



A PSO Based Routing with Novel Fitness Function for Improving Lifetime of WSNs

Damodar Reddy Edla¹ · Mahesh Chowdary Kongara¹ · Ramalingaswamy Cheruku¹

Published online: 29 September 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 base station to monitor physical or environmental conditions. In cluster-based WSNs, the cluster head is treated as the gateway. The gateways perform the multiple activities, such as data gathering, aggregation, and transmission etc. The collected data is transmitted from gateways to the base station using routing information. Routing is a key challenge in WSNs design as gateways are constrained by energy, processing power, and memory. Moreover, heavily loaded gateways die in early stages and cause changes in network topology. It is necessary to conserve gateways energy for prolonging the WSNs lifetime. To address this problem, particle swarm optimization (PSO)-based routing is proposed in this paper. Also, a novel fitness function is designed by considering the number of relay nodes, the distance between the gateway to base station and relay load factor of the network. The proposed algorithm is validated under two different scenarios. The experimental results show that the proposed PSO-based routing algorithm prolonged WSNs lifetime when compared to other bio-inspired approaches.

Keywords Wireless sensor networks · Clustering · Particle swarm optimization · Energy efficient routing · Network lifetime

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

✉ 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

as disaster warning systems, health care, environmental monitoring, safety and in crucial areas such as surveillance, intruder detection, defense reconnaissance etc. [1]. 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 consumption of gateways in the network is a challenging task in WSNs. Energy efficient routing algorithms are studied more in this regard [4, 9, 18].

In cluster based WSNs, sensor nodes are divided into several groups called clusters. Each cluster has a cluster head (CH), which gathers data from nearby sensor nodes and processes that data then send to Base Station (BS) directly or via other CHs. In cluster based WSNs, CHs have more workload compared to other sensor nodes [11, 13]. So, many researches [6, 10, 12, 17] deployed high energy level devices as CHs, also known as gateways in WSNs. This entire process is depicted in Fig. 1. In a clustered WSNs, CHs need to do some additional work to collect data from sensor nodes within the respective cluster 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. 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 [17, 19]. Gateways may send data to another gateway or directly to the BS 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 affects 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. This problem is alleviated using an efficient routing between the gateways.

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) [15], which is used to enhance WSNs lifetime in large-scale surveillance applications. Artificial fish schooling algorithm [20], which has been used to solve traffic monitoring system and fruit fly optimization algorithm [21] is applied in Sensor deployment.

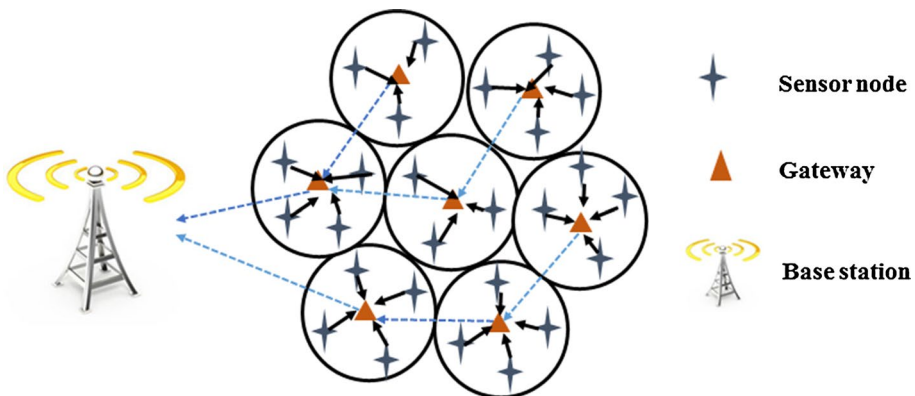


Fig. 1 Cluster based WSN with gateways

In this paper, to address the WSNs energy problem, PSO-based energy efficient routing algorithm is proposed. The rest of the paper is organized as follows: works related to energy efficient routing are discussed in Sect. 2. The required preliminaries are discussed in Sect. 3 to understand the proposed method clearly. The proposed PSO-based energy efficient routing is discussed in Sect. 4. The experimental results and its analysis are presented in Sect. 5. Finally, in Sect. 6, the paper is concluded based on experimental observations.

2 Related Work

Many routing algorithms have been proposed for WSNs. We review some of the routing algorithms in this section. Low-Energy Adaptive Clustering Hierarchy (LEACH) is well-known cluster-based routing 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 fastly inside the network [8]. For solving this problem many improved LEACH algorithms have been proposed for selecting optimal CHs. In [14] authors have proposed I-LEACH (Improved 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.

GAs [4] have been implemented for routing task between gateways and base station. In this algorithm fitness value evaluates the quality of the network, better fitness value gives the better routing path. P. Kuila et al. have used PSO-based routing to enhance the lifetime of WSNs. In this method, routing takes place based on the distance between the gateways and number of relay nodes between the each gateway to the base station. Fitness value of each particle in the swarm is computed using multi-objective fitness function and this fitness value judges the quality of the network. A particle with better fitness function value gives better routing path for the network [12]. However, this proposed multi-objective fitness function not considered the relay load factor which also plays vital role in prolonging the network lifetime. If the load on particular relay nodes is more, network lifetime is decreased. Hence, in this paper authors have designed a novel fitness function by considering the relay load factor of the network.

