final iteration is considered. The energy consumption of each gateway within the solution is computed. By subtracting these computed energy consumptions from the initial energy levels of the gateways, the residual energy of all gateways is determined. This experiment was conducted across all 15 input datasets. The outcomes are visually represented using a violin plot in figure 13. The results highlight a narrower spread in the differences among residual gateway energies in the case of ESFLA-IF when contrasted with ISFLA. A reduced spread indicates a more balanced distribution of the workload among gateways. This equitable distribution not only enhances the efficiency of load balancing but also extends the energy lifetime of the gateways.

B. Results of Gateways Energy Consumption

The energy consumption of gateways is evaluated based on the optimal solution obtained from the algorithm. This metric quantifies the total energy expended by all gateways for activities such as collecting, aggregating, and transmitting data from sensors to the base station. The computation of gateway energy consumption follows equations 1 to 3. The results are compiled across all 15 input datasets, with each dataset subjected to 10 execution instances to mitigate the effects of randomness. The average values obtained from these 10 executions are presented in table 4. The results unambiguously illustrate a noteworthy reduction in gateway energy consumption when employing ESFLA-IF compared to ISFLA. This improvement can be attributed to the proposed fitness function, which takes into account the residual energy of gateways when making decisions about sensor allocation. This intelligent allocation strategy leads to a more efficient distribution of tasks among gateways, contributing to the observed reduction in energy consumption.

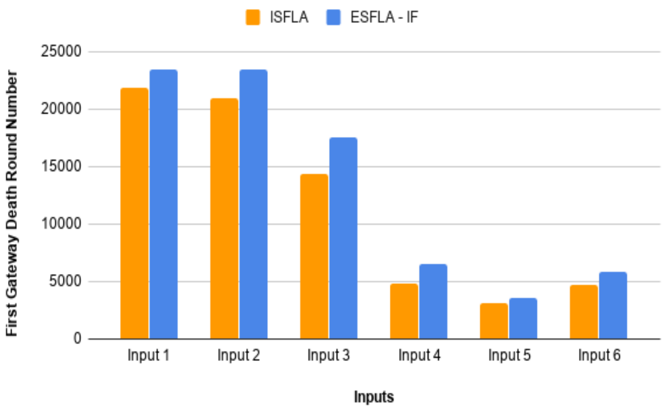
C. Comparison of Network Lifetime

The network lifetime in Wireless Sensor Networks (WSNs) signifies the duration during which all gateways operate efficiently. Once the first gateway fails, the network's stability begins to deteriorate. In order to evaluate the influence of the suggested fitness functions on network lifetime, a comparison was made regarding the round number at which the initial gateway ceases functioning. A gateway is regarded as non-operational when its remaining energy is fully depleted. The results were computed across all 15 input datasets, with each dataset executed 10 times to mitigate the impact of randomness. The averaged outcomes of these 10 executions are visually depicted in figure 14.

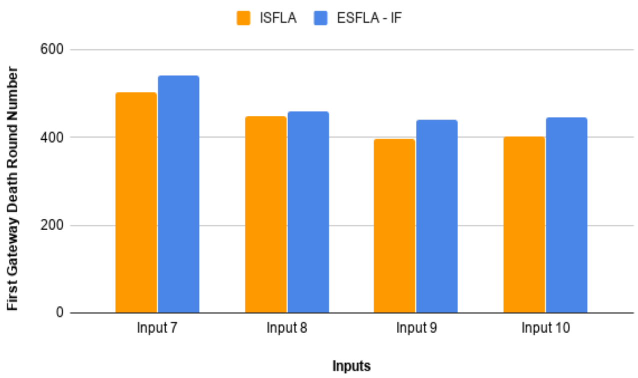
The findings unambiguously demonstrate that the network lifetime is substantially extended with the utilization of ESFLA-IF in contrast to ISFLA. This enhancement can be attributed to the ingenious fitness function proposed, which intelligently allocates sensor nodes to gateways while considering the residual energy available in each gateway at the time of allocation. This approach effectively contributes to a more balanced distribution of the network's load, leading to the observed increase in network lifetime.

