ISSN: 2229-6948(ONLINE)
DOI: 10.21917/ijct.2025.0541

A HYBRID CSBHC-BASED METAHEURISTIC FRAMEWORK FOR ENERGY-BALANCED WIRELESS SENSOR NETWORKS

N.S. Kavitha[1], R. Rathiya[2] and M. Sakthivel[3]

[1,2]Department of Information Technology, Dr. N.G.P. Institute of Technology, India
[3]Department of Computer Science and Engineering, Erode Sengunthar Engineering College, India

Abstract

In many sectors like healthcare, military, automotive sector, and manufacturing, the wireless sensor networks (WSN) have been widely used. Regardless of its widespread applications, WSN also have some limitations. Those limitations include processing power, storage capacity and energy supply (ES). Here, the ES is one of the major challenges in WSN. To address this issue, the WSN aims to enhance the energy efficiency (EE). Then, the data aggregation (DA)-based clustering technique is suggested for resolving those challenges, as it balances energy consumption (EC) across sensor nodes (SN). This will facilitate the suggested method in improving EE. For the purpose of selecting cluster heads (CH) effectively, a robust search algorithm and faster convergence are crucial. An adaptive metaheuristic (MH) algorithm (AMHA) based on Tunicate Swarm Optimisation Algorithm (TSOA) is suggested, and it may support in optimizing deep foundation design and global optimization. In every iteration, 2 crucial phases are included in the suggested Adaptive TSOA (ATSOA). Those steps include a local refinement based on the top-performing tunicate (TC) and a global search (GS) directed by randomly chosen TC. Thus, premature convergence is prevented by these changes, and it also supports in enhancing the exploration capabilities of the model. To enhance the convergence speed and optimise search accuracy (ACC), a new hybrid method (CSBHC) was suggested. The Cuckoo Search (CS) and (BHC) β-Hill Climbing are integrated in this CSBHC method. The benefits of the CS algorithm (CSA) with the BHC method are integrated in the CSBHC method, as similar to probability mechanism in (SA) Simulated Annealing. On the basis of an exponentially decreasing probability, it becomes active at every repetition. The search efficiency is greatly enhanced by the suggested method, and it was demonstrated by the comparative tests with different node density (ND). Thus, the routing performance and effective CH selection (CHS) are improved by the suggested method.

Keywords:

Wireless Sensor Networks (WSNs), Tunicate Swarm Optimization (TSA), Adaptive Tunicate Swarm Optimization (ATSA), Cuckoo Search with β-Hill Climbing (CSBHC)

1. INTRODUCTION

Precision agriculture (PA), industrial automation, military surveillance, smart healthcare, and environmental monitoring are just a few of the contemporary applications that are increasingly depending on WSN [1,2]. WSNs are composed of multiple low-power SN. WSNs are designed to work together to detect, analyze, and send environmental data to a sink (SnK) or central (BS) base station [3,4]. Nevertheless, these SN are usually deployed in inaccessible or isolated areas and run on batteries. SN makes it impossible to recharge or replace batteries [5]. Therefore, maintaining EE and extending network lifetime (NL) continue to be major challenges in the design and operation of WSNs.

The application of clustering algorithms is one popular method for addressing EC in WSNs [6]. This method groups nodes into clusters, with a selected node called the CH. Data from its cluster members (CM) must be aggregated and sent to the BS by this CH [7, 8]. Clustering greatly lowers redundant data transmission (DT) and saves energy, but the best selection of CH is crucial to network efficiency [9]. Poor CHS can lead to uneven energy depletion, frequent re-clustering, and network partitioning, all of which degrade performance. Hence, intelligent and dynamic CHS mechanisms are essential for sustainable WSN operation [10].

Because MHOA may explore huge solution spaces without being stuck in local optima, MH optimisation algorithms (MHOA) have shown tremendous promise in addressing CHS issues [11]. CS, Ant Colony Optimisation (ACO), and Particle Swarm Optimisation (PSO) are some of the widely used nature-inspired algorithms (NIA) [12,13]. Slow convergence rates, sensitivity to initial parameters, and premature convergence (PC) are some of the issues that these methods mostly face [14,15]. The robustness of multiple procedures is integrated in the hybrid MH approaches (MHA) for the purpose of resolving those issues. This application may offer an effective method, and it will deliver better convergence speed, accuracy, and robustness.

In this context, the proposed framework introduces a novel hybrid MHA named CS with β-(HC) Hill Climbing (CSBHC), designed specifically for EE clustering in WSNs. The CSBHC algorithm combines the global search (GS) capability of CS with the local refinement strength of BHC. The BHC phase is probabilistically triggered during iterations, governed by an exponentially decreasing function, which allows the algorithm to maintain exploration in early stages and intensify exploitation in later stages. This balance effectively mitigates PC and enhances the discovery of optimal or near-optimal CH configurations.

A DA-based clustering method incorporates the suggested CSBHC-based framework for ensuring the energy balance (EB) across the network. The EC, NL, DT reliability and clustering ability are all improved by the suggested method, and it was demonstrated by extensive simulations over several ND and deployment scenarios. The energy-aware (EA) WSN design is greatly advanced via the hybrid CSBHC framework, as it selects CH and optimizes (RP) routing paths. A practical applicability in mission-critical and resource-constrained environments is also facilitated by the suggested method.

2. LITERATURE REVIEW

For circular WSN, a unique EB (EB) Unequal Clustering Approach (EBCA) was suggested by Zhao et al. [16]. Its performance was then assessed through extensive simulations. Across a variety of gradients, the experimental results demonstrated that EBCA could successfully balance EC of the CH. Both the classic and state-of-the-art (SOTA) clustering algorithms were compared with EBCA, which significantly lengthened the NL and decreased overall EC.

Jasim et al. suggested an EE unequal clustering (UC) method based on a EB method (EEUCB) [17]. To reduce energy waste, this EEUCB used minimum and maximum distance criteria. Furthermore, the suggested EEUCB employed a double CH approach along with a sleep-awake mechanism, utilising the node's maximum energy capacity. The average energy threshold (AET), average distance (T) threshold, and BS layering node were also considered in the clustering rotation (CR) approach that EEUCB offered. It was based on 2 sub-stages, specifically inter-clustering and intra-clustering (IC) procedures. A number of current methods were then used for executing comparative analysis, it then supports for determining the recommended method's potential. The outcomes demonstrated that the EEUCB protocol yielded lifespan benefits of 57.75%, 19.63%, 14.7%, and 13.06% in comparison to LEACH, factor-based LEACH (FLEACH), EEFUC, and UDCH.

With an emphasis on improving NL and decreasing delay (D), a sequential quadratic programming (SQP)-based multi-path routing (MPR) design named EBDA-DEFL, was suggested by Maratha et al. [18]. An optimisation tool was first used to solve this SQP-based formulation, and then PSO was used to solve it. To mitigate the adverse effects of MPR, a traffic load distribution (TLD) quota approach was also implemented. To assess its superiority over earlier techniques, the suggested strategy was put into practice and contrasted with current algorithms. First node death (NoD), half NoD, last NoD, latency, and the amount of time it took for Fminimax and PSO to compute were among the metrics used in the comparison. The suggested method's superiority over the current approaches was validated by the simulation results.