3 Preliminaries

3.1 Energy Model

We have used similar energy model used in [7]. Energy consumption of sensor nodes and gateways effect the WSNs lifetime. Hence, it is important to design an efficient routing 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 gateways is indicated by $\lambda = \{g_1, g_2, \dots, g_N\}$ and base station in the network is denoted by B_s .
2. $Lifetime(g_i)$ signifies the gateway (g_i) lifetime. If g_i has leftover energy $E_l(g_i)$ and energy consumption per round $E_c(g_i)$ then $Lifetime(g_i)$ is computed as follows:

$$Lifetime(g_i) = \left\lfloor \frac{E_l(g_i)}{E_c(g_i)} \right\rfloor$$

3. d_{max} indicates the each gateway maximum communication distance.
4. $distance(g_i, g_k)$ indicates the distance between two gateways g_i and g_k .
5. $range(g_i)$ denotes the number of gateways within communication range of g_i (sometimes base station (B_s) also a member of $range(g_i)$). In other terms it is denoted as follows:

$$range(g_i) = \{g_k | \forall g_k \in (\lambda + B_s) \wedge distance(g_i, g_k) \leq d_{max}\} \quad (3)$$

6. Relay node: relay node is one of the gateway in the network. All the gateways in the network may not be directly communicate with the base station (B_s), in that situation gateways send data through one or more intermediate gateways, these intermediate nodes are called relay nodes.
7. $RelayNodes(g_i)$: It is a list contains the set of gateways, that can be selected as a relay node for g_i to communicate with base station B_s .

$$RelayNodes(g_i) = \{g_k | \forall g_k \in (range(g_i) - B_s)\} \quad (4)$$

8. $RelayNode(g_i)$: It is one of the gateway selected as relay node from g_i to BS. If g_i is within the communication range of B_s , in that case directly we can communicate with B_s .
9. $RelayNodeCount(g_i)$: Denotes the number of relay nodes needed to reach from g_i to the B_s . If g_i is directly communicates with base station then $RelayNodeCount(g_i)$ is one.

$$RelayNodeCount(g_i) = \begin{cases} 1, & \text{If } RelayNode(g_i) = B_s \\ 1 + RelayNodeCount(g_k), & \\ & \text{If } RelayNode(g_i) = g_k \end{cases} \tag{5}$$

10. *DelayTime(g_i)*: is the time needed to transmit collected data from gateway (*g_i*) to the base station (*B_s*) [3]. It contains average queuing delay (*DT_q*) value per intermediate data dissemination, transmission delay (*DT_t*) and propagation delay (*DT_p*).

$$DelayTime(g_i) = (DT_q + DT_t + DT_p) * RelayNodeCount(g_i) \tag{6}$$

i.e.,

$$DelayTime(g_i) = C * RelayNodeCount(g_i) \tag{7}$$

where $C = DT_q + DT_t + DT_p$, is constant for a particular network [3]. It is also clear from Eq. 7 that *DelayTime(g_i)* is directly proportional to *RelayNodeCount(g_i)* i.e., reducing *RelayNodeCount* reduces delay time.

11. Total distance covered among any two gateways in the routing path is defined by *MaxDistance*

$$MaxDistance = \max\{distance(g_i, RelayNode(g_i)) | \forall i, 1 \leq i \leq N, g_i \in \lambda\} \tag{8}$$

12. Maximum relay node count of the gateway is defined as *MaxRelayCount*. It is formulated as follows:

$$MaxRelayCount = \max\{RelayNodeCount(g_i) | \forall i, 1 \leq i \leq N, g_i \in \lambda\} \tag{9}$$

Note: $\{distance(g_i, g_k) * 1 \leq d_{max}, 1 \leq i \leq N, 1 \leq k \leq \{\lambda + B_s\}\}$

3.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is inspired from the nature based on fish schooling and bird flocking [5, 16]. 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.

PSO consists of predefined number of particles (*S_n*), each particle 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 (*P_{id}*) and velocity (*Vel_{id}*) in *d*th dimension of the hyperspace. So, at any point in time, particle *P_i* is represented as

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

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 iteratively. Using the Eqs. (10) and (11) [12], we perform the recursive position (*P_{id}*) and velocity (*Vel_{id}*) updates.

$$Vel_{i,d}(t + 1) = w * Vel_{i,d}(t) + a1 * r1 * (PLbest_{i,d} - P_{i,d}(t)) + a2 * r2 * (Gbest_d - P_{i,d}(t)) \tag{10}$$

