Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 3113 – 3131
© TÜBİTAK
doi:10.3906/elk-1606-206

An intelligent PSO-based energy efficient load balancing multipath technique in wireless sensor networks

Sukhchandan RANDHAWA*, **Sushma JAIN**
Computer Science & Engineering Department, Thapar University, Patiala, India Punjab, India

Received: 13.06.2016 • **Accepted/Published Online:** 16.12.2016 • **Final Version:** 30.07.2017

Abstract: To provide a reliable and efficient service, load balancing plays an important role in wireless sensor networks (WSNs). There is a need to maximize the network lifetime for WSNs applications with periodic generation of data. Due to the relationship between energy consumption and network sensor node lifetime, energy consumption in a network should be minimized and balanced in order to increase network lifetime. Energy-efficient load-balancing techniques are needed to solve this problem. In this paper, a particle swarm optimization (PSO)-based energy-efficient load-balancing technique is proposed, in which the required number of routing paths and energy consumption of different nodes and paths are calculated. Based on maximum residual energy, paths are selected and further PSO-based load balancing is performed among all the paths for data transfer at a particular point of time. The performance of the proposed technique is evaluated using real testbed and experimental results and shows that the proposed technique performs better than existing techniques in terms of network lifetime, energy consumption, throughput, number of alive nodes, number of data packets received, execution time, and convergence rate.

Key words: Wireless sensor networks, load balancing, energy efficiency, network lifetime, clustering, particle swarm optimization

1. Introduction

A wireless sensor network (WSN) is an isle of a large number of sensing nodes deployed in a region of interest for specific applications. They are usually small and have computational, communicational, and environment-sensing capabilities. However, these nodes have limited resources such as bandwidth, power, memory, processing resources, and network lifetime [1].

Limited energy availability in WSNs is an unavoidable design problem, as charging batteries in WSNs is not viable. WSN network lifetime is an important performance criterion and it depends on the energy consumption of sensors [2]. Load balancing is tremendously vital for network lifetime, as scarce energy is consumed rapidly if all of the traffic is redirected towards a single path [3,4].

Therefore, a particle swarm optimization (PSO)-based energy-efficient load-balancing (PSO-EELB) technique is proposed, in which the required number of routing paths and energy consumption of different paths are calculated. Paths are selected based on optimal residual energy, and further PSO-based load balancing is performed for data transfer.

This paper outlines an energy-efficient load-balancing technique for the efficient transfer of data that considers energy consumption as an optimization parameter. The main contributions of this research are: (i) to

*Correspondence: sukhchandan@thapar.edu

propose an energy-efficient load-balancing technique in which the required number of routing paths and energy consumption of different paths are calculated, (ii) to select different routing paths based on residual energy, with further PSO-based load balancing performed for data transfer, (iii) to optimize network lifetime, throughput, number of alive nodes, number of data packets received, execution time, and energy consumption and to perform operations in optimal execution time, (iii) to implement and perform evaluation using real testbed. Therefore, a PSO-EELB technique is proposed.

The rest of the paper is organized as follows: Section 2 presents related work on existing load-balancing techniques. Section 3 presents existing energy-efficient load-balancing techniques. In Section 4, a PSO-EELB technique is proposed. Section 5 presents the experimental setup used for performance evaluation and results. Section 6 presents conclusions and future scope.

2. Related work

A number of load-balancing algorithms have been proposed by researchers for WSNs. A review of such works based on heuristic and metaheuristic approaches is presented here. The main emphasis is on the metaheuristic approaches, as the proposed PSO-EELB algorithm is based on a metaheuristic approach. There are chances of failure or delay if the same coordinator receives data from a large number of nodes simultaneously, due to regular variations in large-scale topology without effective load balancing [5,6]. By distributing the traffic uniformly across the network, the tradeoff between power consumption and communication efficiency can be solved [7].

Swarm intelligence (SI)-based routing protocols, along with their applications and research issues related to the application of scientific methodological analysis, have been explored in [8]. Further, Muhammad et al. identified important features of routing protocols, such as minimal computational and memory requirements, autonomicity, energy efficiency, and in-network data aggregation. A general model for SI-based routing is discussed.

Kuila and Jana presented a min-heap–based energy-efficient load-balanced clustering algorithm [9] that focuses on load balancing and energy efficiency based on clusters' cardinality. The number of nodes allotted to the cluster head (CH) is used to construct a min-heap. In the second proposed approach, i.e. parameter-based clustering algorithm, the communication load of the CHs is incorporated with respect to the base station (BS). An energy-efficient fault-tolerant clustering and routing algorithm has been proposed in [10], in which distributed run-time recovery of the nodes is used to handle sudden failure of the CHs. The proposed clustering algorithm runs on every node simultaneously. CH forms a time division multiple access (TDMA) schedule after cluster formation for member nodes. It balances energy consumption of the CHs.

