



SCE-PSO based clustering approach for load balancing of gateways in wireless sensor networks

Damodar Reddy Edla¹ · Mahesh Chowdary Kongara¹ · Ramalingaswamy Cheruku¹

Published online: 3 February 2018
© Springer Science+Business Media, LLC, part of Springer Nature 2018

Abstract

Wireless sensor networks (WSNs) consist of spatially distributed low power sensor nodes and gateways along with sink to monitor physical or environmental conditions. In cluster-based WSNs, the Cluster Head is treated as the gateway and gateways perform the multiple activities, such as data gathering, aggregation, and transmission etc. Due to improper clustering some sensor nodes and gateways are heavily loaded and dies early. This decreases lifetime of the network. Moreover, sensor nodes and gateways are constrained by energy, processing power and memory. Hence, to design an efficient clustering is a key challenge in WSNs. To solve this problem, in this paper we proposed (1) a clustering algorithm based on the shuffled complex evolution of particle swarm optimization (SCE-PSO) (2) a novel fitness function by considering mean cluster distance, gateways load and number of heavily loaded gateways in the network. The experimental results are compared with other state-of-the-art load balancing approaches, like score based load balancing, node local density load balancing, simple genetic algorithm, novel genetic algorithm. The experimental results shows that the proposed SCE-PSO based clustering algorithm enhanced WSNs lifetime when compared to other load balancing approaches. Also, the proposed SCE-PSO outperformed in terms of load balancing, execution time, energy consumption metrics when compared to other existing methods.

Keywords Shuffled complex evolution · Particle swarm optimization · Wireless sensor networks · Clustering · Network lifetime · Load balancing

1 Introduction

WSNs are networks that consist a large number of tiny and low energy sensor nodes. These nodes are randomly or manually spread in a given target area or region. The sensor node contains data aggregation unit, sensing unit, the communication component and also an energy unit. Each sensor node may have Global Position System (GPS) to progress within the given target area or region. WSNs have promising applications in different areas such as

disaster warning systems, air pollution monitoring, health care, environmental monitoring, and agriculture and in crucial areas such as surveillance, intruder detection, defense reconnaissance etc [1, 21]. Sensor networks are used to gather data from the environment and build inferences about the monitored object. These sensor nodes are normally characterized by limited communication capabilities due to power and bandwidth constraints. So, minimizing energy conservation of sensor nodes and Cluster Heads (CHs) in the network is challenging task in WSNs. Many clustering algorithms are studied more in this regard [6, 14, 15]. In cluster based WSNs, sensor nodes are divided into several groups called clusters. Each cluster has a CH, which gathers data from nearby sensor nodes and processes that data then send to sink directly or via other CHs. In cluster based WSNs, CHs have more workload compared to other sensor nodes [7, 20]. So, many researches [16, 17] deployed high energy level devices as CHs,

✉ Damodar Reddy Edla
dr.reddy@nitgoa.ac.in

Mahesh Chowdary Kongara
maheshchowdary.k.cse@gmail.com

Ramalingaswamy Cheruku
rmlswamygoud@gmail.com

¹ National Institute of Technology Goa, Ponda, Goa 403401, India

also known as gateways in WSNs. This entire process is depicted in Fig. 1.

In a clustered WSNs, CHs need to do some additional activities such as, data collection from sensor nodes within the respective cluster, processing etc. apart from handling its own data. CHs use data fusion to detect duplicate and unwanted data sent by sensor nodes in the respective cluster. These Gateways or CHs are operated with battery power. So, we need to develop an algorithm that maximizes network lifetime and minimizes power consumption in sensor nodes as well as gateways. Sensor nodes and gateways usually run on low powered batteries, so they need a better way to utilize its battery power in an efficient way. Gateways may send data to another gateway or directly to the sink depending on the network structure and availability of devices. Data processing capability of a gateway is directly proportional to its power. If any gateway fails due to lack of energy, it effects dependent devices in the network. If any intermediate node fails, the nodes using this intermediate node have to search for other alternative nodes to send the data and causes more load on newly identified alternative nodes. In the literature various definitions defined for network lifetime, such as the time until the first node dies or time until the last node dies or time until the desired number of nodes dies in the network [3, 24]. In some WSN scenarios, network lifetime is considered as the time taken to cover entire region. In [25, 26] authors have defined network lifetime in terms of node lifetime, coverage, connectivity and transmission.

With the evolution of soft computing, bio-inspired algorithms have gained more attention in recent years. Many researchers designed algorithms to solve WSNs problems. For example, Genetic Algorithm (GA) [19], which is used to enhance WSNs lifetime in large-scale surveillance applications. Artificial fish schooling algorithm [27], which has been used to solve traffic monitoring system and fruit fly optimization algorithm [29] is applied in sensor deployment. In this paper, to address the WSNs

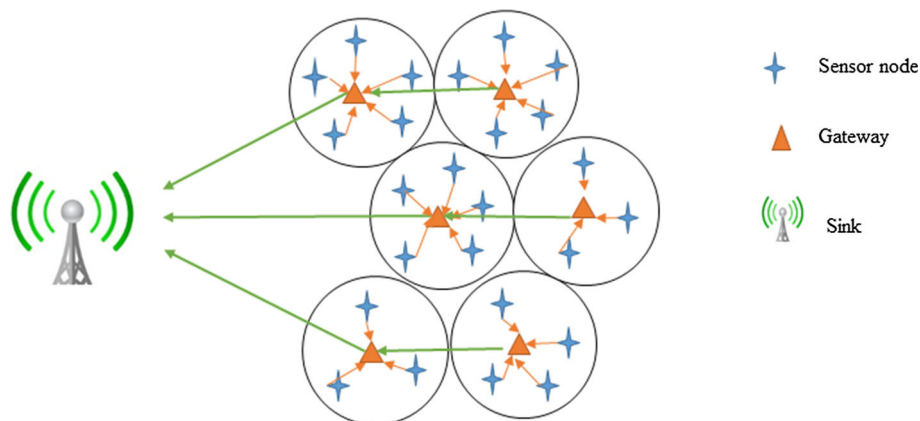
energy problem, SCE-PSO based energy efficient clustering algorithm is proposed. The rest of the paper is organized as follows: works related to clustering in WSNs are discussed in Sect. 2. The required terminology, energy model for WSNs is discussed in Sect. 3 to understand the proposed method clearly. The problem statement for load balancing is discussed in Sect. 4. The SCE-PSO for load balancing problem in WSNs is discussed in Sect. 5. The proposed SCE-PSO based energy efficient clustering algorithm is discussed in Sect. 6. The experimental results and its analysis are presented in Sect. 7. Finally, in Sect. 8, the paper is concluded based on experimental observations.

2 Related work

Many clustering algorithms have been proposed for WSNs. We review some of the clustering algorithms in this section. Low-Energy Adaptive Clustering Hierarchy (LEACH) is well-known clustering algorithm used for WSNs. In LEACH algorithm, one of the sensor node in the network selected as a CH for the cluster. This algorithm dynamically changes the CHs in the network which is good for load balancing. Suppose low energy node selected as a CH that may die rapidly. The main problem in this algorithm is low energy level CHs die fast inside the network [9]. For solving this problem many improved LEACH algorithms have been proposed for selecting optimal CHs. In [18] authors have proposed Improved LEACH (I-LEACH) by considering residual energy to select the CH instead of probability as used in the LEACH. In [2] authors have selected energy efficient CH by considering the energy and distance parameters.

Zhang et al. [28] have defined Node Local Density Load Balancing (NLDLB) with clustering in WSN. In this algorithm, initially they assigned the sensor node which is in range of R with only one gateway. The remaining sensor nodes are assigned in two steps: first they have assigned the

Fig. 1 Cluster based WSN with gateways



sensor nodes to the gateways which are in $R/2$ range, then remaining sensor nodes are assigned to the gateways which are in range and having less number of sensor nodes. Vaishali et al. [6] have proposed Score Based Load Balancing (SBLB) algorithm in WSNs. They have considered score parameter to select intermediate nodes in the path. The node which has minimum distance and maximum energy becomes the best score node in that path. The two best score nodes are chosen for data transmission in this algorithm. These scores are estimated based on ratio of distance between each node to CH, and energy of the each node. After forming the clusters, the CH is selected. The sensor node which has maximum energy becomes the CH.