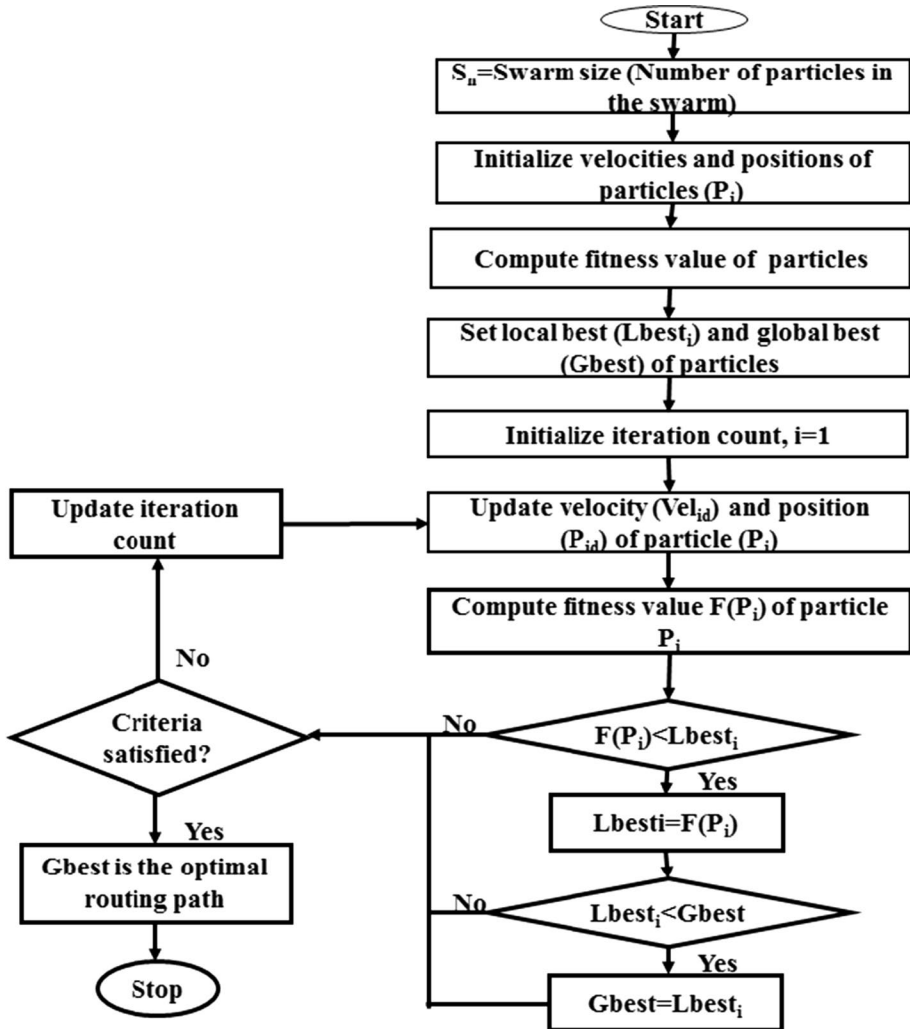


Fig. 2 PSO flowchart

$$P_{i,d}(t+1) = P_{i,d}(t) + Vel_{i,d}(t) \quad (11)$$

where w is the inertial weight, $a1$ and $a2$ are the acceleration constants which are non-negative real numbers. $r1$ and $r2$ 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. 2.

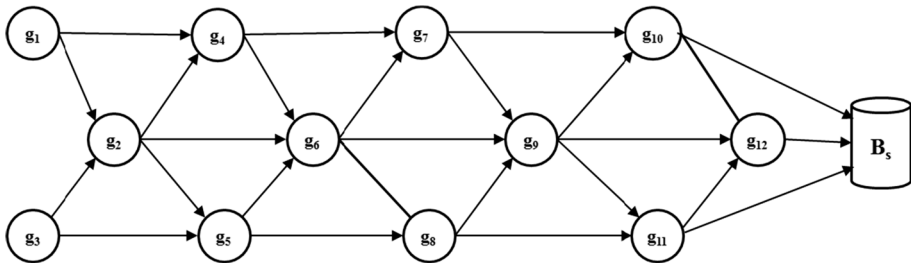


Fig. 3 A WSN topology with 12 gateways and one base station

Table 1 Possible relay nodes of gateways

Gateways	RelayNodes(g_d)	$ RelayNodes(g_d) $
g_1	$\{g_2, g_4\}$	2
g_2	$\{g_4, g_5, g_6\}$	3
g_3	$\{g_2, g_5\}$	2
g_4	$\{g_6, g_7\}$	2
g_5	$\{g_6, g_8\}$	2
g_6	$\{g_7, g_8, g_9\}$	3
g_7	$\{g_9, g_{11}\}$	2
g_8	$\{g_9, g_{10}\}$	2
g_9	$\{g_{10}, g_{11}, g_{12}\}$	3
g_{10}	$\{g_{12}, B_s\}$	2
g_{11}	$\{g_{12}, B_s\}$	2
g_{12}	$\{B_s\}$	1

4 Proposed PSO-Based Routing Algorithm

We have formulated PSO-based routing algorithm for WSNs. This algorithm consists of particle initialization, evaluation of fitness function and updating velocity and position phases.

4.1 Particle Initialization Phase

Each particle is encoded as network (contains path from each gateway (g_i) to B_s). The dimension of each particle P_i is D (number of CHs or gateways). We initialize each gateway ($P_{i,d} | 1 \leq i \leq Sn, 1 \leq d \leq D$) with randomly generated number from a uniform distribution in the range (0, 1]. The value of d th gateway (i.e., $P_{i,d}$ assigns a new node (say g_k) to g_d as an immediate neighbor towards B_s . That is, g_d sends data to g_k which sends data to B_s . This can be formulated as follows:

$$n = \text{ceil}(|p_{i,d} * RelayNodes(g_d)|) \tag{12}$$

i.e., g_k is the n th relay node in $RelayNodes(g_d)$.