For EE and balanced (DC) data collection in WSNs/IoT, Navarro et al. [19] presented a unique routing protocol (RP). This technique uses the parent set concept to use suboptimal network routing options, making it adaptable to any cost-based routing solution. To improve NL in WSNs, the suggested method added a random element to the packet forwarding process and still utilizes the reliable routing topologies created by cost-based RP. The method's implementation was assessed using comprehensive actual testbed trials and tests against a number of SOTA WSN RP. While retaining over 99% reliability, the results showed a significant decrease in EC at the routing layer for the busiest nodes, with savings ranging from 11% to 59%. Furthermore, a field deployment for environmental monitoring in a forest area was conducted in a heterogeneous WSN. The efficacy of the method was demonstrated by the thorough reporting of the experimental results. For community evaluation and adoption, the suggested EE RP, CTP+EER, has been made available as open-source.

The efficiency, balance, and fairness of EC are all taken into consideration by the fair and EB ferry fleet placement (FEEB) strategy, and this FEEB method was suggested by Hu et al [20]. There are two mutual phases to this concept. The powered-Voronoi diagram is used in the first step to define service regions in a network. In the second step, the shortest-path (SP) itinerary across sensors in every service region is determined using a genetic-based variant of the travelling salesman problem (TSP) approach. This plan can therefore ensure an equitable distribution of the workload. This suggested strategy prolongs the lifespan of several ferries (F) teamwork and preserves EC balance. Comparative analysis was conducted by comparing the suggested method with several traditional schemes, including Native, K-Means, and Spiral. From the outcomes of the simulation, it is clear that the suggested method executes better than the conventional methods in terms of cumulative EC, residual energy (RE) distribution, Jain's fairness index on EC, the number of ferries that are still in operation, and the total of task execution times (TET) during F teamwork.

Fan and Xin [21] introduced the EB path tree-based clustering and routing algorithm (EBPT-CRA) for large-scale WSN. To calculate the CH competition coefficients of the nodes, the aggregation betweenness of the nodes was first calculated after the EBPT was formed. The clustering and IC routing were completed by selecting the CH and creating the cluster trees through an iterative manner. At last, the RE of nodes and their communication energy were used to build inter-cluster routing. The EBPT-CRA enhanced NL, T, and service capabilities in large-scale WSNs, based on simulation studies. This EBPT-CRA efficiently conserving and balancing node energy.

In order to create minimum spanning trees (MST), an innovative technique that made use of Prim's algorithm (PA) was suggested by Saad et al. [22]. Improving EB in WSN was the goal of this MST. To find the best connections between network nodes and reduce EC, Prim's algorithm was successfully used. Network initialisation, EC modelling, MST creation using Prim's algorithm, and mobile SnK node movement optimisation were all important components of the suggested methodology. Numerous tests on various datasets showed that the method, which had high sensitivity (S) and moderate complexity, greatly enhanced EB. In order to contribute to more sustainable and efficient network deployments, this study demonstrated how PA may improve EE and prolong the NL.

Both traditional nodes and energy-harvesting nodes are included in the hybrid network architecture that Bhasgi and Terdal [23]. A distributed RP that took failure tolerance (FT), load balancing (LB), and EC into consideration was created. Energy density, inter-node distance, and RE were taken into consideration when choosing relay nodes (RN). The network was dynamically rerouted in case a node failed along the routing path. The suggested algorithm, DEBH, was assessed in comparison to current techniques. The outcomes showed that DEBH outperformed earlier strategies in terms of efficiency.

An innovative methodology called Improved LB Clustering for Energy-Aware (EA) Routing (ILBC-EAR) was put forth by Loganathan et al. [24]. The main purpose of this ILBC-EAR's method was to achieve optimal EC between CH and member nodes (MN) via the LB procedure. The framed clusters' size were measured to provide consistent EC across nodes. Furthermore, the model transmitted sensed data to the SnK or BS via a Finest Routing Scheme based on LB Clustering. According to evaluation results, the suggested EA model maintained a balanced EC rate among CH and produced a longer NL than current techniques. Additionally, the model sent data packets with a lower D and a higher T.

To balance EC among SN and improve the capacity to attain optimal global results, the EB ant-based routing algorithm has been suggested by Wang et al. [25]. In order to confine the search path and preserve node energy, a search angle was first added to the basic ant colony algorithm (ACA) in order to update the pheromone heuristic function (PHF). Furthermore, pheromone concentration upper and lower bounds were determined. The RE threshold of the nodes were then modified to be flexible. In order to improve global optimisation (GO) capabilities and further lower EC, the pheromone increment formula was optimised. According to simulation results, an enhanced approach lowered the optimal path length by 1.47% and 1.59%, respectively, and decreased the average node EC by 15.12% and 11.68% when used in two scenarios. The NL was significantly extended as a result of these noteworthy results, which showed that an improved algorithm significantly improved GO and balanced EC throughout SN.

3. NETWORK MODEL

With a transmitter and a receiver separated apart by a distance d, the free space (FS) network model was employed in the study. Bits are utilised for transmitting the data that has to be sent between the SN. The non-uniform EC of the quasi-stationary (QS) SN is dependent solely on the distance between the SN and the BS, and they are distributed throughout a rectangular cross-section. The self-organising SN with set power levels is thought to be homogeneous and does not know where it is. Eq.(1) and Eq.(2) provide the following formulas for the transmitter and receiver energy levels.

$$E_{Tx}(l,d)=\begin{cases} lE_{\text{elec}}+l\varepsilon_{\text{fs}}d^2, & d\le d_0 \\ lE_{\text{elec}}+l\varepsilon_{\text{mp}}d^4, & d>d_0 \end{cases} \tag{1}$$

$$E_{Rx}=E_{\text{elec}} \tag{2}$$

where, the EC to transmit a single bit of data is denoted by E_{elec}. For the FS model, ε_{fs} is the transmitter side coefficient of amplification. Here, the count of data sent is one and the distance of the transmission is < the distance of T. If the transmission distance exceeds the threshold distance, ε_{mp} is used. Here, one piece of information is sent.