In WSNs, For energy efficiency and load balancing Jana et al. [15] have defined two types of sensor nodes which depends on range of communication between gateways and sensor nodes which are restricted node and open node. The sensor nodes which are in range of one and only one gateway i.e., they can communicate with only one gateway are referred as restricted nodes whereas sensor nodes which can communicate with at least two or more than two gateways are referred as open nodes. Initially all restricted nodes are allocated to their assigned gateways. For allocation of open nodes min heap is build. The nearest gateway is assigned to the sensor node and rearrangement of min heap takes place. Kuila et al. [14] have come up with a Novel Genetic Algorithm (NGA) and Hussain et al. [10] have proposed Simple GA (SGA) based on clustering of sensor nodes for load balancing in WSNs. It usually starts with a set of randomly generated number of solutions known as initial population. Every solution is denoted by an array of genes or as a string, which is known as chromosome. To judge the performance of each solution, a fitness function is evaluated. Damodar et al. [5] have improved shuffled frog leap algorithm for gateways load balancing in WSNs. Kuila et al.[16] have used PSO-based clustering algorithm to enhance the lifetime of WSNs. In this method, clustering takes place based on the average cluster distance and lifetime of the gateway. Fitness value of each particle in the swarm is computed using fitness function and this fitness value judges the quality of the network. A particle with better fitness function value gives better network structure. Xiang et al. [22] have been proposed PSO-based energy efficient routing algorithm by considering residual energy of nodes and transmission distance. Zhou et al. [30] have been proposed improved PSO-based clustering algorithm by considering energy efficiency, transmission distance and relay nodes residual energy.

3 Preliminaries and terminologies

3.1 Energy model

We have used similar energy model used in [8]. Energy consumption of sensor nodes and gateways effect the WSNs lifetime. Hence, it is important to design an efficient clustering technique where every gateway and sensor node energy is mitigated in order to prolong network lifetime. In WSNs, radio signals are used for communication between nodes and the energy consumed for data transmission and reception is evaluated using energy model. There are two channels in energy model namely free space channel and multi-path fading channel. If the distance (d) between sender and receiver is less than a threshold d_0 , then free space channel is used; otherwise, multi-path fading channel is used. The following equation shows the energy required by the model to send a l -bit message over a distance d .

$$E_T(l, d) = \begin{cases} l * E_{elec} + l * \epsilon_{fs} * d^2, & d < d_0 \\ l * E_{elec} + l * \epsilon_{mp} * d^4, & d \geq d_0 \end{cases} \quad (1)$$

where d_0 is a threshold, E_{elec} is the energy required by electronic circuit, ϵ_{fs} is the energy required by the free space and ϵ_{mp} is the energy required by multi-path channel.

The energy required to receive l -bit of data is given as:

$$E_R(l) = l * E_{elec} \quad (2)$$

The energy consumed by an electronic circuit (E_{elec}) depends on various factors, such as the digital coding, modulation, filtering and spreading of the signal etc. The energy consumed by the amplifier in free space ($\epsilon_{fs} * d^2$) or multi-path ($\epsilon_{mp} * d^4$) depends on the distance between the sender and receiver and the bit error rate.

3.2 Terminologies

We have used the below terminologies in the proposed fitness function and proposed algorithm.

1. The set of sensor nodes is indicated by $S = \{s_1, s_2, \dots, s_m\}$.
2. The set of gateways is indicated by $G = \{g_1, g_2, \dots, g_n\}$.
3. $Dist(s_i, g_j)$ The distance between the sensor node s_i and the gateway g_j .
4. $Range(s_i)$ The number of gateways within communication range of s_i .
5. $Load(g_i)$ depends on amount of energy required for gateway g_i in order to receiving and transmitting the l number of bits ($E(l)$) and remaining energy of gateway.

$$\text{Load}(g_i) = E(l)/\text{remaining energy in } g_i \quad (3)$$

where, $E(l)$ is calculated according to energy model and is given below.

$$E(l) = E_T(l, d) + E_R(l)$$

6. Load Factor(g_i) The ratio between the $\text{Load}(g_i)$ and maximum load of the gateway in the network.

$$\text{Load Factor}(g_i) = \frac{\text{Load}(g_i)}{\max\{\text{Load}(g_i), \forall i = 1, 2, \dots, n\}} \quad (4)$$

7. Expected Gateway Factor is the ratio between the sum of load factors of all gateways and total number of gateways in the network.

$$\text{Expected Gateway Factor} = \left[\frac{\sum_{i=1}^n \text{Load Factor}(g_i)}{\text{Total number of gateways}} \right] \quad (5)$$

8. $M(g_j)$: Average euclidean distance from each sensor node to gateway in that particular cluster.

$$M(g_j) = \left[\frac{\sum_{i=1}^m \text{Dist}(s_i, g_j)}{K_j} \right] \quad (6)$$

where K_j = Total number of sensor nodes connected to gateway g_j

4 Problem formulation

We have assumed a WSN scenario, in that scenario the sensor nodes and gateways are deployed randomly in given target area. Any sensor node in the network can connect any gateway in the network if the sensor node in the communication range of that gateway. Figure 2 shows that the load balancing problem in WSNs. According to the Fig. 2 g_1, g_2, g_3 are three gateways in the network and K_1, K_2, K_3 are three clusters. g_1, g_2, g_3 are CHs for K_1, K_2, K_3 clusters respectively. K_1 is over loaded cluster, so this load

has been distributed to g_2 and g_3 with proper assignment of sensor nodes.

5 Adopting SCE-PSO for load balancing problem in WSNs

5.1 Overview of SCE-PSO

The SCE-PSO is an evolutionary algorithm and introduced by [23]. In SCE-PSO approach the SCE approach combined with the PSO. In the SCE-PSO, a population of points sampled randomly from the feasible space. After initialization of particles velocity and position, the population is divided into a predefined number of complexes N . the complex division is based on fitness value of the particle. Now each complex evaluating using PSO algorithm. The detail explanation of SCE-PSO algorithm is given below and flowchart of SCE-PSO algorithm is shown in Fig. 3 [11, 12].

Step 1 *Particles Initialization* Let $N \geq, M \geq 1$ two numbers, where N is the number of complexes, M is the number of particle in each complex. Calculate the sample size $S = NM$. P_1, P_2, \dots, P_S are S sample points (particles). Initialize all particles in sample and calculate fitness value f_i at each particle P_i

Step 2 *Particles Ranking* Store the particles in an array A according to descending order of fitness value.
 $A = \{P_i, f_i, i = 1, 2, 3, \dots, S\}$

Step 3 *Particles Partitioning* Partition an array A in to N number of complexes C_1, C_2, \dots, C_N . Each complex containing M particles in such a way that the particle P_1 allotted to complex C_1 , the particle P_2 allotted to complex C_2 , the particle N is allotted to the C_N complex, and the particle $N + 1$ allotted to the C_1 complex, and so on.