Table 2 Relay node selection from randomly created particle

Gateways	$P_{i,d}$	$ RelayNodes(g_d) $	$ceil(P_{i,d} * RelayNodes(g_d))$	$RelayNode(g_d)$
g_1	0.43	2	1	g_2
g_2	0.65	3	2	g_5
g_3	0.28	2	1	g_2
g_4	0.32	2	1	g_6
g_5	0.68	2	2	g_8
g_6	0.98	3	3	g_9
g_7	0.22	2	1	g_9
g_8	0.72	2	2	g_{10}
g_9	0.57	3	2	g_{11}
g_{10}	0.16	2	1	g_{12}
g_{11}	0.35	2	1	g_{12}
g_{12}	0.19	1	1	B_s

The particle initialization process is explained using following Example 1.

Example 1 Consider a network with 12 gateways $\{g_1, g_2, g_3, \dots, g_{12}\}$ as shown in Fig. 3. The same is represented using Table 1 in terms of gateways, relay nodes in the range and number of relay nodes in the range. Figure 3 shows a directed acyclic graph $G(V, E)$. Where V is set of gateways and E is set of edges. The edge between g_i and g_j specifies that g_i can send data to B_s via g_j (g_j should be in communication range of g_i). It is clear from Fig. 3 that g_2 can use any of the gateways from $\{g_4, g_5, g_6\}$ as next gateway to send data to B_s .

During initialization, each dimension (gateway) $P_{i,d}$ 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 gateway position values constitute a particle or solution. This solution is generated as follows:

Let us consider gateway g_2 and its assigned random value of 0.65. According to the Eq. (12), $ceil(0.65, 3) = 2$, i.e., second gateway (g_5) from the $RelayNodes(g_2)$ is selected for data transmission to B_s . This process is repeated for all gateways to construct entire network (particle). Table 2 shows the final result of network construction, and the same is visualized using Fig. 4.

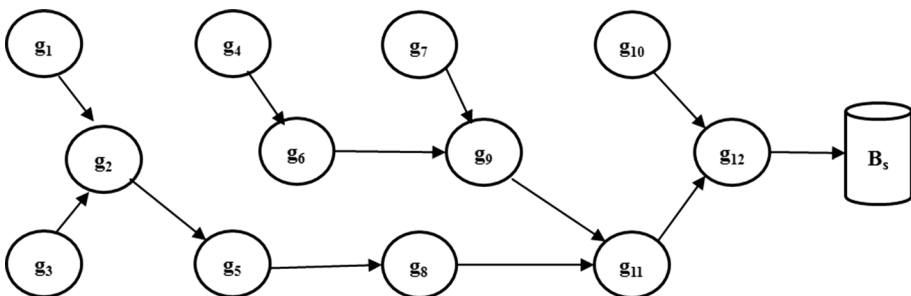


Fig. 4 Representation of random particle

4.2 Proposed Fitness Function

In order to evaluate the goodness of generated particles (networks) a novel fitness function is proposed. The proposed fitness function is given in Eq. (13). It is clear from the equation that the proposed fitness function is constituted by considering the three objectives. The first objective is to minimize the maximum distance between gateways to the base station. The second objective is to minimize the number of relay nodes used by the gateway (between gateway to base station) for data transmission. The third objective is to minimize the relay load factor of the network.

$$F = \alpha * objective1 + \beta * objective2 + \gamma * objective3 \tag{13}$$

$$objective1 : minimize \{distance(g_i, RelayNodes(g_i)), 1 \leq i \leq N\} \tag{14}$$

$$objective2 : minimize \{RelayNodeCount(g_i)\}, 1 \leq i \leq N. \tag{15}$$

$$objective3 : minimize \{RelayLoadFactor\} \tag{16}$$

RelayLoadFactor = sum of all relay nodes *RelayLoad* value whose *RelayLoad* value is above the *AvgRelayLoad* value. i.e.,

$$RelayLoadFactor = \sum_{i=1}^N \{RelayLoad(g_i) | RelayLoad(g_i) \geq AvgRelayLoad\} \tag{17}$$

RelayLoad(g_i) = number of gateways sending data through *g_i* either directly or indirectly. *RelayLoad(g_i)* is zero, if gateway *g_i* is not participated as relay node.

$$AvgRelayLoad = \left[\frac{\sum_{i=1}^N \{RelayLoad(g_i)\}}{Number\ of\ relay\ nodes} \right] \tag{18}$$

where α , β and γ are weighted importance of each objective such that $\alpha + \beta + \gamma = 1$ and $0 < \alpha, \beta, \gamma \leq 1$.

For a better solution the fitness function needs to be minimized i.e., lower fitness value indicates best network.

The *RelayLoadFactor* calculation is explained using Example 2.

Example 2 According to Fig. 4 the relay nodes are {*g₂*, *g₅*, *g₆*, *g₈*, *g₉*, *g₁₁*, *g₁₂*}.

The *RelayLoad* values of {*g₂*, *g₅*, *g₆*, *g₈*, *g₉*, *g₁₁*, *g₁₂*} are {2, 3, 1, 4, 3, 9, 11} respectively.

Then $AvgRelayLoad = \lfloor 33/7 \rfloor = 4$ and

$RelayLoadFactor = 4 + 9 + 11 = 24$.