The $SN(n)$ are randomly placed in a rectangle area of $M\times N_0$ m^2 using ad hoc settings as the application situations in this study. The primary goal of the study is considered to be an optimal cluster's secretion. The following fitness function (FF) is used to carry out the optimal cluster secretion. The following Eq.(3)-Eq.(5) provides it:

$$f=\varepsilon f_1+(1-\varepsilon)f_2 \tag{3}$$

$$f_1=\max_k\left\{\frac{\sum_{\forall\,\text{node}(i)\in C(k)} d\left(\text{node}(i),CH(k)\right)}{\lVert\text{cluster}(k)\rVert}\right\} \tag{4}$$

$$f_2=\frac{\sum_{i=1}^{k}E(\text{node}(i))}{\sum_{j=1}^{k}E(CH(j))} \tag{5}$$

where, the scaling factor is denoted by ε. Between 0 and 1, the value of ε falls. The number of SN that are part of the cluster(k) is known as $\|cluster(k)\|$. The maximum Euclidean distance (ED) that separates the SN from the CH is denoted by f_1. The f_2 is the ratio of the entire energy of the CH $E(CH(j))$ to the initial SN energy $E(node(i))$.

$$F_{\text{router}}=P_1\frac{1}{x_0}+P_2\frac{1}{x_1}+\cdots+P_n\frac{1}{x_{n-1}} \tag{6}$$

Eq.(6) provides the corresponding node weights for Eq.(3b) and Eq.(4), where, the objective function (OF) is polynomial.

$$F_{r(n)}=H_{r(n)}+\frac{C_{r(n)}}{C_{\max}}+\frac{R_{r(n)}}{(C_{\max})^2} \tag{7}$$

where, the fitness of the CH is denoted as F_{router}. The CH features are $P_1,\ldots, P_n$, and the CH weight is x.

The fitness of the nth CH is denoted as $F_{r(n)}$. The hop (H) distance among nth CH and BS is denoted by $H_{r(n)}$. The SN of the nth router is $C_{r(n)}$. The SN with the maximum energy is C_{max}. The RSSI value of the nth CH is $R_{r(n)}$.

Using BS broadcasting, variable x is controlled to optimise the SN, and H count is more significant than the total SN. In order to enable a more seamless assessment of BS offers and their parameters, it provides a larger hop with fewer SN, that is determined using smoother conditional statements.

4. PROPOSED METHODOLOGY

The suggested work enhances EE in WSNs by employing a DA-based clustering strategy that ensures balanced EC. An ATSA algorithm is developed, incorporating global exploration using random tunicates and local refinement via the best tunicate. To address premature convergence and further accelerate search, a hybrid algorithm named CSBHC is introduced. CSBHC dynamically integrates BHC based on an exponentially decreasing probability, improving the accuracy of CHS. This dual-optimization approach enhances convergence speed and routing performance under varying node densities.

4.1 TUNICATE SWARM ALGORITHM (TSA)

During navigation and foraging, jet propulsion and swarm behaviour of marine TC served as the inspiration for TSA, a simple MH optimiser. The size of this animal is millimetres. In the sea, TC can find food sources. The food source is not indicated in the provided search space (SS). When a tunicate uses jet propulsion, it needs to meet three fundamental requirements: 1. In the SS, the TC needs to stay apart from other TC. 2. To find the ideal search location, TC must take the correct path. 3. As much as possible, TC should look like the best search (AG) agent.

To find the optimal food supply (i.e., the top value of the OF), TSA's candidate solutions (i.e., TC) are searching. In each iteration of this procedure, the TC adjust their placements based on the best TC that is saved and enhanced. The TSA begins with a population of arbitrarily created TC based on the acceptable limits of the design variables (DV), as shown in Eq.(8)

$$\vec{T}_P=\vec{T}_P^{\min}+\text{rand}\times\left(\vec{T}_P^{\max}-\vec{T}_P^{\min}\right) \tag{8}$$

The position of every tunicate is denoted by $\vec{T}_P$. A random number between [0; 1] is called *rand*. The lower and higher boundaries of the DV are $\vec{T}_P^{\min}$ and $\vec{T}_P^{\max}$. During the iterations, the TC use the following formula using Eq.(9) to modify their location:

$$\vec{T}_P(x+1) = \frac{\vec{T}_P(x)+\vec{T}_P(x)}{2+C_1} \tag{9}$$

where, the random number C_1 falls in the interval [0,1]. Based on Eq.(10), $\vec{T}_P(x)$ represents the most recent location of the TC in relation to the location of the food source.

$$\vec{T}_P(x) = \begin{cases} SF + A\cdot| SF - \text{rand}\cdot\vec{T}_P|, & \text{rand} \geq 0.5 \\ SF - A\cdot| SF - \text{rand}\cdot\vec{T}_P|, & \text{rand} < 0.5 \end{cases} \tag{10}$$

where, the population's ideal *TC* position represents *SF*, the food source. The randomised vector denoted as *A* that keeps TC from running into each other. Eq.(11) is used to model this:

$$A = \frac{c_2 + c_3 - 2c_1}{VT_{\min} + c_1(VT_{\max} - VT_{\min})} \tag{11}$$

where, the random values c_1, c_2 and c_3 fall in the interval [0; 1]. The lowest and greatest speeds are indicated by VT_{min} and VT_{max}. These VT_{min} and VT_{max} are regarded as 1 and 4. These speeds are utilised to create social interaction.

The steps in the TSA algorithm are as follows:

Step 1: Based on Eq.(8), initialise the *TC* population *ETp*.

Step 2: The maximum number of iterations and the initial parameters should be chosen.

Step 3: Determine the value of _tness for each search *AG*.

Step 4: In the provided SS, an optimal TC is explored.

Step 5: Each location of the *IC* is updated by Eq.(9).

Step 6: In each *SS*, adjust the updated TC that crosses the boundary.

Step 7: Determine the most recent *TC* fitness value (FV). Update the best solution if a better one exists than the one that was previously considered optimal.

Step 8: The algorithm stops if the stopping criterion is met. If not, go through Steps 5-8 again.

4.2 ATSA

The TSA is prone to getting stuck in local optima (LO), even if it can generate remarkable outcomes in contrast to some standard methods. For extremely complicated issues involving multiple LO, TSA is not the best option. Each TC in TSA changes its location according on the location of the food source (FS) (i.e., the location of the best TC in the entire population), as illustrated in (9) and (10). However, if the algorithm converges too soon without knowing the location of the FS, there won't be any recovery. Then, the algorithm loses its exploratory potential and becomes inactive after it has converged. Consequently, this mechanism causes the TSA algorithm to become trapped at local minimum points. To address these issues and improve the algorithm's search capabilities and flexibility, an adaptive version of the TSA (ATSA) is suggested.

Exploration and exploitation (E-E) are the two stages of the search process that the effective MH algorithm (MHA) must separate. Looking into new locations in the entire SS that are distant from the current place is known as exploration. When an MHA tries to find the whole solution space and investigate the promising areas, it is called the exploration phase. Conversely, exploitation refers to an optimisation algorithm's ability to find solutions that are almost optimal. The optimiser can focus on the neighbourhood of higher-quality (HQ) solutions inside the SS during this phase. As mentioned above, the TSA procedure adjusts the location of candidate solutions (CS) around a single point that represents the best option for the entire population during each iteration run. It indicates that the TSA is capable of effective exploitation. But the algorithm's ineffective exploration capability and absence of an efficient global search (GS) remain its main drawbacks.