Fig. 2 Load balancing problem in WSN

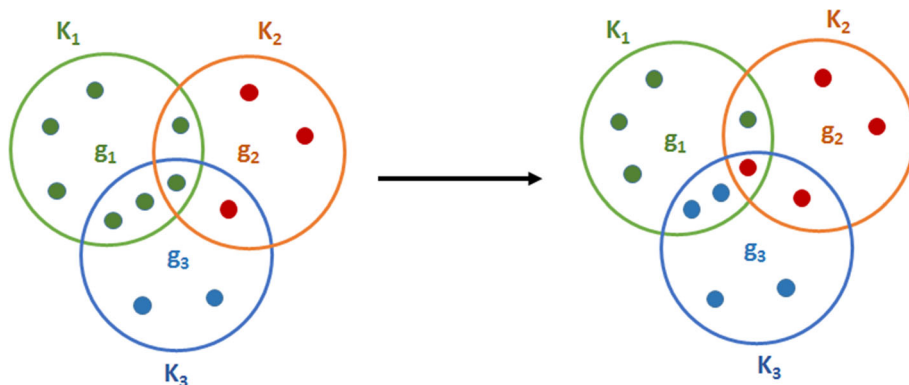
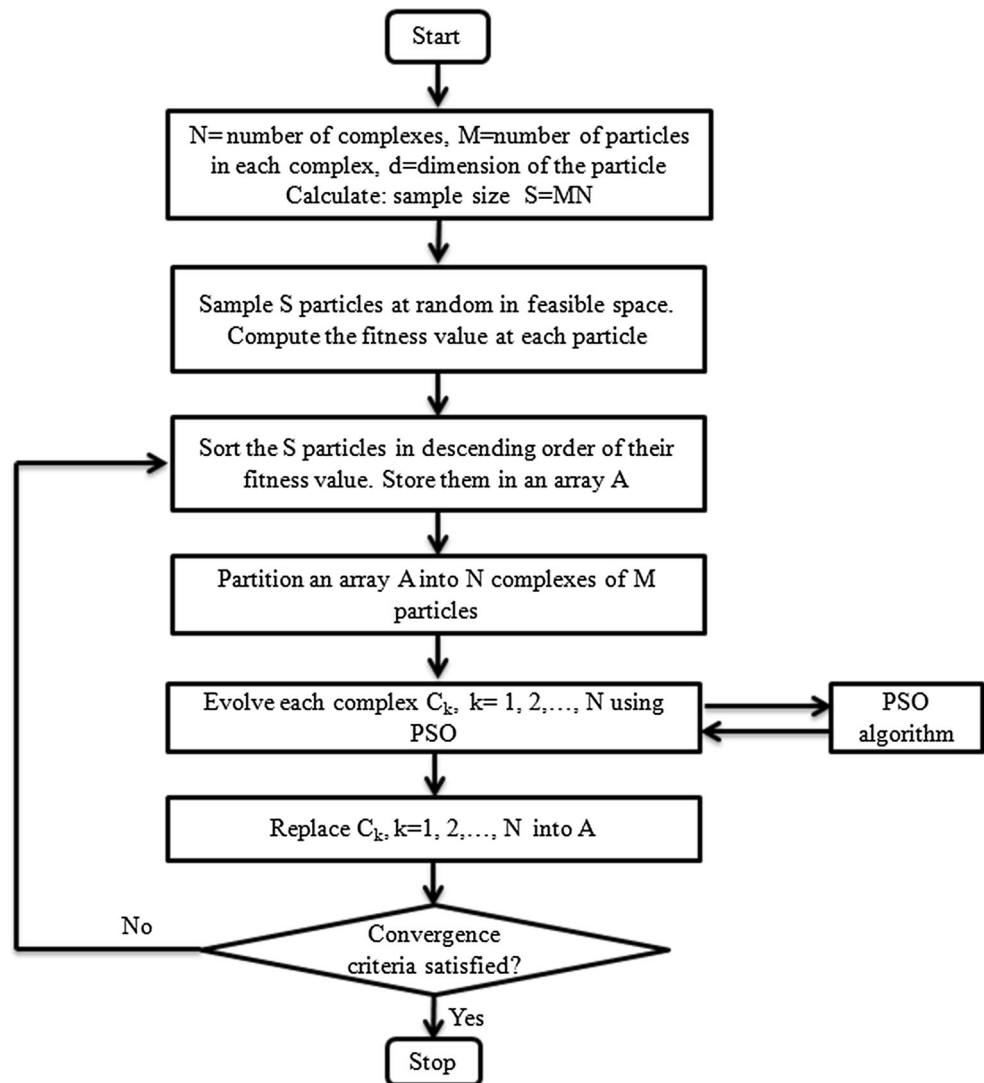


Fig. 3 Flowchart of SCE-PSO algorithm

$$C_k = \{ P_j, f_j, j = 1, 2, \dots, M \}$$

- Step 4 *Complex Evaluation* Each complex C_k is being evaluated, where $k = 1, 2, \dots, N$ according to PSO algorithm. PSO algorithm is described further.
- Step 5 *Complex Shuffling* Replace C_1, C_2, \dots, C_N into an array A. Sort an array A with descending order according to their fitness value.
- Step 6 *Convergence Checking* Check for convergence criteria is satisfied or not. If it is satisfied, stop; else go to step 3. Here convergence criteria means number of evaluations or fitness value limitation.

5.2 Adopting PSO for complex evolution

PSO is inspired from the nature based on fish schooling and bird flocking [4, 13]. These birds regularly travel together

in a group without any colliding for searching food or shelter. Each member or bird in a group follows the group information by adjusting its velocity and position. Each bird or member individual effort searching for food and shelter reduces in a group because of sharing group information. Each particle in complex gives solution to a specified instance of the problem. Each particle is going to be evaluated by a fitness function. All particles have same dimension. Each particle P_i has a position $(X_{i,d})$ and velocity $(V_{i,d})$ in d th dimension of the hyperspace. So, at any point in time, particle P_i is represented as

$$P_i = \{X_{i,1}, X_{i,2}, X_{i,3}, \dots, X_{i,d}\}$$

To reach global best position, each particle P_i follows its own best ($Lbest_i$) and global best ($Gbest$) to update its position and velocity recursively. Using the following equations, we perform the recursive position (X_i) and velocity (V_i) updates [16].

$$V_{i,d}(t+1) = W * V_{i,d}(t) + a_1 * r_1 * (Lbest_{i,d} - X_{i,d}(t)) + a_2 * r_2 * (Gbest_d - X_{i,d}(t)) \quad (7)$$

$$X_{i,d}(t+1) = X_{i,d}(t) + V_{i,d}(t) \quad (8)$$

where W is the inertial weight, a_1 and a_2 are the acceleration constants which are non-negative real numbers. r_1 and r_2 are two random numbers in the range $[0, 1]$ which are uniformly distributed. This update process is repeated until we find global best or we reach maximum iterations count. This whole process is shown in Fig. 4. In every iteration, the position and velocity of each particle are updated using Eqs. 7 and 8. While updating velocity and positions of particles we may get new position because of algebraic addition and subtraction. New position values may be less than or equal to zero, or greater than one. However, according to Eq. 9, the particles position must follow in the range of $(0, 1]$. In order to get correct range values, we have to do the following changes in our algorithm:

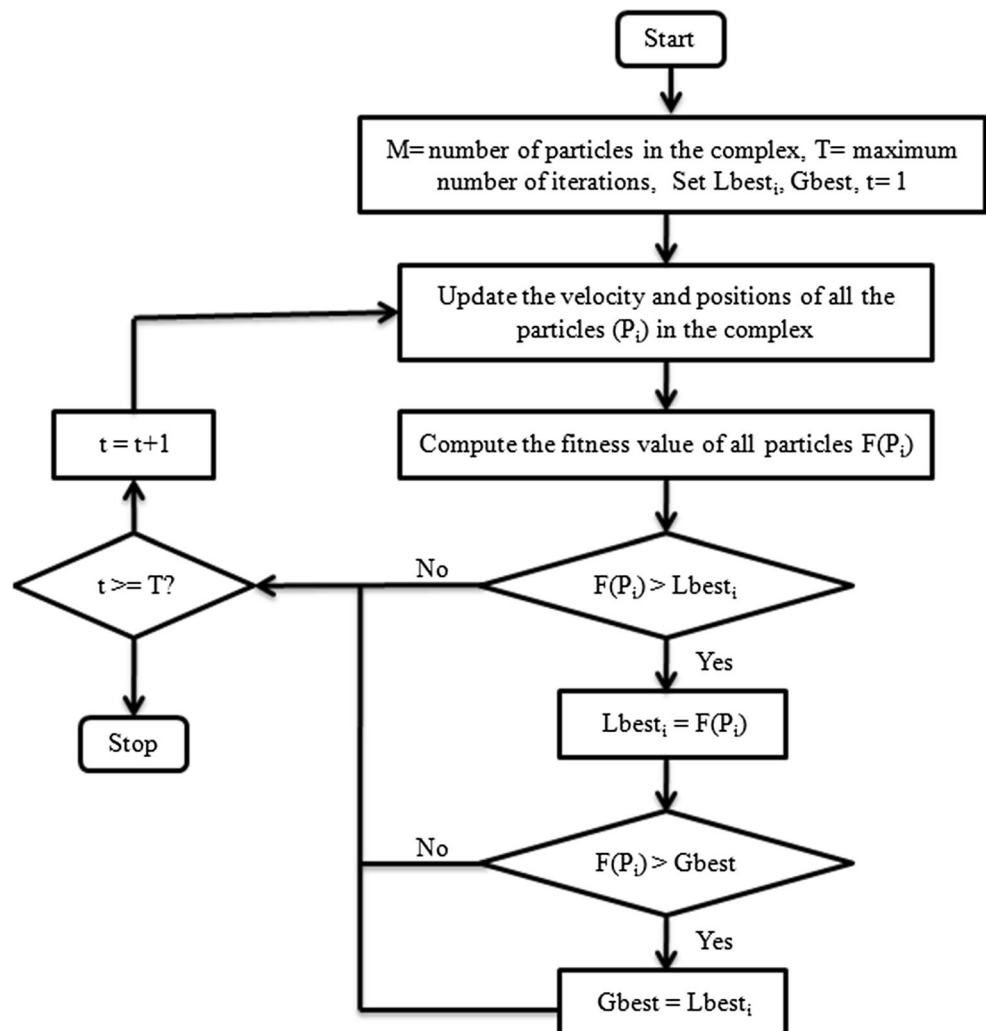
1. If updated position value is less than or equal to zero, then modify the value with a newly generated random number whose value tends to zero.
2. If updated position value is above one, then modify the value to one.