Note that while achieving any one of the above objective, we may loose control over others. We should include all three objectives into single fitness function and minimize it. To overcome the problem of trade-offs among objectives we have used weighted sum approach to formulate proposed fitness function by considering all objectives.

Algorithm 1 PSO-based routing

- **Input**
 - A set of gateways $\lambda = \{g_1, g_2, \dots, g_N\}$
 - $range(g_i)$: collection of all gateways which are within the communication range of g_i .
 - $RelayNodes(g_i)$ and $RelayNodeCount(g_i), \forall i, 1 \leq i \leq N$
 - Swarm size = S_n (Number of particles in the swarm)
 - **Output**
 - Optimal routing path from each gateway to base station in the given network
 - Routing R: $\lambda \rightarrow \{\lambda + B_s\}$
 - **Algorithm**
 - 1: **Step 1:**
 - 2: Initialize particles (P_i) in the swarm (S_n), $\forall i, 1 \leq i \leq S_n$
 - 3: **Step 2:**
 - 4: **for** $i = 1$ to S_n **do**
 - 5: Compute fitness value of P_i , i.e., $F(P_i) = \alpha * objective1 + \beta * objective2 + \gamma * objective3$
 - 6: $Lbest_i \leftarrow P_i$
 - 7: **end for**
 - 8: **Step 3:**
 - 9: $Gbest = \{Lbest_x \mid \text{Fitness}(Lbest_x) = \text{minimum}(\text{fitness}(Lbest_i), \forall i, 1 \leq i \leq S_n)\}$
 - 9: **Step 4:**
 - 10: **while** (!*terminate*) **do**
 - 11: **for** $i = 1$ to S_n **do**
 - 12: Update position and velocity of particle P_i using equation
 - 13: Compute fitness value of P_i
 - 14: **if** $F(P_i) < F(Lbest_i)$ **then**
 - 15: $Lbest_i \leftarrow P_i$
 - 16: **end if**
 - 17: **if** $F(Lbest_i) < F(Gbest)$ **then**
 - 18: $Gbest \leftarrow Lbest_i$
 - 19: **end if**
 - 20: **end for**
 - 21: **end while**
 - 22: **Step 5:**
 - 23: Compute RelayNode (g_i), $\forall i, 1 \leq i \leq N$ using $Gbest$ i.e., routing path for the network.
 - 24: **Step 6:**
 - 25: Stop
-

4.3 Position and Velocity Updating Phase

In every iteration, the position and velocity of each particle are updated using Eqs. (10) and (11). 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. (12), 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 [12] 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 (number of iterations), the velocity and the position values are updated in iterative manner. After termination of the PSO-based routing algorithm, the endmost solution is represented by the $Gbest$.

The proposed PSO-based routing algorithm that comprises of above mentioned phases is explained in Algorithm 1.

5 Results and Discussion

We have implemented the proposed method using Matlab 2015Ra tool and C++ language. All the experiments are carried out under two different WSN scenarios namely WSN1 and WSN2. WSN1 and WSN2 consists of 60 to 80 number of gateways and 100 to 400 number of sensor nodes. WSN1 and WSN2 differ only in network structure (topology) and base station placements. The base station positions for WSN1 and WSN2 are (200, 200) and (350, 150) respectively. Table 3 shows the common simulation parameters for both WSN1 and WSN2. For comparison purpose GA [4], Greedy Load Balancing Clustering Algorithm (GLBCA) [17] and PSO [12] based routing algorithms in the literature have been implemented. In order to validate performance of the proposed method, it is compared with GA, GLBCA, PSO

Table 3 Simulation parameters

Parameters	Value
<i>WSN parameters</i>	
Area	500 * 500 m ²
Sensor nodes	100–400
Capacity of each sensor node	2.0 J
Gateways	60–80
Capacity of each gateway	8.0 J
Iterations	200
Communication range	120 m
E_{elec}	50 nJ/bit
ϵ_{fs}	10 pJ/bit/m ²
ϵ_{mp}	0.0013 pJ/bit/m ⁴
d_0	87.0 m
E_{DA}	5 nj/bit
Packet size	4000 bits
Message size	500 bits
<i>PSO parameters</i>	
S_n	60
a_1	1.4962
a_2	1.4962
W	0.7968
V_{max}	0.5
V_{min}	– 0.5