Table 4. Comparison of network energy consumption (in nanoJoules)

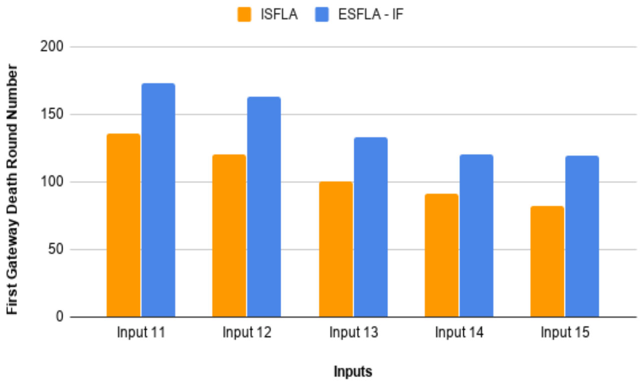
Input Datasets	ISFLA	ESFLA-IF
Input 1	42672	42657
Input 2	85159	85152
Input 3	170526	170425
Input 4	504305	500892
Input 5	771602	770713
Input 6	999287	992955
Input 7	4781510	4747246
Input 8	5074812	5019357
Input 9	6535455	6505010
Input 10	7269752	7232107
Input 11	30816469	29821903
Input 12	33595581	32784521
Input 13	37898615	36667472
Input 14	40620318	39157712
Input 15	45388804	43900232



(a)



(b)



(c)

Fig.14. Comparison of first gateway die (a) small instances (b) medium instances (c) large instances

The table 5 presents the improvement in network lifetime achieved by ESFLA-IF (Enhanced Shuffled Frog Leaping Algorithm with Improved Fitness) compared to ISFLA (Improved Shuffled Frog Leaping Algorithm) in terms of percentage improvement for various input scenarios. In all fifteen different input scenarios (Input 1 to Input 15), ESFLA-IF demonstrates a significant improvement in network lifetime compared to ISFLA. The percentage improvement varies across different input scenarios, ranging from 2.60% (Input 8) to as high as 45.60% (Input 15). This indicates that the effectiveness of ESFLA-IF in enhancing network lifetime depends on the specific characteristics and requirements of the network scenario. Input scenarios 11, 12, 13, 14, and 15 show substantial improvements of 27.54%, 35.24%, 32.20%, 32.71%, and 45.60%, respectively. These scenarios likely represent network conditions or configurations where ESFLA-IF's optimization techniques have a particularly strong positive impact on network lifetime. Input scenarios 2, 5, and 6 exhibit moderate improvements in the range of 12.19% to 23.93%. These scenarios indicate that ESFLA-IF consistently provides noticeable enhancements under various conditions. The consistent improvements across all scenarios highlight the effectiveness of ESFLA-IF as an optimization algorithm for extending the network lifetime in wireless sensor networks. It can be considered a robust and reliable choice for improving network performance in a wide range of scenarios.

Table 5. Improvement in network lifetime of ESFLA-IF over ISFLA (in percentage)

Input	Improvement in %	Input	Improvement in %
Input 1	7.30	Input 9	10.60
Input 2	12.19	Input 10	10.35
Input 3	21.54	Input 11	27.54
Input 4	34.20	Input 12	35.24
Input 5	12.58	Input 13	32.20
Input 6	23.93	Input 14	32.71
Input 7	7.28	Input 15	45.60
Input 8	2.60		

5.3. Results to Proposed ESFLA-MSU

In order to evaluate the efficacy of the introduced multi-solution evolution phenomenon within the ESFLA-MSU algorithm, a comparative analysis was conducted against ESFLA-IF. Both ESFLA-IF and ESFLA-MSU employed the novel fitness function that was proposed. The comparison encompassed multiple performance metrics, including best fitness values, mean fitness values, standard deviation, and convergence rate. The purpose of this comparison was to gauge how the ESFLA-MSU algorithm, with its enhanced multi-solution evolution approach, performs in relation to the baseline ESFLA-IF. The outcome of this analysis helps to elucidate the advantages and potential improvements achieved by integrating the multi-solution evolution phenomenon within ESFLA.