After obtaining the modified positions, the particle P_i is evaluated using fitness function. Each particle best fitness value ($Lbest_i$) is modified by itself, only if its present fitness value better than $Lbest_i$ fitness value. Until the termination condition is satisfied, the velocity and the position values are modified recursively.

6 Proposed clustering algorithm

To solve the problem of balancing the load of gateways in wireless sensor network, an optimization algorithm, SCE-PSO strategy is applied. The proposed algorithm contain particle generation, formulation of fitness function, sorting of particles with respect to their fitness value, partitioning

Fig. 4 Flowchart of PSO algorithm



into complexes, Evaluation of complex using PSO in the sections as follows.

6.1 Particle generation

Each particle is encoded as clustered network (contains connection from each sensor node to gateway). The dimension of each particle P is D (number of sensor nodes in the network). We initialize each gateway $(X_{i,d} | 1 \leq i \leq M, 1 \leq d \leq D)$ with randomly generated number from a uniform distribution in the range $(0, 1]$. The value of d th sensor node (i.e., $X_{i,d}$) assigns a gateway (say g_k) to s_d . That is, s_d sends data to g_k and this can be formulated as follows:

$$n = \text{ceil}(|X_{i,d} * \text{Range}(s_d)|) \tag{9}$$

i.e., g_k is the n th gateway in $\text{Range}(s_d)$.

The particle initialization process is explained using following Example 1.

Example 1 Consider a WSN scenario with 4 gateways and 10 sensor nodes, i.e., $G = g_1, g_2, g_3, g_4$ and $S = s_1, s_2, s_3, \dots, s_{10}$. As number of sensor nodes are 10 therefore the dimension of particle is 10. The Fig. 5 shows that the sensor nodes communication range and the edge between s_i, g_j represents sensor node s_i can send data to the gateway g_j . The same is represented using Table 1 in terms of sensor nodes, gateways in the range of sensor nodes.

During initialization, each dimension (sensor node) $X_{i,d}$ of particle P_i is initialized with a random number in $(0, 1]$ range. For a random particle P_i the values assigned are shown in Table 2. This random initialization of sensor nodes position values constitute a particle or solution. This solution is generated as follows:

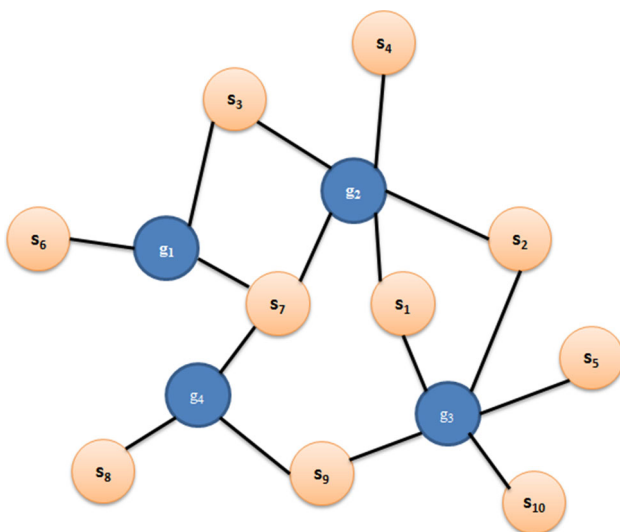


Fig. 5 WSN before clustering

Table 1 Sensor nodes and their communication range

Sensor nodes	$\text{Range}(s_i)$	$ \text{Range}(s_i) $
s_1	$\{g_2, g_3\}$	2
s_2	$\{g_2, g_3\}$	2
s_3	$\{g_1, g_2\}$	2
s_4	$\{g_2\}$	1
s_5	$\{g_3\}$	1
s_6	$\{g_1\}$	1
s_7	$\{g_1, g_2, g_4\}$	3
s_8	$\{g_4\}$	1
s_9	$\{g_3, g_4\}$	2
s_{10}	$\{g_3\}$	1

Let us consider sensor node s_2 and its assigned random value of 0.65. According to the Eq. 9, $\text{ceil}(0.65, 2) = 2$, i.e., second gateway (g_3) from the $\text{Range}(s_2)$ is selected for data sending. This process is repeated for all sensor nodes to construct entire network (particle). Table 2 shows the final result of network construction, and the same is visualized using Fig. 6.

6.2 Initial sample representation

Initial sample is a set of particles which are randomly generated. Every particle represents the solution to the load balancing problem. All particles are considered to be valid if sensor node is allocated to one of the gateway which is in the range of corresponding sensor node. This initial sample generation is explained with an example given below:

Example 2 Consider a WSN Scenario with 4 gateways and 10 sensor nodes. i.e., $G = g_1, g_2, g_3, g_4$ and $S = \{s_1, s_2, \dots, s_{10}\}$. As there are 10 sensors, the dimension of each particle is 10. Table 1, shows gateways in range of each sensor node. A sensor node can be allocated to the any of the gateways in their range. According to Table 2, sensor node s_1 is assigned to either gateway g_2 or g_3 and so on.

From the Example 2, sensor node s_1 has selected gateway g_2 among g_2 and g_3 , likewise s_2 selects gateway g_3 among g_2, g_3 . s_7 selects g_2 among g_1, g_2, g_4 , and so on. This forms the initial sample generation, which is shown in Table 2.

6.3 Proposed fitness function

An efficient fitness function is designed to evaluate each particle. This is given in Eq. 10. For efficient clustering we considered a metrics called expected gateway factor and mean cluster distance and number of heavy loaded gateways in the fitness function. The considered metrics measures the quality of the network.

Table 2 Particle generation

Sensor nodes	$X_{i,d}$	$ Range(s_d) $	$ceil(X_{i,d} * Range(d))$	Selected gateway
s_1	0.43	2	1	g_2
s_2	0.65	2	2	g_3
s_3	0.28	2	1	g_1
s_4	0.32	1	1	g_2
s_5	0.68	1	1	g_3
s_6	0.98	1	1	g_1
s_7	0.50	3	2	g_2
s_8	0.72	1	1	g_4
s_9	0.57	2	2	g_4
s_{10}	0.16	1	1	g_3

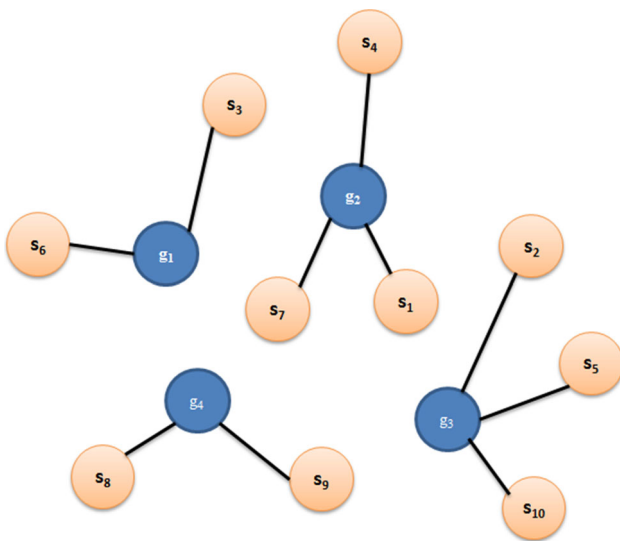


Fig. 6 WSN after clustering

$$Fitness = \text{Expected Gateway Factor} * \frac{\text{Total number of gateways}}{\text{Number of heavily loaded gateways}} * \frac{1}{\sigma_c} \tag{10}$$

In order to check the number of heavily loaded gateways in the network, every gateway is checked for whether the gateway is heavily loaded or not. To check the status of gateway threshold load is defined in Eq. 11. The gateway which load is more than the threshold value called heavily loaded gateway in the network.

$$Threshold = \frac{\sum_{i=1}^n Load(g_i)}{n} \tag{11}$$

$$Standard\ Deviation(\sigma_c) = \sqrt{\frac{\sum_{i=1}^n (\mu - M(g_i))^2}{n}} \tag{12}$$

$$\mu = \frac{\sum_{i=1}^n M(g_i)}{n} \tag{13}$$

It is clear from Eq. 10 that proposed fitness function considers load of the gateways, mean cluster distance and number of heavily loaded gateways to judge the quality of the particle. In Clustered WSN, The sensor nodes sends data to the CHs or Gateway. The number of connected sensor nodes increases for gateway load increases. i.e., the number of sensor nodes to be assigned are proportional to the gateway load. This helps in reducing the gateway energy consumption and maximizing the network lifetime. Hence, proposed fitness function is good for load balancing as well as energy efficiency. Particle which is having higher fitness value that is well balanced clustering network.