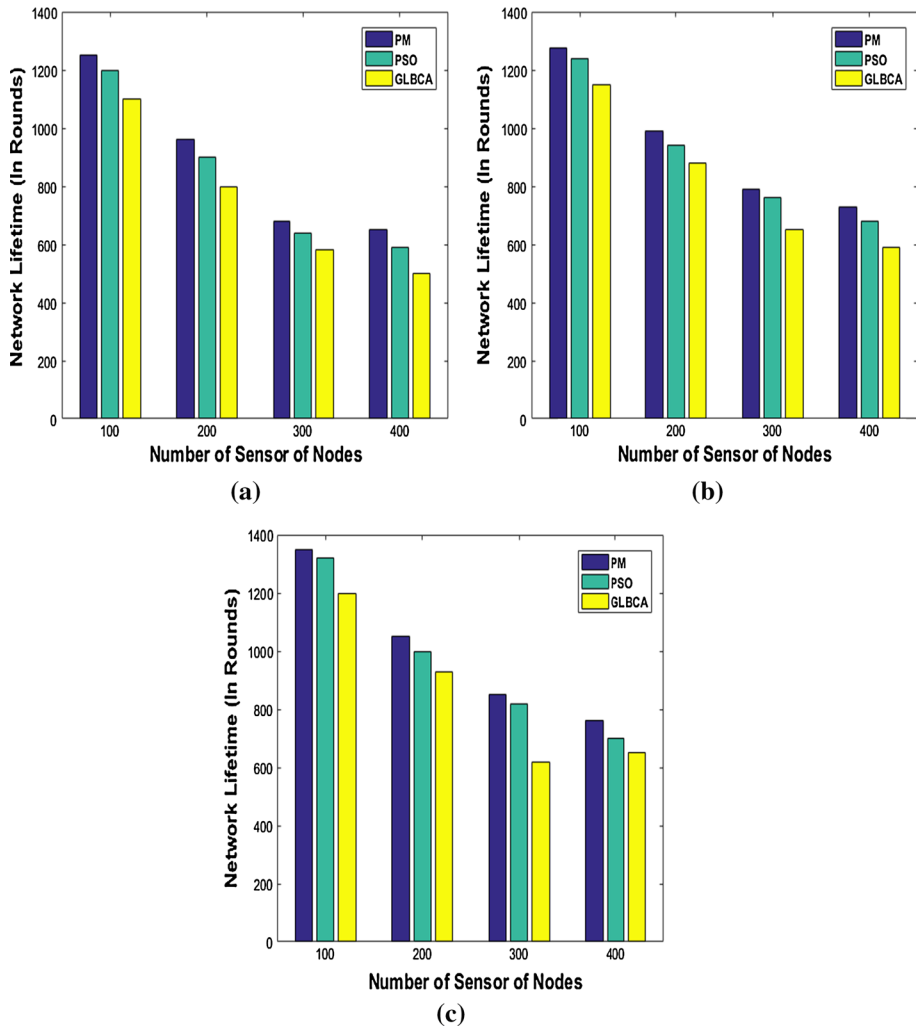


Fig. 5 Comparison of network life time for WSN#1. **a** Comparison of lifetime with 60 gateways. **b** Comparison of lifetime with 70 gateways. **c** Comparison of lifetime with 80 gateways

based routing algorithms in terms of network lifetime, total number of hops and average relay load on network.

5.1 Number of Gateways Versus Network Lifetime

Experiments are performed for number of sensor nodes 100, 200, 300 and 400. For both WSN1 and WSN2 lifetime of networks are calculated in terms of number of rounds (how many number of rounds the network is alive). Experimental results for WSN#1 are shown in Fig. 5 for 60, 70 and 80 number of gateways. Similarly WSN#2 results for 60, 70 and 80 number of gateways are shown in Fig. 6. It is observed from the results that using proposed method the life time of WSN1 and WSN2 is prolonged as compared to state-of-the-art PSO