A clustering algorithm based on differential evolution has been proposed in [11] to improve network lifetime by avoiding quicker failure of highly loaded CHs. An efficient vector-encoding scheme is used to derive the fitness function for prolonging network lifetime. The generation of the initial population is restricted by considering the connectivity between nodes and their CHs. Based on the genetic algorithm (GA) approach, only children chromosomes that balance the load are generated, and hence the proposed algorithm converges faster than a basic GA.

PSO-based linear/nonlinear programming formulations have been presented for energy-efficient clustering and routing, in which multiobjective fitness function and an efficient particle-encoding scheme are used [12]. A trade-off between number of relays and transmission distance for the particle-encoding scheme, along with a routing solution, is also presented. It efficiently balances the load to save energy and performs efficiently in terms of delivery of total data packets to the BS and network lifetime.

Zungeru et al. explored existing routing protocols in WSNs and categorized them based on path establishment, energy efficiency, network structure, and computational complexity [13]. Further, SI-based and classical routing protocols are compared. Different standard performance metrics have been presented for future comparisons. An energy-efficient clustering approach based on multiobjective particle swarm optimization is presented in [14] to reduce energy consumption, improve network lifetime, and optimize the number of clusters. Hamid et al.'s proposed approach considers intercluster, power of transmission, and degree of nodes. CH manages intracluster and intercluster traffic. A GA-based clustering algorithm [15] is proposed for efficient load balancing, in which communication between CH and nodes is considered to generate initial population. This approach generates children chromosomes to select a mutation point instead of random selection. Performance metrics, such as rate of convergence, number of active CHs and nodes, energy consumption, and execution time, are used to evaluate the proposed approach.

Yao et al. proposed an energy-efficient delay-aware lifetime-balancing data collection technique in which the centralized heuristic is designed to decrease its computational overhead and the distributed heuristic is designed for large-scale network operations to make the proposed solution scalable [16]. Further, the proposed technique is integrated with compressive sensing to decrease total traffic cost for gathering sensor readings under loose delay bounds. A load-balancing clustering algorithm [17] is presented that performs efficiently with nodes having equal load and runs in $O(n \log n)$ time for n nodes. Further, a polynomial time 2-approximation algorithm in which nodes have variable load is presented to test the performance in terms of execution time and network lifetime.

Fatma et al. investigated the consumption of energy while balancing traffic and proved that multiple-paths–based traffic generation is effective in energy consumption as compared to single path [18]. Further, the analytical model for load balancing is expanded and it is concluded that efficient management of traffic reduces the network lifetime. Ipek [19] proposed a load-balancing technique based on a bee pheromone propagation mechanism, which solves the tradeoff between service availability and energy consumption, in which individual nodes locally decide their execution process.

A hybrid differential evolution and simulated annealing (DESA) approach is proposed in [20], which performs clustering to select CH and prevents the early death of CHs (due to improper selection of CHs) and subsequently improves network lifetime. DESA includes a fitness function that takes into consideration the residual energy and distance between the CH and the nodes. Ashok and Kumar proposed a clustering technique based on modified artificial bee colony for load balancing clusters, in which there is quick convergence and improved search area in choice of CHs [21]. The fitness value is taken as the inverse of the energy consumption for a round.

An artificial bee colony (ABC)-based data collection technique that performs three functions is proposed in [22]: mobile sink path-planning optimization, routing the path from node to CH, and then CH selection. It permits a small data latency to identify the mobile sink balance: network reliability optimization, mobile path length optimization, and data collection maximization. Abdolreza and Gharavian proposed an ant colony optimization (ACO)-based routing technique [23] to reduce energy consumption. The energy consumption and hop count are integrated with routing choice by designing a new pheromone update operator. Link cost is defined as a function of node remaining energy and the required transmission energy using that link.

An ACO-based load-balancing routing algorithm (ACOLBR) is presented in [24], in which a spanning tree is used for intracluster routing and intercluster routing is performed by improved ACO that finds optimal and suboptimal paths. The message's positive feedback is utilized to consider transmission delay, residual energy, and propagation distance as the heuristic factor. An ant-colony–based multipath routing algorithm (ACMRA) is presented in [25], in which disjoint multipaths between nodes and CH are identified. The traffic is distributed over identified multipaths. ACMRA is an on-demand multipath algorithm. It works in two phases: constructing a route and transmitting data. Abdelmoniem et al. improved the ad hoc on-demand distance vector (AODV) [26] protocol by proposing two techniques: AODV- and ACO-based multipath routing protocol, namely multiroute AODV ant-routing and load-balanced multiroute AODV ant-routing algorithms. Data are transmitted using identified paths simultaneously, which reduces energy consumption, end-to-end delay, buffer overflow, and routing overhead.

A GA-based construction of load-balanced connected dominating set to reduce the number of participant nodes in communication to improve network lifetime is presented in [27]. Further, workloads of all the dominators are balanced by allocating dominatees to improve network lifetime. A load-balanced clustering algorithm (LBCA) [28] is proposed to balance the load among different clusters. In this, a gateway is used to control the network instead of the CH, which controls different cluster of nodes.