6.4 Particles sorting and partitioning

Every particle in the sample is evaluated by the fitness function to get a fitness value. Then store the particles in an array A according to descending order of fitness value, where $A = \{P_i, f_i | i = 1, \dots, S\}$, Here first element in an array A represents a solution to the problem with highest fitness value. after this process, Partition an array A into N number of complexes C_1, C_2, \dots, C_N . each complex containing M particles in such a way that the first particle P_1 allotted to complex C_1 , the particle P_2 allotted to complex C_2 , the particle M is allotted to the C_N complex, and the particle $M + 1$ allotted to the C_1 complex, and so on.

Example 3 Let us Consider 9 particles, 3 complexes in our approach and each complex contains 2 particles. There are 9 particles taken randomly for initial sample generation. Let us consider set of particles in sample $S = \{P_1, P_2, \dots, P_9\}$. These 9 particles are evaluated and sorted in an array A such that P_1 has highest fitness value and P_9 has lowest fitness value. Then an array is partitioned into 3 complexes. This process is shown in Table 3.

Complex	Particle 1	Particle 2	Particle 3
C_1	P_1	P_4	P_7
C_2	P_2	P_5	P_8
C_3	P_3	P_6	P_9

6.5 Complex evolution using PSO

Each complex in the sample is evaluated using PSO algorithm. From Example 2, consider complex C_1 contains P_1, P_4, P_7 particles. We have to evaluate all particles in the complex C_1 upto maximum number of iterations T . In complex evaluation, First update each particle velocity and positions in the complex using Eqs. 7 and 8. After updating velocity and positions in D dimension of particle we may get better particle. The process of evaluating each complex using PSO shown in Fig. 4.

6.6 Complexes shuffling

The fitness value of newly generated particles has to calculate then replace complexes C_1, C_2, \dots, C_N into an array A . Sort an array A with descending order of their fitness value along with their position information. this process repeats until the convergence criteria satisfies (convergence criteria may be the number of evaluations or expected fitness value of the particle).

This entire process in proposed algorithm including all these steps is explained in Algorithm 1.

Algorithm 1 Pseudo code for SCE-PSO based Clustering

- **Input**
 - A set of sensor nodes $S = \{s_1, s_2, \dots, s_m\}$.
 - A set of gateways $G = \{g_1, g_2, \dots, g_n\}$, where $n < m$.
 - $Range(s_i)$: Set of all those gateways which are within the range of communication of s_i .
 - T = Maximum number of iterations.
 - N = Number of complexes.
 - M = Number of particles in each complex.
 - D = Dimension of each particle.
 - A set of solutions $S = \{P_1, P_2, \dots, P_S\}$.
 - $Lbest$ = Local best solution
 - $Gbest$ = Global best solution among all complexes.
 - **Output**
 - An assignment $A: S \rightarrow G$
 - **Algorithm**
 - 1: **Step 1:**
 - 2: Initialize the velocity and positions of all particles in the sample.
 - 3: **Step 2:**
 - 4: compute fitness value for every solution (particle) in sample.
 - 5: **Step 3:**
 - 6: Store all particles in an array A with decreasing order of their fitness values.
 - 7: Choose a random number N .
 - 8: Partition an array A in N complexes say C_1, C_2, \dots, C_N each containing M particles, such that $P_1 \in C_1, P_2 \in C_2, \dots, P_M \in C_N, M+1 \in C_1$.
 - 9: **Step 4:**
 - 10: Evolve each complex using PSO.
 - 11: **for** $i = 1$ to M **do**
 - 12: **if** $F(P_i) > Lbest_i$ **then**
 - 13: $Lbest_i = F(P_i)$
 - 14: **end if**
 - 15: **if** $F(P_i) > Gbest$ **then**
 - 16: $Gbest = F(P_i)$
 - 17: **end if**
 - 18: **for** $d = 1$ to D **do**
 - 19: Update particle's velocity and positions using Eqs. (7) and (8) respectively.
 - 20: **end for**
 - 21: **end for**
 - 22: **Step 5:**
 - 23: Replace C_1, C_2, \dots, C_N into an array A . Sort an array A with descending order according to their fitness value.
 - 24: **if** convergence criteria is satisfied **then**
 - 25: Stop
 - 26: **else**
 - 27: goto **Step 3**
 - 28: **end if**
-

Table 4 Simulation parameters

Parameter	Value	Parameter	Value
<i>WSN parameters</i>		<i>PSO parameters</i>	
Area	$50 \times 50 \text{ m}^2$	Max no. of iterations	500
Sink	(48, 25)	a_1	1.4962
Communication range	10 m	a_2	1.4962
E_{elec}	50 nJ/bit	W	0.7968
E_{DA}	5 nJ/bit	V_{max}	0.5
ϵ_{fs}	10 pJ/bit/m ²	V_{min}	- 0.5
ϵ_{mp}	0.001 pJ/bit/ m ²		
Number of sensors	50, 100, 150		
Number of gateways	5–19		
Number of complexes	5–15		

7 Results and discussion

7.1 Experimental setup

The experiments are performed using MATLAB 2015a on an Intel(R) Core(TM) i3 processor, 3.40 GHz CPU and 4 GB RAM running on the platform Microsoft Windows 8. The experiments are performed by assuming a WSN scenario with number of sensor nodes and gateways positioned in $50 \times 50 \text{ m}^2$ area with the sensing range of sensor node is 10 m. The simulation parameters for WSN and PSO-SCE algorithm are shown in Table 4.

7.2 Performance analysis

The proposed SCE-PSO algorithm compared with state-of-the-art load balancing techniques namely, Score Based Load Balancing (SBLB), Node Local Density Load balancing (NLDLB), Simple Genetic Algorithm (SGA), Novel Genetic Algorithm (NGA). The performance of proposed load balancing algorithm is evaluated in terms of network lifetime, energy consumption, execution time, number of sensor nodes die and number of gateways die.

7.2.1 Number of sensor nodes versus network lifetime

Network lifetime is the total number of rounds from the start of the operation of a WSN until the death of the last alive gateway or sensor node. It is measured in terms of number of rounds. Experiments are performed using number of sensor nodes 50, 100 and 150 with equal and unequal load on sensor nodes. For comparison purpose we have carried out experiments using SGA, NGA, SBLB, and NLDLB and proposed algorithm. The simulation results are shown in Fig. 7(a), (b) for equal and unequal load respectively.

It is clear from the figure that the proposed SCE-PSO has more network life as compared to SGA, NGA, SBLB, and NLDLB algorithms.. It is due to consideration of mean cluster distance, expected gateway load, heavily loaded gateways in proposed fitness function.

7.2.2 Number of sensor nodes versus execution time

Experiments are performed using number of sensor nodes 50, 100 and 150 with equal and unequal load on sensor nodes. For comparison purpose we have carried out experiments using SGA, NGA, SBLB, NLDLB and proposed algorithm. For each algorithm we calculated time required to terminate the algorithm. These comparison results are shown in Fig. 8. Figure 8(a), (b) shows the results for equal and unequal load respectively.

It is clear from figure results that for both equal and unequal loads, the proposed algorithm requires takes less time as compared to SGA, NGA, SBLB, and NLDLB algorithms. It is the fact that PSO velocity and position equations gives best particle position in less time.