A. Performance Comparison of ESFLA-MSU with ESFLA-IF

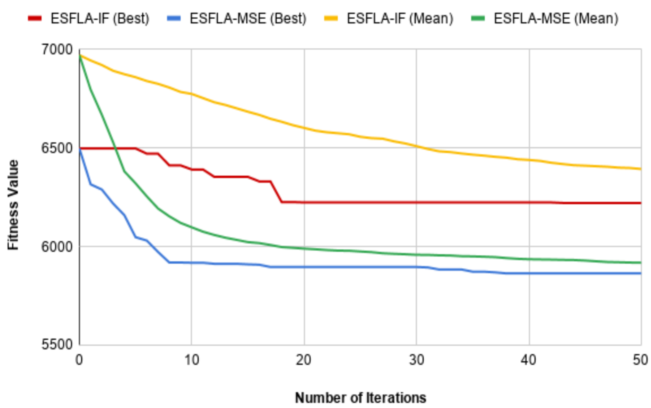
Table 6 presents the averaged outcomes derived from 10 execution runs, encompassing the finest fitness values, mean fitness values, and standard deviations. The “best fitness” denotes the fitness value of the most optimal solution achieved during the evolution process. On the other hand, the “mean fitness” represents the collective average fitness across all solutions within the population during the final iteration. The findings underscore that the ESFLA-MSU exhibits superior performance when juxtaposed with ESFLA-IF. This superiority is evident in multiple aspects, including improved best fitness values, elevated mean fitness values, and reduced standard deviation.

Table 6. Performance comparison of ESFLA-MSE and ESFLA-IF

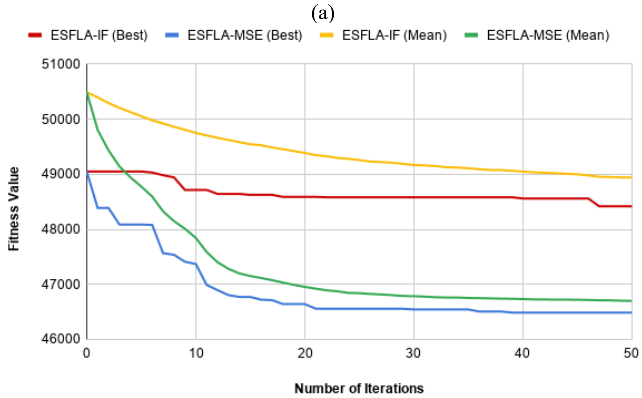
Input	Best Fitness Value		Mean Fitness Value		Standard deviation	
	ESFLA-IF	ESFLA-MSE	ESFLA-IF	ESFLA-MSE	ESFLA-IF	ESFLA-MSE
Input1	332	79	600	148	97	90
Input2	697	243	986	349	115	82
Input3	966	423	1267	563	107	95
Input4	3078	2685	3334	2794	93	79
Input5	6117	5923	6323	5978	79	50
Input6	3742	3101	3994	3229	95	89
Input7	44467	43773	45047	43863	263	83
Input8	48582	48086	49105	48146	253	62
Input9	50201	49059	50853	49217	289	146
Input10	47848	47081	48441	47268	237	144
Input11	195070	190714	199826	193093	2022	1439
Input12	173086	166456	177133	168638	1734	1593
Input13	197055	188489	201776	190768	1762	1509
Input14	215095	206443	220046	208454	1991	1577
Input15	214230	205746	218700	207573	1896	1446

B. Convergence Rate of ESFLA-MSU and ESFLA-IF

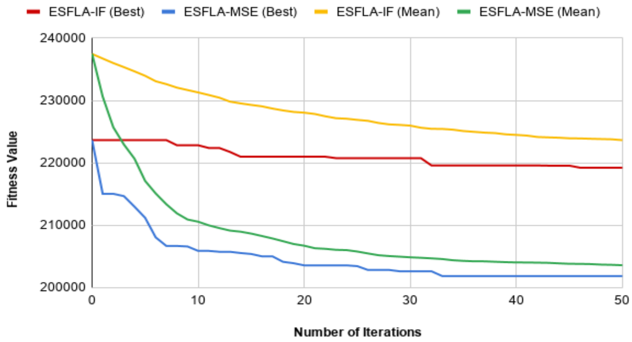
The experimental trials were conducted using representative input datasets from each of the distinct categories: small, medium, and large. Specifically, Input 5 was selected from the small category, Input 10 from the medium category and Input 15 from the large category. To visualize the findings, figure 15 portrays the convergence patterns of both ESFLA-MSU and ESFLA-IF algorithms for these chosen input datasets. The outcomes of the study consistently indicate that ESFLA-MSU exhibits swifter convergence compared to ESFLA-IF. This observation holds true for both the best fitness values and the average population fitness.