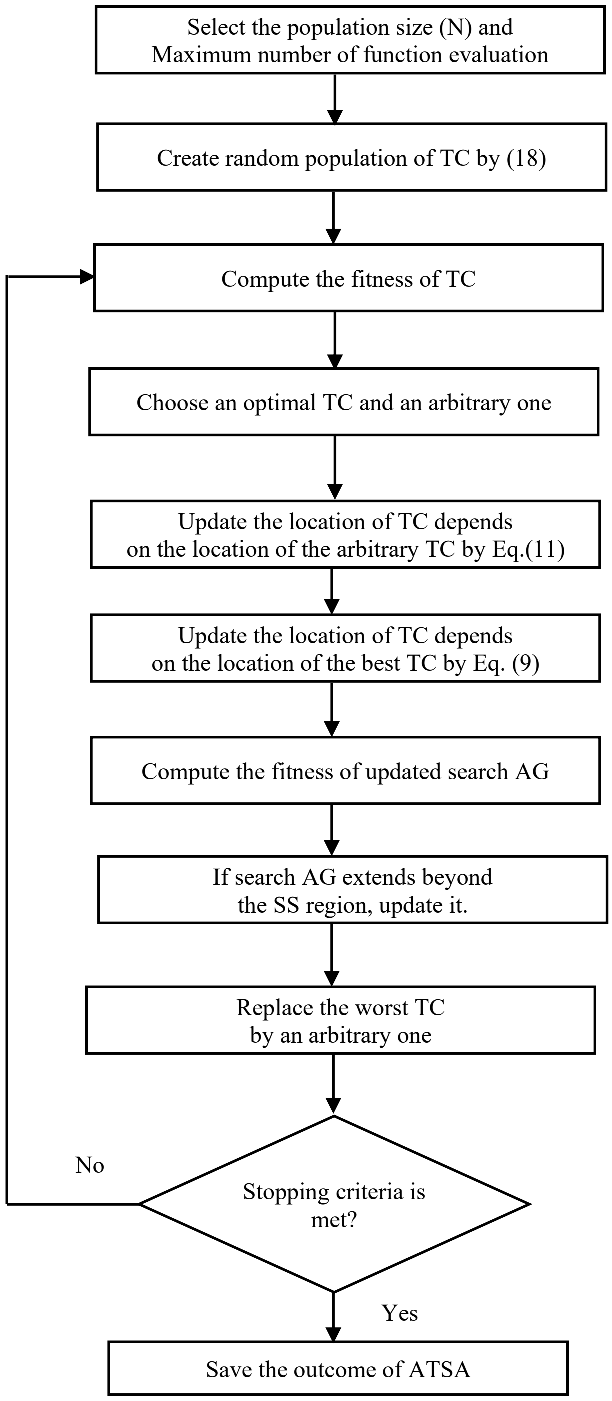

Fig.1. The flow diagram of ATSA

Every repetition of the suggested ATSA consists of 2 primary stages created to enhance the procedure's performance and exploration ability. During the first phase (exploration phase), a CS is randomly selected rather than the optimal solution. The locations of this random TC are then used to update the positions of the CS. Additionally, an optimiser should extensively investigate a variety of SS areas using its randomised operators in order to have effective exploration. For the purpose of creating solutions in several SS areas, the TC's updating equation in the suggested ATSA considers two distinct arbitrary values.

The following is a mathematical model of the ATSA's exploration phase:

$$\vec{T}_p(x+1) = \vec{T}_p(r) - rand_1 \cdot \vec{T}_p(r) - 2rand_2 \cdot \vec{T}_p(x)$$

where, TC is chosen at random from present population $\vec{T}_p(r)$. Random values between 0 and 1 are denoted by $rand_1$ and $rand_2$. The TSA algorithm can execute a more robust GS across the entire SS via this process, which also promotes exploration.

The position of the best TC discovered thus far is used by the TC to update their positions in the 2nd stage of the ATSA algorithm, known as the exploitation phase, determined by (9). For each iteration of the suggested ATSA, a randomly generated TC will be used to replace the worst TC with the highest OF value. The suggested ATSA algorithm's flowchart is shown in Figure 1.

4.3 COMPARATIVE (TCA) TIME COMPLEXITY ANALYSIS

The computational TCA can be used to assess a novel optimisation algorithm's overall performance from several perspectives. *O* is a mathematical notation used in computer sciences that shows how long an algorithm must run for by taking into account how quickly it grows when handling various inputs.

Three components are analysed in the TCA of the majority of algorithms. Analyses of these 3 components are also necessary for the TCA of the suggested ATSA:

1. $O(N\times D)$ is typically used to calculate the time complexity of population initialisation. where, the population size is indicated by N. The problem's dimensions are indicated by D.
2. $O(N\times F(\boldsymbol{X}))$ is typically used to estimate the time complexity of the initial fitness evaluation. The OF is represented by $F(\boldsymbol{X})$.
3. $O(Maxiterations \times(N\times D+N\times F(\boldsymbol{X})))$ is typically used to calculate the main loop's time complexity. where, *Maxiterations* denotes the maximum number of iterations.

Therefore, $O(Maxiterations \times(N\times D+N\times F(\boldsymbol{X})))$ is the ATSA algorithm's overall time complexity.

4.4 HYBRID ALGORITHM FOR PREMATURE CONVERGENCE

Theoretically and experimentally, the CS approach, like any optimisation algorithm, cannot be certain to stay out of suboptimal solutions. The hybrid CS strategy is typically used to reduce the chance of being stuck in suboptimal solutions and improve the accuracy (ACC) of the CS outcomes. But, compared to the original CS method, this approach is less appealing due to its high computing cost. In contrast, a popular variation of the HC procedure, the BHC procedure seeks to balance the E-E of the solutions in the solution space during the search phase. Some hybrid CS algorithms completely integrate the CS algorithm with different search methods, as was previously discussed. Heavy computation is required for this integration, especially when employing the searching methods at every CS iteration. The efficiency of CS is not appreciably improved by other hybrid CS algorithms that employ search strategies as selection methods.

Only when the objective value (ObV) of a cuckoo (novel solution) is higher than the ObV of a randomly chosen nest (stored solution) does the CSBHC employ the BHC method with an exponentially reducing likelihood (i.e., the acceptance likelihood of SA) in the current paper.