7.2.3 Number of sensor nodes versus energy consumption

Energy consumption is a metric to measure the consumed energy (in Joules) per round by an algorithm. For comparison purpose SGA, NGA, SBLB, NLDLB and proposed SCE-PSO are experimented for number of sensor nodes 50, 100 and 150 with equal and unequal loads. These comparison results are shown in Fig. 9. Results for equal and unequal load are shown in Fig. 9(a), (b) respectively.

It is observed from the figure results that proposed SCE-PSO algorithm gives better results as compared to SGA, NGA, SBLB, and NLDLB algorithms for both equal and unequal loads. It is due to proposed fitness function considers remaining energy of gateways and expected gateway load while assigning the sensors. Hence, the solution produced by proposed algorithm is energy efficient.

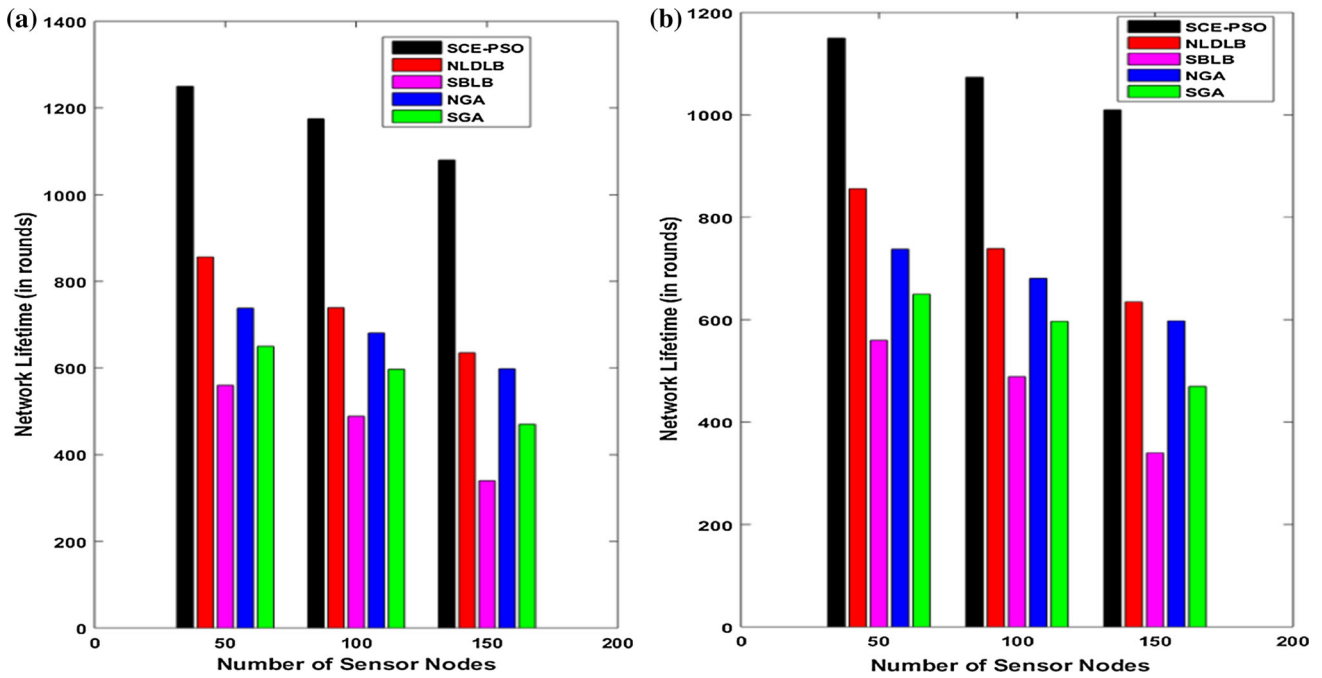


Fig. 7 Comparison of the proposed SCE-PSO with SGA, NGA, SBLB, and NLDLB algorithms in terms of number of sensor nodes and network lifetime for: a Equal load and b Unequal load

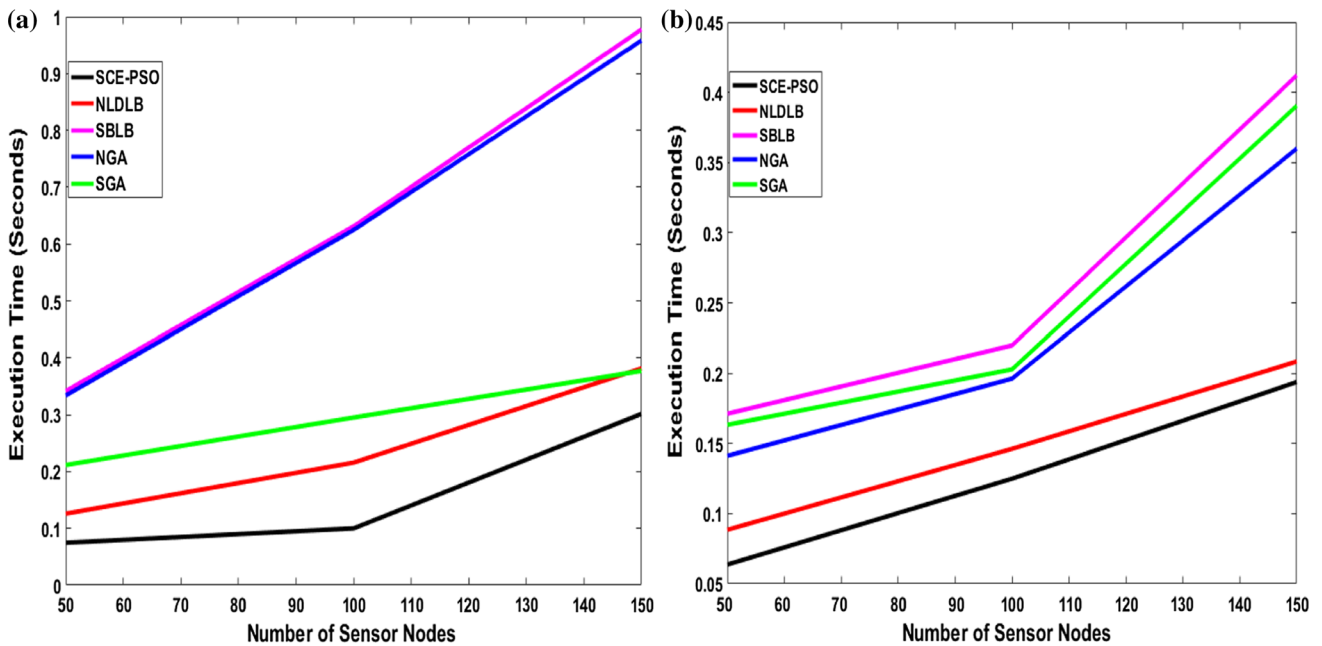


Fig. 8 Comparison of the proposed algorithm with SGA, NGA, SBLB, and NLDLB algorithms in terms of number of sensor nodes and execution time for: a Equal load and b Unequal load

7.2.4 First gateway die and half the gateways die

First gateway die is a metric measures the life span of first gateway in the network. It means number of rounds when the first gateway dies. The highest first gateway die value

indicates the longer stability period of network. For comparison purpose SGA, NGA, SBLB, NLDLB and proposed SCE-PSO are experimented with 50, 100 and 150 sensor nodes and 15 maximum gateways. These experimental results are shown in Fig. 10. After how many rounds the