(a)

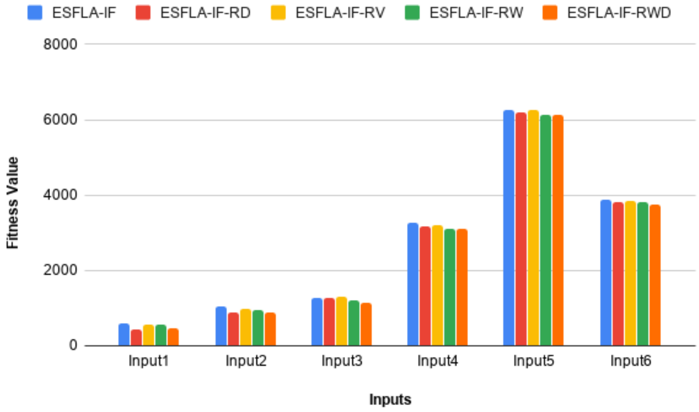


(b)

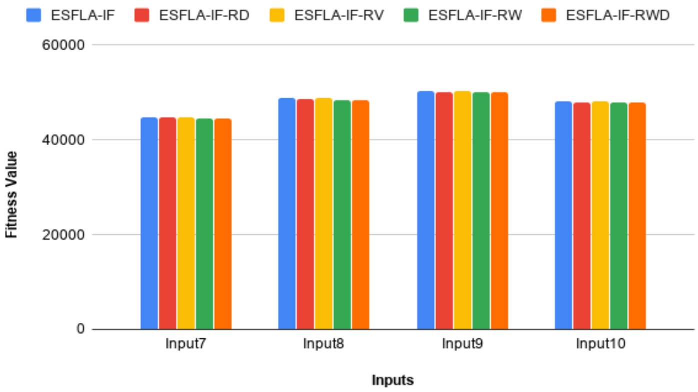


(c)

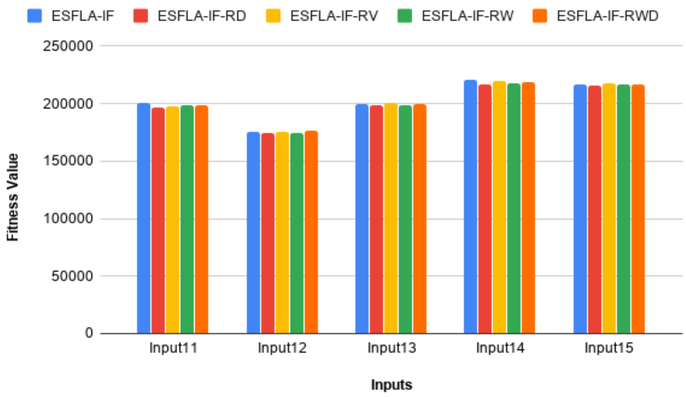
Fig.15. Comparison of convergence rate (a) Input 5 (b) Input 10 (c) Input 15



(a)



(b)



(c)

Fig.16. Comparison of partitioning schemes (a) small instances (b) medium instances (c) large instances

5.4. Results of Explored Partitioning Schemes

To assess the influence of diverse partitioning strategies, a comparative analysis was conducted by juxtaposing the best fitness values acquired from the algorithms with the fundamental partitioning scheme found in ESFLA-IF. The comparison encompasses ESFLA-IF, ESFLA-IF-RD, ESFLA-IF-RV, ESFLA-IF-RW, and ESFLA-IF-RWD. The results of this comparison are depicted in figure 16. The findings portray that the choice of partitioning scheme has a limited and non-significant impact on the performance of the SFLA algorithm. It appears that the algorithm possesses robustness, exhibiting commendable performance across various partitioning schemes. This resilience indicates that the algorithm effectively adapts and performs proficiently regardless of the specific partitioning approach employed.