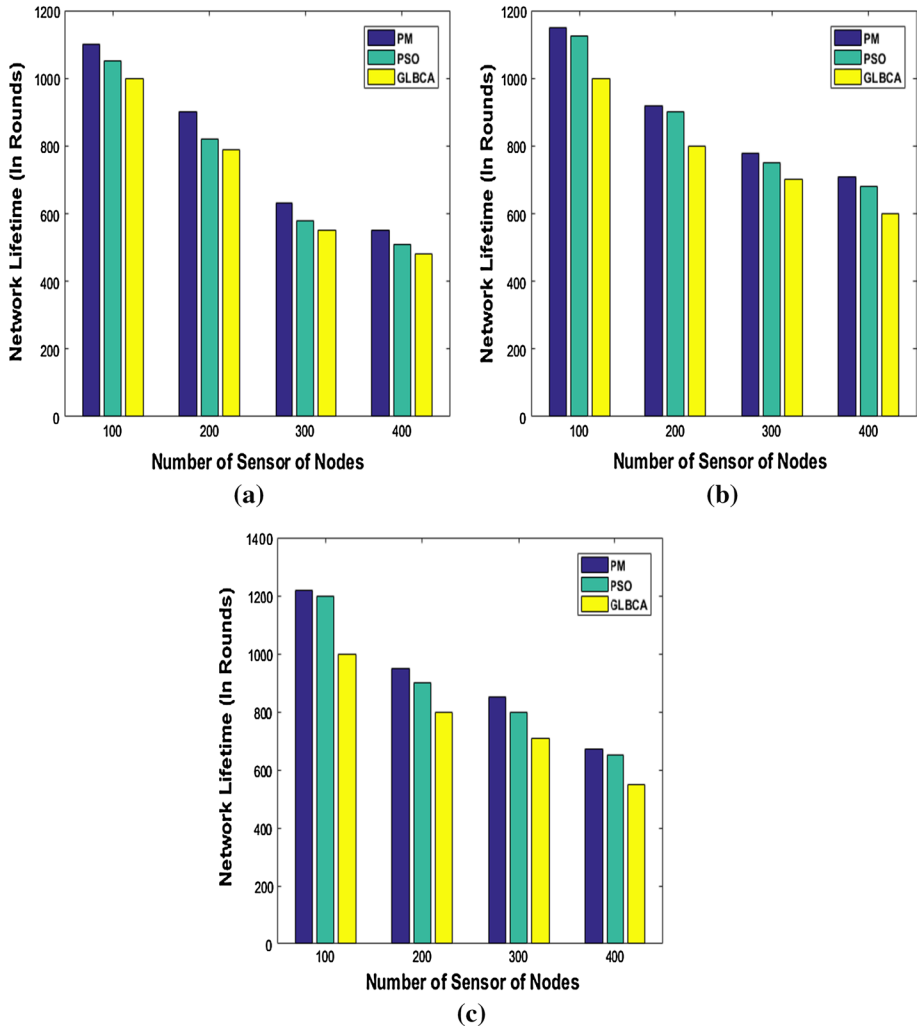


Fig. 6 Comparison of network life time for WSN#2. **a** Comparison of lifetime with 60 gateways. **b** Comparison of lifetime with 70 gateways. **c** Comparison of lifetime with 80 gateways

and GLBCA techniques. It is due to the fact that load on particular relay node is balanced by redirecting through alternative routing path. This process some times may choose a longer routing path as an alternative. Though, it increases number of hops but overall load distribution is proper. Hence, it will increase the network life time.

5.2 Number of Gateways Versus Total Number of Hops

Experiments are performed for number of gateways 60, 70 and 80. For both WSN1 and WSN2 number of hops are calculated in terms of communication between gateways (how many

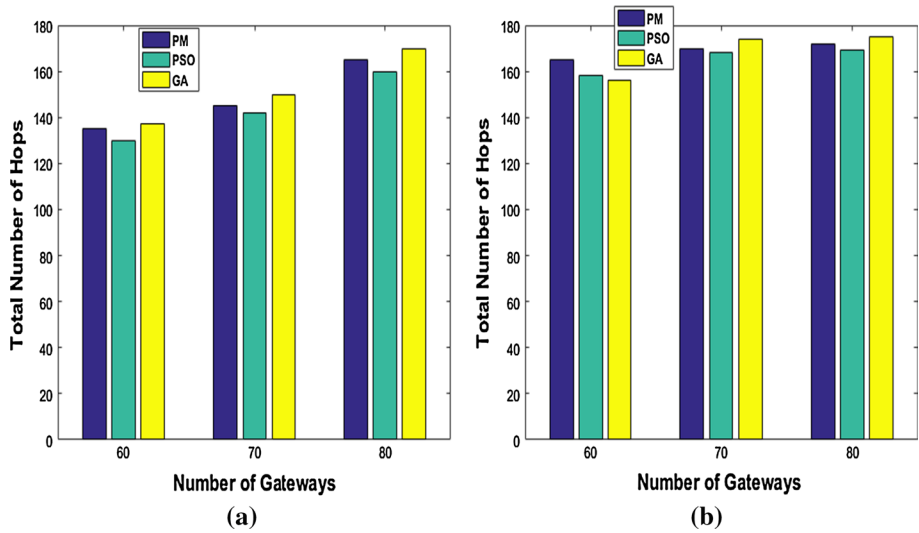


Fig. 7 Comparative analysis in terms of number of hops Vs number of gateways. **a** WSN#1. **b** WSN#2

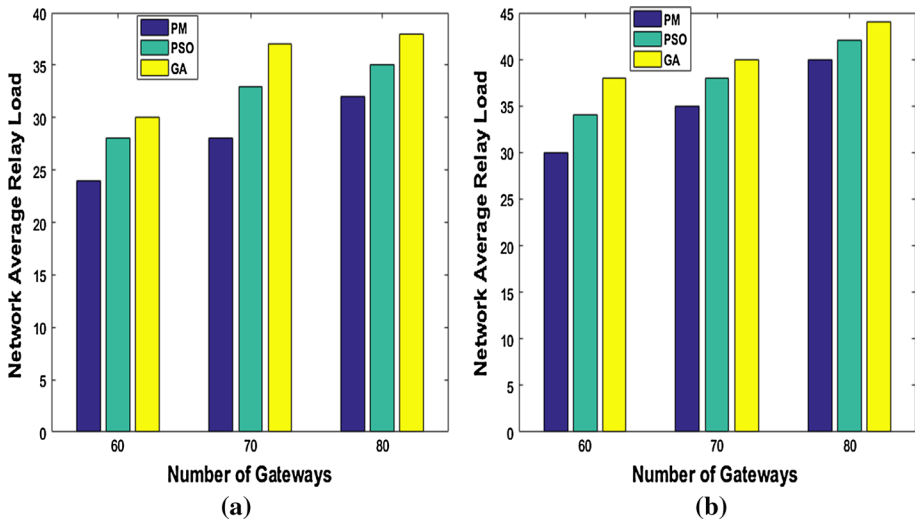


Fig. 8 Comparative analysis in terms of number of gateways Vs average relay load. **a** WSN#1. **b** WSN#2

number of hops required for each gateway to reach base station). Experimental results for WSN#1 are shown in Fig. 7a for 60, 70 and 80 number of gateways. Similarly, WSN#2 results for 60, 70 and 80 number of gateways are shown in Fig. 7b. It is observed from the results the proposed method some times requires more number of hops for routing when compared to state-of-the-art GA [4] and PSO techniques. It is due to the reason that if any relay node is

heavily loaded then the proposed method chooses another alternative routing path which may be longer. Though, proposed method some times chooses longer routing path this helps in prolonging the network life time.