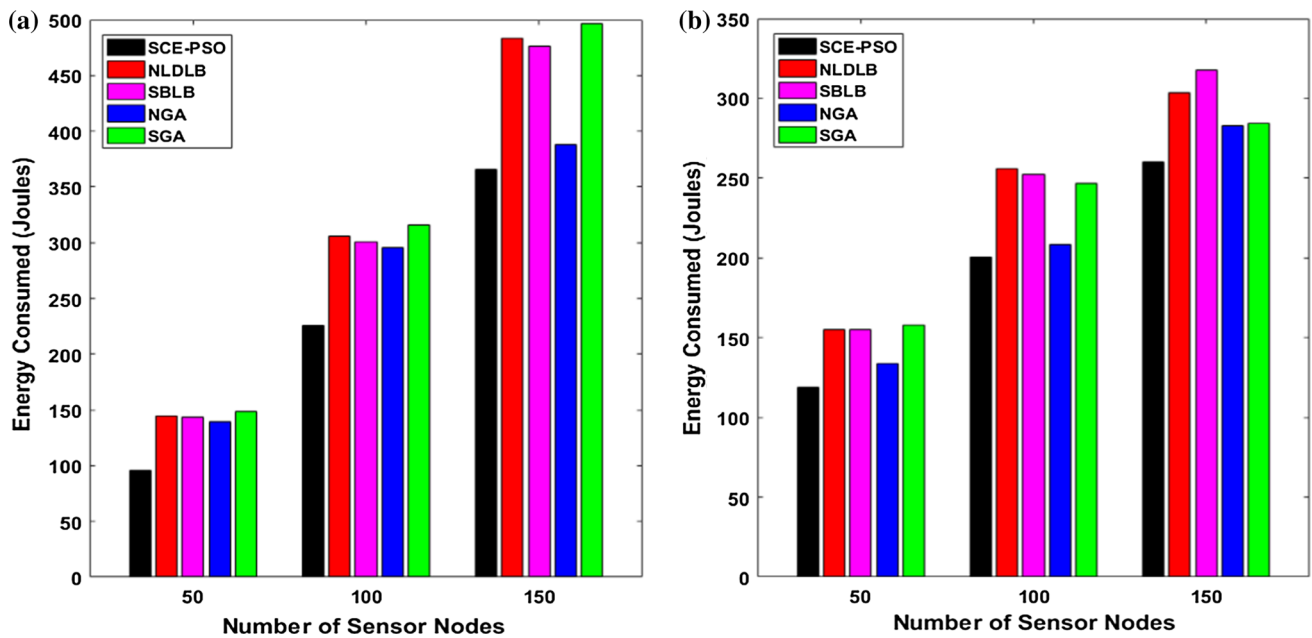


Fig. 9 Comparison of the proposed algorithm with SGA, NGA, SBLB, and NLDLB algorithms in terms of number of sensor nodes and energy consumption for: **a** Equal load and **b** Unequal load

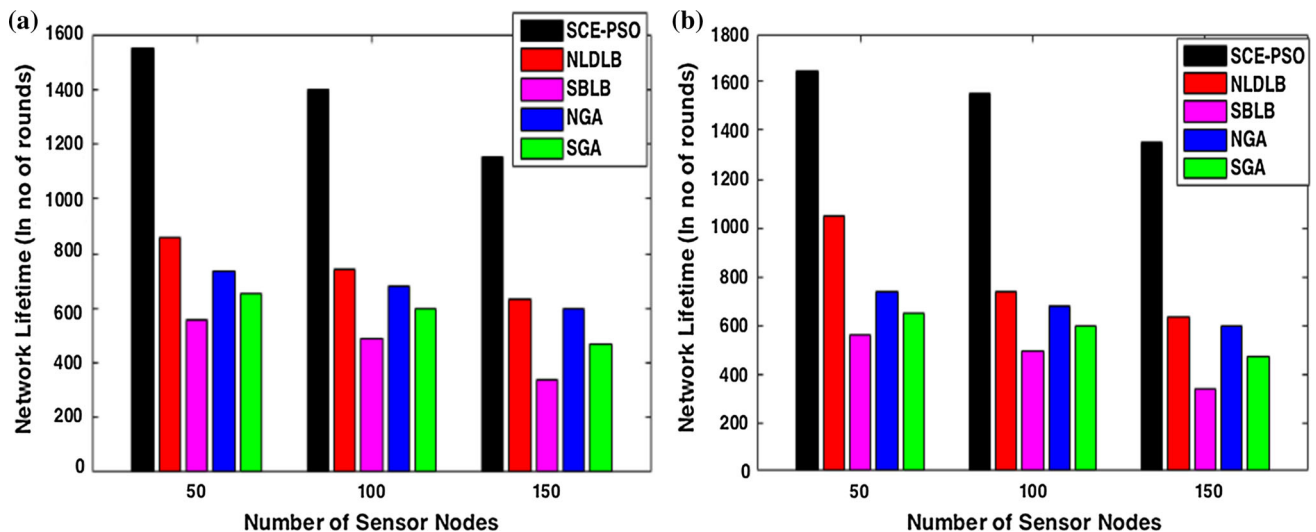


Fig. 10 Comparison of network life time in terms of number of rounds using SGA, NGA, SBLB, NLDLB and proposed SCE-PSO algorithms for: **a** First gateway die, **b** half the gateways die

first gateway dies and half the gateways die in the network are shown in Fig. 10(a), (b) respectively.

It is observed from the figures results that proposed SCE-PSO algorithm have highest first gateway die and half the gateways die values (in terms of number of rounds) as compared to SGA, NGA, SBLB, and NLDLB algorithms. It indicates that the proposed SCE-PSO algorithm does proper load balancing. This in turn prolongs the network lifetime.

7.2.5 First node die and half the nodes die

First node die is a metric measures the life span of first sensor node in the network. It means number of rounds when the first sensor dies. The highest first node die value indicates the longer stability period of network. For comparison purpose SGA, NGA, SBLB, NLDLB and proposed SCE-PSO are experimented with 50, 100 and 150 sensor nodes. These experimental results are shown in Fig. 11.

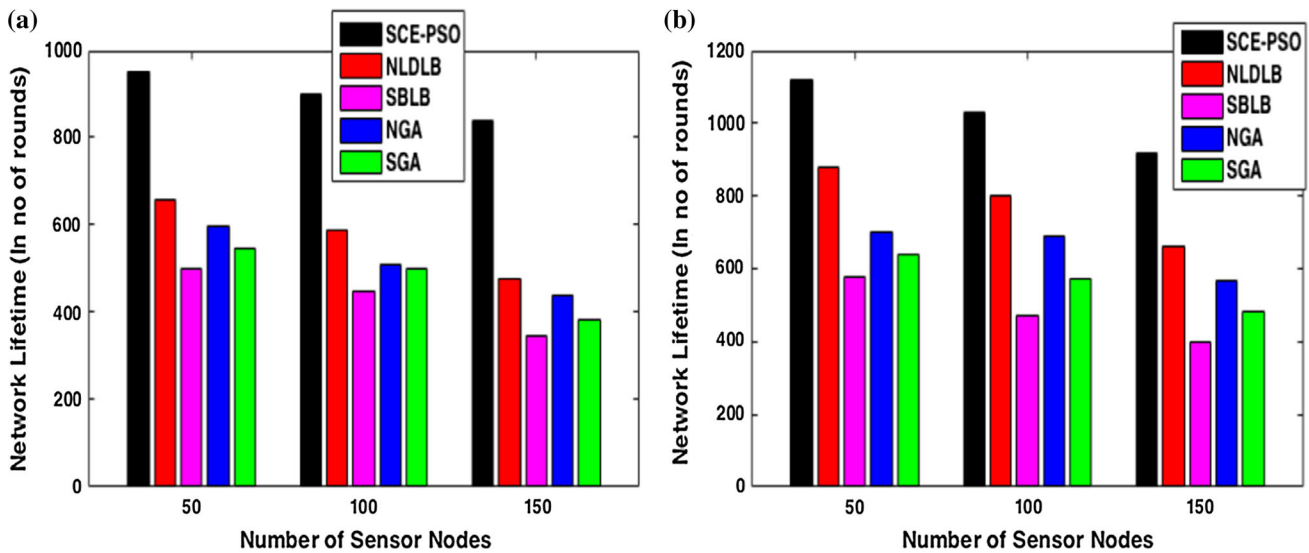


Fig. 11 Comparison of network life time in terms of number of rounds using SGA, NGA, SBLB, NLDLB and proposed SCE-PSO algorithms for: **a** First node die, **b** Half the nodes die

Table 5 Comparison of proposed and state-of-the-art algorithms

Sr. no.	Feature	SGA 2012 [10]	NGA 2013 [14]	NLDLB 2013 [28]	SBLB 2016 [6]	Proposed algorithm
1	Is the algorithm follows evolutionary approach?	Yes	Yes	No	No	Yes
2	Is the algorithm uses clustering approach?	Yes	Yes	Yes	Yes	Yes
3	Is the algorithm uses energy parameter in fitness function?	No	No	Yes	No	Yes
4	Is there variable number of sensor nodes and gateways?	No	No	Yes	Yes	Yes
5	Is there inter-cluster communication?	Yes	Yes	No	Yes	Yes
6	Is there intra-cluster communication?	Yes	Yes	Yes	No	No
	Number of features present	4	4	4	3	5

After how many rounds the first node dies and half the nodes die in the network are shown in Fig. 11(a), (b) respectively.