6. Conclusions

This research delved into the optimization of Wireless Sensor Networks (WSNs) through the innovative integration of the Enhanced Shuffled Frog Leaping Algorithm (ESFLA) with multi-solution evolution mechanisms. Through a meticulous exploration of various partitioning schemes, fitness functions, and algorithmic enhancements, this study aimed to enhance the efficiency, robustness, and lifetime of WSNs.

The Enhanced Shuffled Frog Leaping Algorithm with Improved Fitness (ESFLA-IF) significantly outperforms the Improved Shuffled Frog Leaping Algorithm (ISFLA) in optimizing gateway load balancing in Wireless Sensor Networks (WSNs). ESFLA-IF achieves better residual energy distribution, with a narrower spread in energy levels indicating a more balanced workload among gateways. Specifically, ESFLA-IF reduces energy consumption by an average of approximately 2.15% across 15 input datasets. Additionally, ESFLA-IF extends network lifetime substantially, with improvements ranging from 2.60% to 45.60%, depending on the input scenario. Notably, in scenarios 11 to 15, network lifetime is enhanced by 27.54% to 45.60%. These improvements underscore the algorithm's effectiveness in ensuring balanced load distribution and prolonged gateway operation. The integration of the multi-solution evolution phenomenon in ESFLA-MSU further enhances performance metrics. Compared to ESFLA-IF, ESFLA-MSU shows superior best fitness values, elevated mean fitness values, and a reduction in standard deviation, indicating higher solution quality and diversity maintenance.

In a broader perspective, this research contributes to the advancement of optimization techniques in WSNs. By addressing challenges such as load balancing, energy efficiency, and network lifetime, the proposed methodologies hold promise for real-world applications in IoT and beyond. This study underscores the potential of algorithmic enhancements to revolutionize the efficiency and sustainability of network systems, with implications spanning various domains.

References

[1] Z. Wang, H. Ding, B. Li, L. Bao and Z. Yang. An energy efficient routing protocol based on improved artificial bee colony algorithm for wireless sensor networks. *ieee access*, Vol. *8*, pp. 133577-133596, 2020.

[2] G. Anastasi, M. Conti, M. Francesco Di, and A. Passarella. Energy conservation in wireless sensor networks: A survey. *Ad hoc networks*, Vol. *7*(3), pp. 537-568, 2009.

[3] Y. Chong, & S. P. Kumar, Sensor networks: evolution, opportunities, and challenges. *Proceedings of the IEEE*, Vol. *91*(8), pp. 1247-1256, 2003.

[4] Y. K. Yousif, R. Badlishah, N. Yaakob and A. Amir. An energy efficient and load balancing clustering scheme for wireless sensor network (WSN) based on distributed approach. IOP Publishing In *Journal of Physics: Conference Series* Vol. 1019, p. 012007, 2018.

[5] S. Mor and M. V. Saroha. Load balancing in wireless sensor networks. *Int. J. Softw. Web Sci*, Vol. *4*(2), pp.116-119, 2013.

[6] M. M. Eusuff, K. Lansey & F. Pasha. Shuffled frog-leaping algorithm: a memetic meta-heuristic for discrete optimization. *Engineering optimization*, Vol. *38*(2), pp. 129-154, 2006.

[7] G. Y. Zhu and W. B. Zhang. An improved shuffled frog-leaping algorithm to optimize component pick-and-place sequencing optimization problem. *Expert Systems with Applications*, Vol. *41*(15), pp. 6818-6829, 2014.

[8] K. K. Bhattacharjee and S. P. Sarmah, Shuffled frog leaping algorithm and its application to 0/1 knapsack problem. *Applied soft computing*, Vol. *19*, pp. 252-263, 2014.

[9] X. H. Luo, Y. Yang & X. Li. Solving TSP with shuffled frog-leaping algorithm. In 2008 eighth international conference on intelligent systems design and applications Vol. 3, pp. 228-232, 2008.