In order to decrease the computing time of CSBHC, this procedure seeks to decrease the rate of BHC usage in CSBHC. The ten primary stages of CSBHC are depicted as a flowchart in Figure 2 and it is explained below:

- Phase 1: The problem and CSBHC parameters are initialized. The optimisation problem is typically modelled as $\min\{f(x)|x\in X\}$ or $\max\{f(x)|x\in X\}$ in this step. The *OF* of the selected solution, which is made up of N DV $x=\{x_1,\ldots,x_N\}$, is denoted by $f(x)$. $x_j\in[LB,UB]$ is the DV's value. where, the lower bound of the search range (SR) is denoted by LB. The upper bound of the SR is denoted by UB.
- Phase 2: The population of M nests (solutions) are initialized. $X=\{x_1,\ldots,x_M\}$ is an augmented matrix of size $M\times N$ that represents the population of M nests. The random function $x_j=LB_j+(UB_j-LB_j)\times U(0,1)$, $\forall j=1,2,\ldots,N$, is used to create the values of the DV of $x=\{x_1,\ldots,x_N\}$. Random values between 0 and 1 are generated using the uniform random function U (0,1).

$$X = \begin{bmatrix} X_1^1 & X_2^1 \\ X_1^2 & X_2^2 \\ \vdots & \vdots \\ X_1^M & X_2^M \end{bmatrix} \cdots \begin{bmatrix} X_N^1 & X_N^2 \\ \vdots & \vdots \\ X_N^M & \cdots \end{bmatrix}$$

- Phase 3: Make use of Lévy flights (LF) to get a cuckoo. The OF $f(x_i)$ is used to assess the new solution (cuckoo, as x_i), which is created at random using LF in this phase.
- Phase 4: From the population, a nest is selected at random. The OF $f(x_i)$ is used to evaluate a stored solution (nest, as x_j) that is chosen at random from the M nests.
- Phase 5: Examine and contrast objective values of the x_i and x_j. For minimisation problems, if $f(x_i)\leq f(x_j)$, swap x_j for x_i and go to step 8. If not, proceed to step 6.
- Phase 6: Make an exponentially decreasing probability call to the BHC function. Proceed to Step 7, where $\Delta f=f(x_j)-f(x_i)$, and T is a decreasing temperature, with probability $e^{\Delta f/T}$, if $f(x_i)>f(x_j)$. If not, proceed to Step 8.
- Phase 7: Utilise the BHC function to enhance the optimal solution. The best solution discovered by CS is enhanced using the BHC function, which is displayed in Fig. 5. The initial solution in Figure 2. is the best one that CS could find. That's the only difference.
- Phase 8: Update the CSBHC population. New solutions are substituted for a fraction (pa) of the worst ones.
- Phase 9: Return the best solution after ranking the others. After ranking the solutions according to the OF, the optimal solution is determined.

- Phase 10: Verify the stopping condition. As long as the stopping condition is not met, phases three to ten are repeated.

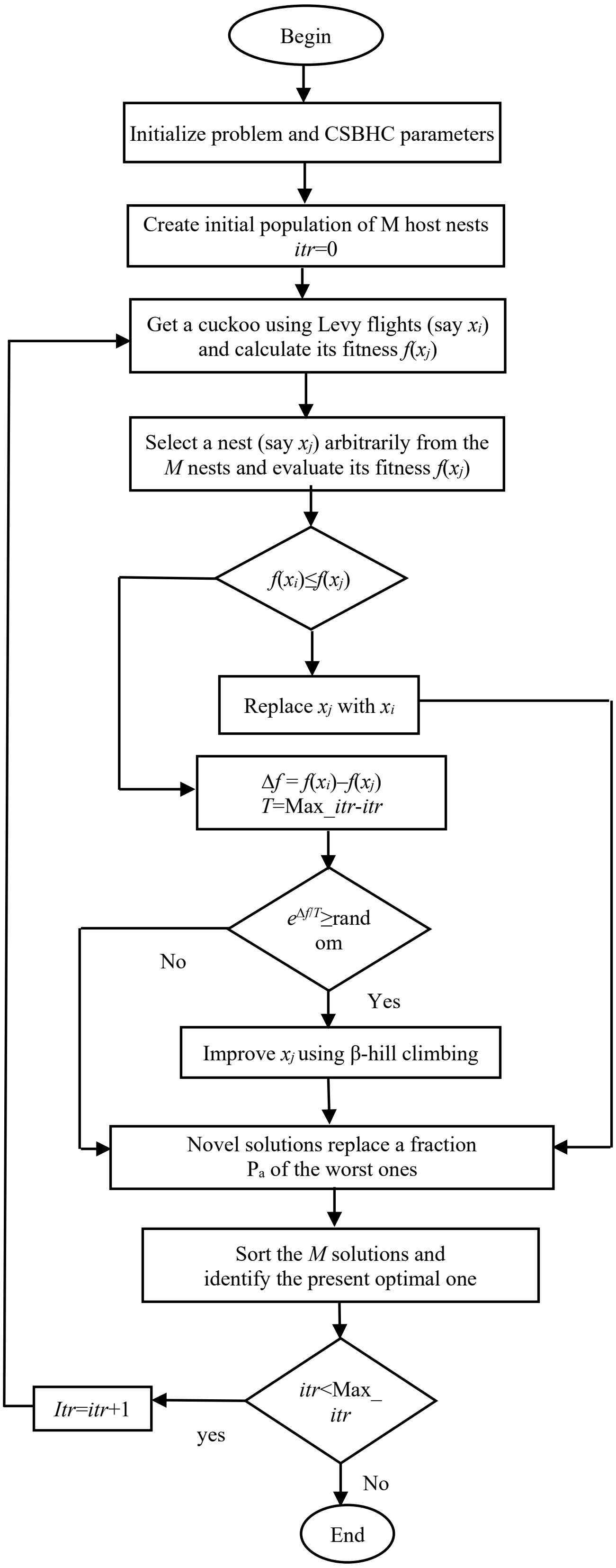

Fig.2. The flow chart of CSBHC

Algorithm 1: The Hybrid CS and BHC Algorithm

1: Begin

2: OF $f(x_i)$, where $x_i=\{x_1,..,x_N\}$ is a nominated solution

$x_j = LB_j+(UB_j-LB_j)\times U(0,1), \forall j=1,2,\ldots N.$

3: Create initial population X=$\{x_1,....x_M\}$ of M solutions

4: itr=0

5: while (itr<Max_iterations) or (stop criterion) do

6: Choose a solution i randomly from the present population and replace its solution x_i by LF

7: Compute the quality/fitness value $f(x_i)$ of x_i

8: Choose a solution randomly from the present population (j))

9: Compute the difference among the fitness of x_j and the fitness of x_i: $\Delta f=f(x_j)-f(x_i)$

10: Calculate the current temperature $T\leftarrow$schedule [itr]

11: if $f(x_i)$ is better than $f(x_j)$ then

12: Replace x_j by x_i

13: else if $e^{\Delta f/T}\geq$ rand then

14: $x_{j\text{-}new}\leftarrow$β-hill-climbing ($x_{j\text{-}old}$,$LB$,$UB$,$problem$);

15: end if {rand ∈[0,1]}

16: A fraction (p_α) of worst solutions are substituted with novel ones

17: Keep the best solutions (i.e., solutions with quality solutions)

18: Rank the solutions and determine the current best x_{itr}*

19: itr=itr+1

20: end while

21: Post-process results and visualization

22: End

5. RESULT AND DISCUSSION

On a high-end computing system with an Intel Core i5 processor running at 2.30 GHz and 8 GB of primary memory, the suggested approach is coded using Matlab software. In terms of the average of (i) end-to-end delay (E2ED), (ii) packet delivery ratio (PDR), (iv) average throughput (AT), and (v) average RE (ARE), the suggested approach is evaluated against three further RP. For ease of implementation, the suggested system makes use of the same energy model and routing parameters as WSN.