5.3 Number of Gateways Versus Average Relay Load

Experiments are performed for number of gateways 60, 70 and 80. For both WSN#1 and WSN2 average relay load on network is calculated as mean of all relay loads in the network (using Eq. 18). These experimental results for WSN#1 are shown in Fig. 8a for 60, 70 and 80 number of gateways. Similarly, WSN#2 results for 60, 70 and 80 number of gateways are shown in Fig. 8b. It is observed from the results the proposed method requires less average relay load when compared to state-of-the-art GA and PSO techniques. It is due to considering relay load factor in proposed fitness function.

6 Conclusion

In this paper, particle swarm optimization algorithm has been used for effective routing in WSNs for prolonging the WSNs lifetime. For effective routing a novel fitness function also designed by considering relay load factor of the network along with distance and number of relay nodes between gateways to base station. The proposed PSO-based routing algorithm has been compared with state-of-the-art routing techniques, such as GA, GLBCA, etc. Performance of the proposed algorithm is evaluated in terms of network life time, number of hops and average relay load under two different typologies of WSN. The experimental results showed that proposed PSO-based routing algorithm increased the network life time and reduced the average relay load at the cost slightly more number of hops when compared with GA [4] and GLBCA [17] routing techniques. Moreover, our proposed PSO-based routing algorithm with novel fitness function is outperformed PSO-based algorithm [12] in the literature.

Finally, it is concluded from above results that proposed routing algorithm is efficient for improving the lifetime of WSNs.

References

1. Akyildiz, I. F., Su, W., Sankarasubramaniam, Y., & Cayirci, E. (2002). A survey on sensor networks. *IEEE Communications Magazine*, 40(8), 102–114.
2. 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 (2016).
3. Ammari, H. M., & Das, S. K. (2008). A trade-off between energy and delay in data dissemination for wireless sensor networks using transmission range slicing. *Computer Communications*, 31(9), 1687–1704.
4. Bari, A., Wazed, S., Jaekel, A., & Bandyopadhyay, S. (2009). A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks. *Ad Hoc Networks*, 7(4), 665–676.
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. MHS'95* (pp. 39–43). IEEE.
6. Gupta, G., & Younis, M. (2003). Load-balanced clustering of wireless sensor networks. In *IEEE international conference on communications. ICC'03* (Vol. 3, pp. 1848–1852). IEEE.

7. Heinzelman, W. B. (2000). Application-specific protocol architectures for wireless networks. Ph.D. thesis, Massachusetts Institute of Technology.
8. 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.
9. Kaur, R., & Singh, K. P. (2015). An efficient multipath dynamic routing protocol for mobile wsns. *Procedia Computer Science*, 46, 1032–1040.
10. Kuila, P., & Jana, P. K. (2011). Improved load balanced clustering algorithm for wireless sensor networks. In *International conference on advanced computing, networking and security* (pp. 399–404). Springer.
11. Kuila, P., & Jana, P. K. (2012). Energy efficient load-balanced clustering algorithm for wireless sensor networks. *Procedia Technology*, 6, 771–777.
12. 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.
13. Kuila, P., & Jana, P. K. (2014). A novel differential evolution based clustering algorithm for wireless sensor networks. *Applied Soft Computing*, 25, 414–425.
14. 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.
15. 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. CEC 2007* (pp. 3531–3538). IEEE.
16. Lim, W. H., & Isa, N. A. M. (2014). Particle swarm optimization with increasing topology connectivity. *Engineering Applications of Artificial Intelligence*, 27, 80–102.
17. 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.
18. Mann, P. S., Singh, S., & Kumar, A. (2016). Computational intelligence based metaheuristic for energy-efficient routing in wireless sensor networks. In 2016 IEEE congress on evolutionary computation (CEC) (pp. 4460–4467). IEEE.
19. Tang, J., Hao, B., & Sen, A. (2006). Relay node placement in large scale wireless sensor networks. *Computer Communications*, 29(4), 490–501.
20. 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 *IET international conference on smart and sustainable city. ICSSC 2011, Shanghai, China* (pp. 201–205).
21. Zhao, H., Zhang, Q., Zhang, L., & Wang, Y. (2015). A novel sensor deployment approach using fruit fly optimization algorithm in wireless sensor networks. In *2015 IEEE Trustcom/BigDataSE/ISPA* (Vol. 1, pp. 1292–1297). IEEE.

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Damodar Reddy Edla is an Assistant Professor and Head of the Computer Science and Engineering department in 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 is Data Mining, Wireless Sensor Networks. He has published more than 20 research papers articles in reputed journal and International conferences. He is senior member of IACSIT. He is also Editorial Board member of several International journals.



Mahesh Chowdary Kongara is recently completed M.Tech. Degree in Department of Computer Science and Engineering from National Institute of Technology Goa, India.



Ramalingaswamy Cheruku is presently working as full-time research scholar in department of Computer Science and Engineering at National Institute of Technology Goa. He received B.Tech. Degree in Computer Science and Engineering from JNT University, Kakinada campus in 2008, M.Tech. Degree in Computer Science and Engineering 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.