[10] Fang & L. Wang. An effective shuffled frog-leaping algorithm for resource-constrained project scheduling problem. *Computers & Operations Research*, Vol. *39*(5), pp. 890-901, 2012.

[11] D. Lei & X. Guo. A shuffled frog-leaping algorithm for hybrid flow shop scheduling with two agents. Expert Systems with Applications, Vol. 42(23), pp. 9333-9339, 2015.

[12] J. Cai, R. Zhou, & D. Lei, Dynamic shuffled frog-leaping algorithm for distributed hybrid flow shop scheduling with multiprocessor tasks. *Engineering Applications of Artificial Intelligence*, Vol. *90*, pp. 103540, 2020.

[13] P. Kaur & S. Mehta. Resource provisioning and work flow scheduling in clouds using augmented Shuffled Frog Leaping Algorithm. *Journal of Parallel and Distributed Computing*, Vol. *101*, pp. 41-50, 2017.

[14] J. Tang, R. Zhang, P. Wang, Z. Zhao, L. Fan and X. Liu. A discrete shuffled frog-leaping algorithm to identify influential nodes for influence maximization in social networks. *Knowledge-Based Systems*, Vol. *187*, pp. 104833, 2020.

[15] J. Luo, M. R. Chen. Improved shuffled frog leaping algorithm and its multi-phase model for multi-depot vehicle routing problem. Expert Systems with Applications, Vol. 41(5), pp. 2535-2545, 2014.

[16] P. Roy, P. Roy and A. Chakrabarti. Modified shuffled frog leaping algorithm with genetic algorithm crossover for solving economic load dispatch problem with valve-point effect. Applied Soft Computing, Vol. 13(11), pp. 4244-4252, 2013.

[17] S. Sharma, T. K. Sharma, M. Pant, J. Rajpurohit, and B. Naruka. Centroid mutation embedded shuffled frog-leaping algorithm. *Procedia Computer Science*, Vol. *46*, pp. 127-134, 2015.