The Table.1 lists the network parameters (NP) needed to simulate the suggested task. ε_{fs}, ε_{mp}, and E_{elec} are some of the EC parameters that are covered in section 3. The size of control packets determines the length of the notification message. The size of the message packet is used to establish the data message length.

A rectangular cross-section of 1,000 SN is used to deploy the full sensor area. To assess the rate of full convergence, the SnK induction is maintained at a random location. Since the CH is in charge of the entire operation and maintains a high (IE) initial energy, the IE of SN and CH is kept at 0.1 and 0.3 J, respectively.

5.1 PERFORMANCE METRICS (PM)

E2ED, AT, PDR, ARE, and EC are among the PM that are employed for assessing the suggested method’s efficiency.

Table.1. NP setting for testing

NP	Numbers
ε_{fs}	10(pJ/bit (b)/m^2)
ε_{mp}	0.0013 (pJ/b/m^4)
E_{elec}	50(nJ/b)
Dimensions of control in (*P*) packets	200(b)
Dimensions of messages in the *P*	4000(b)
SnK node location	Arbitrary
RP	Zone RP in balanced (B) and unbalanced network settings
Area	400*800 (m^2) for B 800*800
Overall count of SN	1000
d_0	0.5L
IE of SN	0.1J
IE of CH	0.3J
CH rate	20%

5.1.1 PDR:

The proportion of all P transmitted by the source node to all P delivered to the destination node (DN) is known as PDR. When it comes to efficiently sending source P to the DN, the PDR offers the advantages of the RP. There is a greater ratio than the protocol's performance.

$$\text{PDR} = \frac{P_r}{P_t} \times 100 \tag{13}$$

The total count of *P* received is represented by P_r. The total number of P sent is represented by P_t.

5.1.2 Throughput or AT:

The total count of bits received at the DN is known as throughput, or AT. The throughput metric, that is based on the *Pr* at the DN. This throughput metric is used to assess the RP's efficiency.

$$\text{Throughput} = \frac{8B_r \times 10^3}{T} \tag{14}$$

The total count of bytes received is represented by *Br*. The total simulation time is denoted by *T*.

5.1.3 Average E2ED (AE2ED):

The average period required for a P to transmit from its source and intermediate nodes to its DN via a sensor network is known as the average E2ED. The total of the propagation time, transfer rate, retransmission rate, and buffering rate at route discovery (RD) is used to calculate the average E2ED.

$$\text{Delay} = \frac{\sum_{i=1}^{n}\left(R(i) - S(i)\right)}{n} \tag{15}$$

where, *P* sent from the source node at time *i* are represented by *S*(*i*). The number of P received from the source node at time *i* is *R*(*i*). The total count of SN moving over the network at i^{th} time is denoted by *n*.

5.1.4 EC:

The total EC by the SN in time i is known as EC. Based on the factors of the RE of each node, the determination is based on calculating the energy level of each SN at the end of the iteration.

$$AEC = \sum_{i=1}^{n}\left(E_{\text{ini}}(i) - E(i)\right) \tag{16}$$

where, the IE of the SN at time *i* is represented by $E_{ini}(i)$. The total energy of a SN at the time *i* is denoted as *E*(*i*).

5.1.5 NL:

The time it takes for the mobile node to run out of battery power for data collection and DT is known as NL.

$$NL = \sum_{i=1}^{n}\mathbf{1}_{E(i)=0} \tag{17}$$

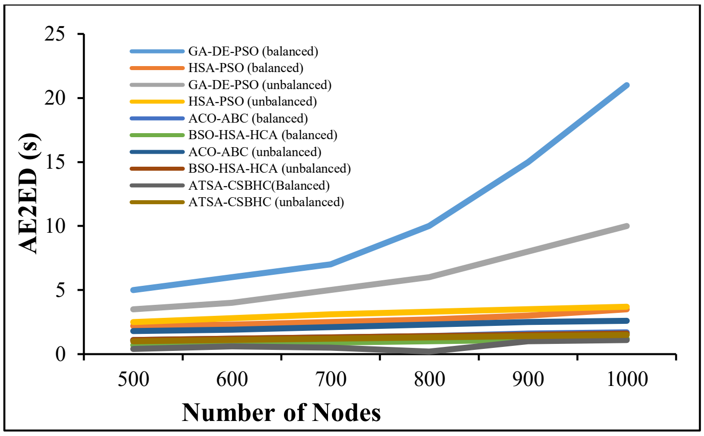

Fig.3. Comparison of AE2ED for Balanced and Unbalanced condition

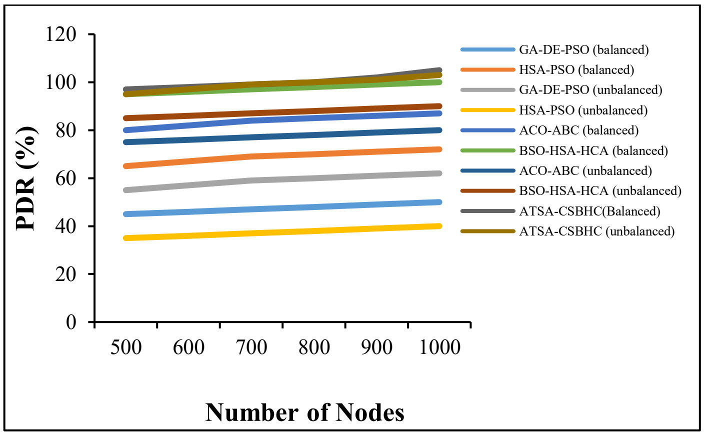

Fig.4. Comparison of PDR for Balanced and Unbalanced condition

The Fig.3 presents the AE2ED across maximizing node counts (500 to 1000) for various routing algorithms under balanced and unbalanced conditions. Among all, ATSA-CSBHC (balanced) achieves the lowest delay, remaining under 1.5 seconds even at 1000 nodes, indicating superior scalability and routing efficiency. In contrast, GA-DE-PSO (balanced) shows the highest delay, exceeding 21 seconds, revealing poor adaptability in larger networks. ACO-ABC and BSO-HSA-HCA maintain consistently low delays in both scenarios, demonstrating robust performance. Overall, ATSA-CSBHC emerges as the most effective, while GA-DE-PSO suffers significant delay degradation with increased network density.

The Fig.4 illustrates the PDR across various node counts (500 to 1000) for different routing algorithms under balanced and unbalanced conditions. Among them, ATSA-CSBHC (balanced) consistently achieves the highest PDR, surpassing 105% at 1000 nodes, indicating exceptional reliability in packet transmission. BSO-HSA-HCA and ACO-ABC (balanced) also show strong performance, maintaining PDRs above 95%. In contrast, GA-DE-PSO and HSA-PSO (unbalanced) record the lowest PDRs, below 65% and 40% respectively, highlighting their inefficiency in high-density networks. Overall, ATSA-CSBHC demonstrates superior reliability and scalability, making it the most effective approach among the evaluated algorithms.