It is observed from the figures results that proposed SCE-PSO algorithm gives have highest first node die and half the nodes die values (in terms of number of rounds) as compared to SGA, NGA, SBLB, and NLDLB algorithms. It indicates that the proposed SCE-PSO algorithm does proper load balancing. This in turn prolongs the network lifetime.

7.3 Comparative study

A comparison of state-of-the-art load balancing algorithms and proposed algorithm is shown in Table 5. This study considers the six features related to load balancing in WSNs. It clear from the table that the proposed approach

has more distinct features as compared to other state-of-the-art algorithms in the literature.

8 Conclusion

In this paper, SCE-PSO based clustering approach is proposed for effective gateways load balancing in WSNs. A novel fitness function is also proposed by considering load of the gateways, mean cluster distance and number of heavily loaded gateways. The proposed load balancing algorithm has been compared with state-of-the-art SGA, NGA, NLDLB and SBLB algorithms. The experimental results proved that proposed approach outperformed in terms of network lifetime, execution time, energy consumption, first gateway die, half the gateways die, first node die and half the nodes die as compared to state-of-the-art algorithms.

References

- Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8), 102–114.
- Amirthalingam, K., et al. (2016). Improved leach: A modified leach for wireless sensor network. In *IEEE international conference on advances in computer applications (ICACA)* (pp. 255–258). IEEE.
- Dietrich, I., & Dressler, F. (2009). On the lifetime of wireless sensor networks. *ACM Transactions on Sensor Networks (TOSN)*, 5(1), 5.
- Eberhart, R., Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science, 1995. MHS'95* (pp. 39–43). IEEE.
- Edla, D. R., Lipare, A., Cheruku, R., & Kuppili, V. (2017). An efficient load balancing of gateways using improved shuffled frog leaping algorithm and novel fitness function for WSNs. *IEEE Sensors Journal*, 17(20), 6724–6733.
- Gattani, V. S., Jafri, S. H. (2016). Data collection using score based load balancing algorithm in wireless sensor networks. In *International conference on computing technologies and intelligent data engineering (ICCTIDE)* (pp. 1–3). IEEE.
- Gupta, G., Younis, M. (2003). Load-balanced clustering of wireless sensor networks. In *IEEE international conference on communications, 2003. ICC'03* (Vol. 3, pp. 1848–1852). IEEE.
- Heinzelman, W. B. (2000) Application-specific protocol architectures for wireless networks. Ph.D. thesis, Massachusetts Institute of Technology.
- Heinzelman, W. B., Chandrakasan, A. P., & Balakrishnan, H. (2002). An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on wireless communications*, 1(4), 660–670.
- Hussain, S., Matin, A. W., & Islam, O. (2007). Genetic algorithm for hierarchical wireless sensor networks. *JNW*, 2(5), 87–97.
- Jakubcová, M., Máca, P., & Pech, P. (2014). A comparison of selected modifications of the particle swarm optimization algorithm. *Journal of Applied Mathematics*, 2014, 10.
- Jakubcová, M., Máca, P., & Pech, P. (2015). Parameter estimation in rainfall-runoff modelling using distributed versions of particle swarm optimization algorithm. *Mathematical Problems in Engineering*, 2015, 1–13.
- Kennedy, J. (2011). Particle swarm optimization. In C. Sammut & G. I. Webb (Eds.), *Encyclopedia of machine learning* (pp. 760–766). Berlin: Springer.
- Kuila, P., Gupta, S. K., & Jana, P. K. (2013). A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. *Swarm and Evolutionary Computation*, 12, 48–56.
- Kuila, P., & Jana, P. K. (2012). Energy efficient load-balanced clustering algorithm for wireless sensor networks. *Procedia Technology*, 6, 771–777.
- Kuila, P., & Jana, P. K. (2014). Energy efficient clustering and routing algorithms for wireless sensor networks: Particle swarm optimization approach. *Engineering Applications of Artificial Intelligence*, 33, 127–140.
- Kuila, P., & Jana, P. K. (2014). A novel differential evolution based clustering algorithm for wireless sensor networks. *Applied Soft Computing*, 25, 414–425.
- Kumar, N., Kaur, J. (2011). Improved leach protocol for wireless sensor networks. In *2011 7th international conference on wireless communications, networking and mobile computing (WiCOM)* (pp. 1–5). IEEE.
- Lai, C. C., Ting, C. K., Ko, R. S. (2007). An effective genetic algorithm to improve wireless sensor network lifetime for large-scale surveillance applications. In *IEEE congress on evolutionary computation, 2007. CEC 2007* (pp. 3531–3538). IEEE.
- Low, C. P., Fang, C., Ng, J. M., & Ang, Y. H. (2008). Efficient load-balanced clustering algorithms for wireless sensor networks. *Computer Communications*, 31(4), 750–759.
- Wang, N., Zhang, N., & Wang, M. (2006). Wireless sensors in agriculture and food industry-recent development and future perspective. *Computers and electronics in agriculture*, 50(1), 1–14.
- Xiang, W., Wang, N., & Zhou, Y. (2016). An energy-efficient routing algorithm for software-defined wireless sensor networks. *IEEE Sensors Journal*, 16(20), 7393–7400.
- Yan, J., Tiesong, H., Chongchao, H., Xianing, W., Faling, G. (2007). A shuffled complex evolution of particle swarm optimization algorithm. In *International conference on adaptive and natural computing algorithms* (pp. 341–349). Berlin: Springer.
- Yetgin, H., Cheung, K. T. K., El-Hajjar, M., & Hanzo, L. (2015). Cross-layer network lifetime maximization in interference-limited WSNs. *IEEE Transactions on Vehicular Technology*, 64(8), 3795–3803.
- Yetgin, H., Cheung, K. T. K., El-Hajjar, M., & Hanzo, L. (2015). Network-lifetime maximization of wireless sensor networks. *IEEE Access*, 3, 2191–2226.
- Yetgin, H., Cheung, K. T. K., El-Hajjar, M., & Hanzo, L. H. (2017). A survey of network lifetime maximization techniques in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 19(2), 828–854.
- Yu, S., Wang, R., Xu, H., Wan, W., Gao, Y., & Jin, Y. (2011). WSN nodes deployment based on artificial fish school algorithm for Traffic Monitoring System. In *2011 IET international conference on smart and sustainable city (ICSSC 2011)* (pp. 1–5). IEEE.
- Zhang, J., Yang, T. (2013). 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 (EIDWT)* (pp. 273–276). IEEE.
- Zhao, H., Zhang, Q., Zhang, L., Wang, Y. (2015). A novel sensor deployment approach using fruit fly optimization algorithm in wireless sensor networks. In *Trustcom/BigDataSE/ISPA, 2015 IEEE* (Vol. 1, pp. 1292–1297). IEEE.
- Zhou, Y., Wang, N., & Xiang, W. (2017). Clustering hierarchy protocol in wireless sensor networks using an improved PSO algorithm. *IEEE Access*, 5, 2241–2253.



Damodar Reddy Edla is an Assistant Professor and Head of the Computer Science and Engineering Department at National Institute of Technology Goa, India. He received M.Sc. Degree from University of Hyderabad in 2006, M.Tech. and Ph.D. Degree in Computer Science and Engineering from Indian School of Mines Dhanbad in 2009 and 2013 respectively. His research interest includes Data Mining and Wireless Sensor Networks. He

has published more than 40 research papers articles in reputed journals and International conferences. He is senior member of IACSIT. He is also Editorial Board member of several International journals.



Mahesh Chowdary Kongara completed M.Tech. Degree in Department of Computer Science and Engineering at National Institute of Technology Goa, India. He is currently Full-time research scholar at National Institute of Technology Goa.



Ramalingaswamy Cheruku is currently working as an Assistant Professor at Mahindra Ecole Centrale, Hyderabad, India. Prior to this, he was the full-time Ph.D. scholar in CSE Department at National Institute of Technology Goa, India. He received B.Tech. Degree in CSE from JNT University, Kakinada campus in 2008, M.Tech. Degree in CSE from ABV-Indian Institute of Information Technology, Gwalior in 2011. He has served as developer in

Tata Consultancy Services for 2 years. He has also published several papers in reputed journals and conferences.