[18] P. Sharma, N. Sharma and H. Sharma. Binomial crossover embedded shuffled frog leaping algorithm. In *2016 International Conference on Computing, Communication and Automation (ICCCA)* IEEE, pp. 321-326, 2016.
[19] P. Sharma, N. Sharma and H. Sharma. Elitism based shuffled frog leaping algorithm. In *2016 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* IEEE, pp. 788-794, 2016
[20] H. Wang, X. Zhen and X. Tu. SFDE: Shuffled frog-leaping differential evolution and its application on cognitive radio throughput. *Wireless Communications and Mobile Computing*, *2019*.
[21] R. Edla, A. Lipare, R. Cheruku, & V. Kuppili. An efficient load balancing of gateways using improved shuffled frog leaping algorithm and novel fitness function for WSNs. *IEEE Sensors Journal*, Vol. *17*(20), pp. 6724-6733, 2017.
[22] J. Zhang and T. Yang. Clustering model based on node local density load balancing of wireless sensor network. In *2013 Fourth International Conference on Emerging Intelligent Data and Web Technologies* IEEE, pp. 273-276, 2013.
[23] V. S. Gattani & S. H. Jafri. Data collection using score based load balancing algorithm in wireless sensor networks. In *2016 International Conference on Computing Technologies and Intelligent Data Engineering (ICCTIDE' Vol. 16,* pp. 1-3, 2016.
[24] S. Hussain, A. W. Matin, & O. Islam. Genetic algorithm for hierarchical wireless sensor networks. *J. Networks*, Vol. *2*(5), pp. 87-97, 2007.
[25] P. Kuila, S. K. Gupta & P. K. Jana. A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. Swarm and Evolutionary Computation, Vol. 12, pp. 48-56, 2013.
[26] G. P. Gupta & S. Jha. Integrated clustering and routing protocol for wireless sensor networks using Cuckoo and Harmony Search based metaheuristic techniques. *Engineering Applications of Artificial Intelligence*, Vol. *68*, pp. 101-109, 2018.
[27] R. Edla, M. C. Kongara & R. Cheruku. SCE-PSO based clustering approach for load balancing of gateways in wireless sensor networks. *Wireless Networks*, Vol. *25*(3), pp. 1067-1081, 2019.
[28] N. V. Archana. Efficient Load Balancing and Extended Network Lifetime for Cluster Based Routing Wireless Sensor Networks Using Fitness Function Algorithm. In *2021 IEEE Mysore Sub Section International Conference (MysuruCon)* pp. 653-659, IEEE2021.
[29] Q. Huamei, L. Chubin, G. Yijiahe, X. Wangping & J. Ying. An energy-efficient non-uniform clustering routing protocol based on improved shuffled frog leaping algorithm for wireless sensor networks. *IET Communications*, Vol. *15*(3), pp. 374-383, 2021.
[30] A. Lipare, D. R. Edla & R. Dharavath. Fuzzy rule generation using modified PSO for clustering in wireless sensor networks. IEEE Transactions on Green Communications and Networking, Vol. 5(2), pp. 846-857, 2021.
[31] Adamuthe & A. Pathan. Enhanced shuffled frog leaping algorithm with improved local exploration and energy-biased load reduction phase for load balancing of gateways in WSNs. *Open Computer Science*, Vol. *11*(1), pp. 437-460, 2021.
[32] Z. Wang, H. Ding, B. Li, L. Bao, Z. Yang and Q. Liu. Energy efficient cluster based routing protocol for WSN using firefly algorithm and ant colony optimization. *Wireless Personal Communications*, Vol. *125*(3), pp. 2167-2200, 2022.
[33] Benelhouri, H. Idrissi-Saba, and J. Antari. An improved gateway-based energy-aware multi-hop routing protocol for enhancing lifetime and throughput in heterogeneous WSNs. *Simulation Modelling Practice and Theory*, Vol. *116*, pp. 102471, 2022.
[34] P. Moinyak, K. Singh, S. Vats. Robust Load Balancing Scheme in WSNs. In 2023 International Conference on Computational Intelligence, Communication Technology and Networking (CICTN) 2023 Apr 20 (pp. 734-738). IEEE.
[35] S. Chaurasia, K. Kumar, & N. Kumar, Mocraw: A meta-heuristic optimized cluster head selection based routing algorithm for wsns. *Ad Hoc Networks*, Vol. *141*, pp. 103079, 2023.
[36] H. K. Vijayvergia and U. S. Modani. Energy Efficient Load Balancing and Routing using Multi-Objective Based Algorithm in WSN. *Intelligent Automation and Soft Computing*, Vol. *35*(3), pp. 3227-3239, 2023.
[37] W. B. Heinzelman, A. P. Chandrakasan & H. Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, Vol. *1*(4), pp. 660-670, 2002.
[38] M. M. Eusuff & K. E. Lansey. Optimization of water distribution network design using the shuffled frog leaping algorithm. *Journal of Water Resources planning and management*, Vol. *129*(3), pp. 210-225, 2003.

Authors' Profiles

Abdulhameed R. Pathan is a Technical Consultant at Adobe, leveraging his expertise to drive technological advancements. He earned his Master of Technology degree from Rajarambapu Institute of Technology in Rajaramnagar, Sakharale, Sangli. With over 8 years of extensive experience in the industry, Abdul possesses a robust technical skill set and a profound understanding of cutting-edge technologies. His proficiency spans various domains within the realm of information technology, contributing significantly to the development and implementation of innovative solutions.

Amol C. Adamuthe is a Professor and Head of Information Technology at Rajarambapu Institute of Technology Rajaramnagar, Sakharale, Sangli, MS, India. He received his Ph.D. from Mumbai University and Master of Technology in Computer Engineering from Dr. B. A. Technological University, Lonere in year 2017 and 2008 respectively. With a substantial academic and research background, he has authored over 65 papers at international and national levels. He honored with the Institution of Engineers (India) Young Engineers Award 2018-19 in the Computer Engineering discipline. His area of research includes technology forecasting, optimization algorithms, soft computing, and cloud computing.

How to cite this paper: Abdulhameed Pathan, Amol C. Adamuthe, "Enhanced Wireless Sensor Network Lifetime using Modified SFLA with Improved Fitness Function", International Journal of Computer Network and Information Security(IJCNIS), Vol.17, No.5, pp.91-113, 2025. DOI:10.5815/ijcnis.2025.05.07