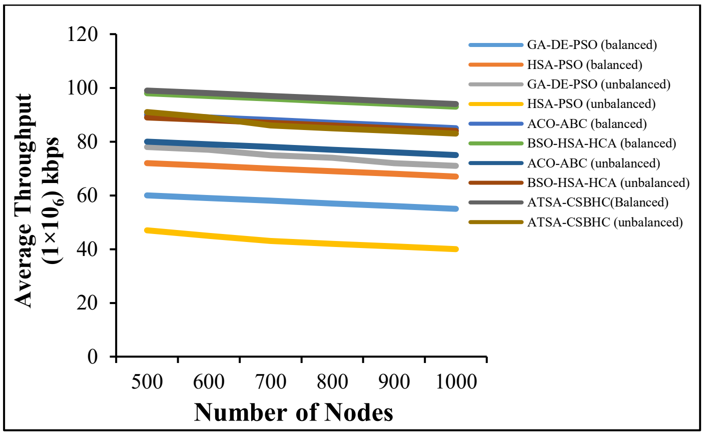

Fig.5. Comparison of Average Throughput for Balanced and Unbalanced condition

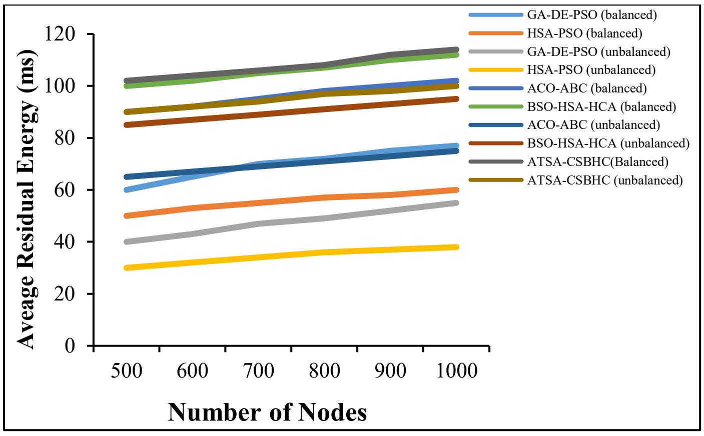

Fig.6 Comparison of ARE for Balanced and Unbalanced condition

The Fig.5 shows the variation in average throughput (×10⁶ Kbps) with increasing node count (500 to 1000) for multiple RP under balanced and unbalanced scenarios. ATSA-CSBHC (balanced) maintains the highest throughput, slightly above 100 ×10⁶ Kbps, followed closely by BSO-HSA-HCA (balanced), demonstrating high DT efficiency and scalability. Conversely, HSA-PSO (unbalanced) shows the lowest throughput, dropping below 45 ×10⁶ Kbps as node density increases. Most algorithms exhibit a gradual decline in throughput with node growth, indicating congestion or overhead. Overall, ATSA-CSBHC proves to be the most robust in sustaining high throughput, especially in balanced conditions.

The Fig.6 displays the average residual energy (in mJ) against the number of nodes (500 to 1000) for various RP under both balanced and unbalanced scenarios. ATSA-CSBHC (balanced) consistently achieves the highest RE, exceeding 110 mJ at 1000 nodes, closely followed by BSO-HSA-HCA (balanced), indicating excellent energy efficiency and conservation. In contrast, HSA-PSO (unbalanced) records the lowest RE, below 40 mJ, reflecting higher energy depletion. All algorithms show an upward trend in RE with increasing nodes, but ATSA-CSBHC and BSO-HSA-HCA clearly outperform others, demonstrating their effectiveness in preserving node energy in dense WSN.

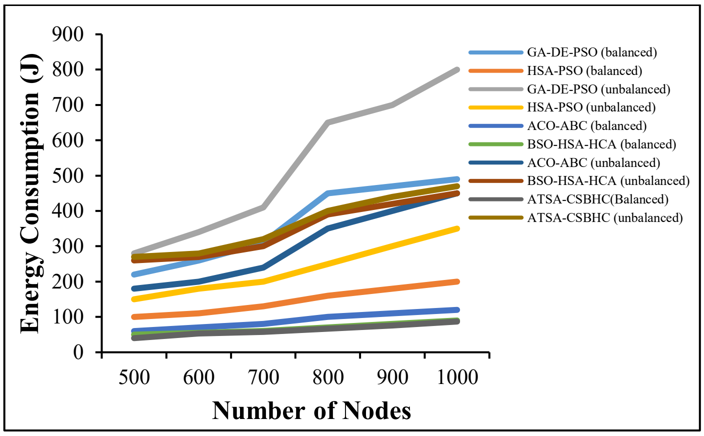

Fig.7. Comparison of EC for Balanced and Unbalanced condition

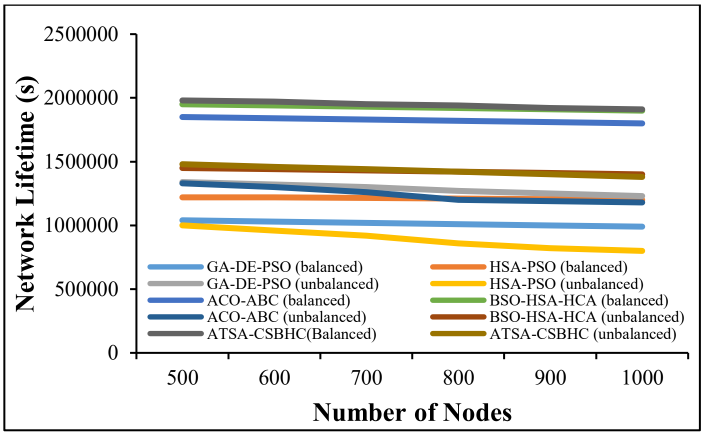

Fig.8. Comparison of NL for Balanced and Unbalanced condition

The Fig.7 depicts the EC (in joules) relative to the number of nodes (500 to 1000) for various algorithms under balanced and unbalanced conditions. GA-DE-PSO (unbalanced) exhibits the highest and steepest rise in EC, reaching around 800 J at 1000 nodes, indicating inefficiency and poor energy management. In contrast, ATSA-CSBHC (balanced) and BSO-HSA-HCA (balanced) maintain the lowest EC throughout, reflecting superior energy-saving capabilities. Most algorithms show increased consumption with node growth, but ATSA-CSBHC remains the most efficient in both balanced and unbalanced modes. Overall, ATSA-CSBHC proves to be the most EE and scalable solution among the compared models.

The Fig.8 illustrates the NL (in seconds) versus the number of nodes (500 to 1000) for several RP under balanced and unbalanced conditions. ATSA-CSBHC (balanced) and BSO-HSA-HCA (balanced) achieve the longest NL, consistently maintaining values near or above 1.9 million seconds, highlighting their superior EE and load distribution.

Conversely, HSA-PSO (unbalanced) shows the shortest lifetime, dropping below 900,000 seconds as node density increases. Most algorithms exhibit a slight decline in lifetime with more nodes, indicating increased EC. Overall, ATSA-CSBHC and BSO-HSA-HCA in balanced configurations are the most effective in maximizing NL.

6. CONCLUSION

Thus, the paper proposes a hybrid metaheuristic framework combining ATSA and CSBHC for EE and balanced clustering in WSN. Through intelligent CHS and effective data aggregation, the method enhances energy conservation, scalability, and network performance. Simulation results across various node densities demonstrate that the ATSA-CSBHC algorithm significantly outperforms existing methods in terms of reduced E2ED, higher PDR, improved T, enhanced RE, minimized EC, and extended NL. The suggested model presents a robust and scalable solution for sustainable WSN operation in real-world applications.

REFERENCES

[1] H. Singh, N. Kumar and M. Kaur, "Energy-Efficient Clustering Protocol based on Improved Cuckoo Search Algorithm for WSNs", *Wireless Networks*, Vol. 27, No. 2, pp. 1145-1160, 2021.

[2] M. Alazab and M. Tang, "Hybrid Metaheuristic Clustering Algorithm for Energy-Efficient WSN Routing", *Future Generation Computer Systems*, Vol. 130, pp. 226-240, 2022.

[3] P.K. Sahu, S.N. Panda and A.K. Sahu, "A Novel Energy-Efficient Hybrid Clustering and Routing Scheme using PSO and GA in WSN", *Journal of Network and Computer Applications*, Vol. 203, pp. 1-9, 2023.

[4] Y. Zhang and Q. Wang, "Cuckoo Search Algorithm-based Energy-Aware Cluster Head Selection for WSNs", *Applied Soft Computing*, Vol. 113, pp. 1-7, 2022.

[5] C. Jin and L. Li, "β-Hill Climbing Embedded Genetic Algorithm for Node Optimization in WSNs", *Sensors*, Vol. 21, No. 4, pp. 1-6, 2021.

[6] S. Tayal and N. Gupta, "A Hybrid Tunicate Swarm-Gray Wolf Optimizer Algorithm for Energy-Aware Clustering in Wireless Sensor Networks", *Wireless Personal Communications*, Vol. 131, pp. 1063-1082, 2023.

[7] Y. Mehmood and M. Imran, "Hybrid Ant Colony and Particle Swarm Optimization for Energy-Efficient Routing in WSNs", *Ad Hoc Networks*, Vol. 112, pp. 1-6, 2021.

[8] R. Gupta and S. Kumari, "Dynamic Cluster Head Selection in WSNs using Improved Levy-Flight-based Cuckoo Search", *Cluster Computing*, Vol. 26, pp. 1889-1904, 2023.

[9] J. Liu and F.R. Yu, "Recent Advances in Machine Learning and Metaheuristic Techniques for WSN Optimization", *IEEE Internet of Things Journal*, Vol. 9, No. 5, pp. 3630-3645, 2022.

[10] H. Wang and T. Ma, "Enhanced PSO with Simulated Annealing for Clustering in Wireless Sensor Networks", *IEEE Access*, Vol. 9, pp. 85678-85691, 2021.

[11] S. Rani and J. Malhotra, "A Review on Metaheuristic-based Cluster Head Selection Algorithms for Wireless Sensor Networks", *Journal of Ambient Intelligence and Humanized Computing*, Vol. 14, No. 2, pp. 1231-1249, 2023.

[12] M. Azizi and M.A. Roudbari, "Load Balancing and Lifetime Enhancement in WSN using Modified Cuckoo Search and Genetic Algorithm", *Wireless Networks*, Vol. 28, pp. 2181-2195, 2022.

[13] M.A. Basha and F. Al-Turjman, "Intelligent WSNs for Green IoT using Hybrid Metaheuristics: A Survey", *Computer Communications*, Vol. 180, pp. 172-184, 2021.

[14] A. Karimian and H. Goudarzi, "Adaptive Clustering using Fuzzy-based Multi-Objective Optimization in WSNs", *Journal of Supercomputing*, Vol. 79, pp. 1862-1884, 2023.

[15] S.T.H. Rizvi and F. Minhas, "A Hybrid Multi-Objective Evolutionary Framework for Optimizing Clustering and Routing in Energy-Constrained WSNs", *IEEE Transactions on Green Communications and Networking*, Vol. 8, No. 1, pp. 112-124, 2024.

[16] C. Zhao, Q. Wu, D. Lin, Z. Zhang, Y. Zhang, L. Kong and Y.L. Guan, "An Energy-Balanced Unequal Clustering Approach for Circular Wireless Sensor Networks", *Ad Hoc Networks*, Vol. 132, pp. 1-6, 2022.

[17] A.A. Jasim, M.Y.I. Idris, S. Razalli Bin Azzuhri, N.R. Issa, M.T. Rahman and M.F.B. Khyasudeen, "Energy-Efficient Wireless Sensor Network with an Unequal Clustering Protocol based on a Balanced Energy Method (EEUCB)", *Sensors*, Vol. 21, No. 3, pp. 1-7, 2021.

[18] P. Maratha, K. Gupta and P. Kuila, "Energy Balanced, Delay Aware Multi-Path Routing using Particle Swarm Optimisation in Wireless Sensor Networks", *International Journal of Sensor Networks*, Vol. 35, No. 1, pp. 10-22, 2021.

[19] M. Navarro, Y. Liang and X. Zhong, "Energy-Efficient and Balanced Routing in Low-Power Wireless Sensor Networks for Data Collection", *Ad Hoc Networks*, Vol. 127, pp. 1-6, 2022.

[20] C.L. Hu, S.Z. Huang, Z. Zhang and L. Hui, "Energy-Balanced Optimization on Flying Ferry Placement for Data Gathering in Wireless Sensor Networks", *IEEE Access*, Vol. 9, pp. 70906-70923, 2021.

[21] B. Fan and Y. Xin, "EBPT-CRA: A Clustering and Routing Algorithm based on Energy-Balanced Path Tree for Wireless Sensor Networks", *Expert Systems with Applications*, Vol. 259, pp. 1-8, 2025.

[22] H.M. Saad, A. Shdefat, A. Nawaz, A.M. El-Sherbeeny, M.A. El-Meligy and M.R.R. Rana, "Enhancing Energy Balance in Wireless Sensor Networks through Optimized Minimum Spanning Tree", *PeerJ Computer Science*, Vol. 10, pp. 1-6, 2024.

[23] S.S. Bhasgi and S. Terdal, "Distributed and Energy Balanced Routing for Heterogeneous Wireless Sensor Network", *Proceedings of International Conference on*

Smart Computing and Informatics, Vol. 2, pp. 711-717, 2021.

[24] D. Loganathan, M. Balasubramani, R. Sabitha and S. Karthik, "Improved Load-Balanced Clustering for Energy-Aware Routing (ILBC-EAR) in WSNs", *Computer Systems Science and Engineering*, Vol. 44, No. 1, pp. 1-7, 2023.

[25] G. Wang, M. Sun, S. Cao, H. Sun, Y. Zhang and Z. Teng, "An Energy-Balanced Ant-based Routing Algorithm for Wireless Sensor Networks", *Proceedings of International Conference on Electronic Information Engineering and Computer Science*, pp. 97-106, 2021.