Kumar and Kumar [29] proposed an energy-efficient multiobjective fractional artificial bee colony algorithm to select the CH optimally. Delay, distance, and energy-consumption–based fitness function are designed to control to convergence rate. Raha et al. proposed a GA-inspired protocol for congestion control in WSNs using trust-based routing (GACCTR) [30] for balancing traffic among different nodes between the source and BS according to the trust values of different routes. GAs are utilized to model data transmission through the various alternate route. A general self-organized tree-based energy balance (GSTEB) routing protocol is presented in [31], in which an ABC is utilized to investigate the shortest path between source and sink, based on clustering. The proposed algorithm works in four phases: the initial phase, the tree construction phase, the self-organized data collection and transmission phase, and the information exchange phase. A further intelligent approach can be incorporated to improve the clustering process.

Apart from these techniques, dynamic power management techniques [32–35], LSRA [36], HLBS [37], EEOM [38], ELBS [39], and several other techniques [40–44] have been proposed for load balancing in order to enhance network lifetime. The proposed technique in this research paper addresses several issues, with the following advantages over most of the existing algorithms:

1) It is more energy-efficient and load-balanced.

2) It performs efficiently in terms of active sensor nodes, and it is more reliable, as it performs routing over multiple paths (based on energy consumption) using erasure coding [45] along with clustering.

3) It has optimal time complexity, i.e. O(npx) (n denotes number of nodes, p represents number of paths, and x denotes number of iterations), in contrast to other existing techniques. Moreover, a deterministic PSO is utilized for faster convergence.

3. Energy-efficient load-balancing technique

A data packet is routed from a source node to the BS via a number of intermediate nodes that act as forwarder nodes in single-path routing. In a multipath routing scheme, the same packet is routed via multiple paths discovered between the source and the destination. In single-path routing, there is a probability of failure of intermediate nodes, due to which the reliability of data transmission is decreased. In multipath routing, a packet is divided into n number of subpackets of equal size with some added amount of redundancy and is communicated over n disjoint paths. Only a small number of subpackets are required to reconstruct the original packet at the destination.

Let S_n be a random variable that represents the number of successful data-delivering paths. S_n is upper bounded by n; that is, $S_n \leq n$. The process of transmitting a data packet is considered a Bernoulli experiment. For the ith path, if the transmission process is successful, 1 is assigned to subrun; otherwise, 0 is assigned. The value of S_n is the sum of the values assigned to the n subruns for n disjoint paths. Thus, the expected number of successful data delivering paths can be calculated as Eq. (1):

$$E\left(S_n\right) = \sum_{i=1}^{n} P_i, \tag{1}$$

where P_i is the probability of successfully delivering a packet to the destination node path i, and α is an upper bound for required probability of successfully reconstructing the sent message at the destination. In order to compute the value of E_n for a given α bound by a standard distribution $N(\mu\ ,\ \sigma)$, the mean is given by Eq. (2):

$$\mu = E\left(S_n\right) = \sum_{i=1}^{n} P_i \tag{2}$$

and the standard deviation is calculated as Eq. (3):

$$\sigma^2 = \sum_{i=1}^{n} P_i\left(1-P_i\right) \tag{3}$$

The degree of multipath routing n determines the total number of subpackets. A given pair $(k, \{\ p_1\ ,\ \ldots,\ p_n\}\ )$ generates a different normal distribution, $N(\mu(n),\ \sigma\ (n))$. Therefore, to address this issue, the random variable S_n is transformed into $S_n{}^* = (\mathrm{S}_n - \ \mu)\ /\ \sigma$, which is normally distributed. However, the values of the bound x_α are given for any given α such that $P(S_n{}^* \geq x_\alpha) \geq \alpha$ is satisfied. As a result, $S_n^* = (S_n - \mu\ )\ /\ \sigma \geq\ x_\epsilon$ implies $S_n \geq\ x_\alpha\ \times \sigma + \mu$, and hence probability is given in Eq. (4):

$$P(Sn \geq x_\alpha \times \sigma\ +\ \mu)\ \geq\ \alpha \tag{4}$$

By equating this probability with $P(S_n \geq E_n) \geq\ \alpha$, an estimation of E_n can be obtained for a given bound α using Eq. (5):

$$E_n = \max(\lfloor x_\alpha \times \sigma + \mu \rfloor, 1) \tag{5}$$

By using the values of Eqs. (2) and (3), E_n is obtained from Eq. (6):

$$E_n = \max\left(\lfloor x_\alpha\ \times\ \sqrt{\sum_{i=1}^{n} P_i\left(1\ -\ P_i\right)} + \sum_{i=1}^{n} P_i,\ 1\right), \tag{6}$$

which represents an estimated number of paths successfully delivering data for a given value of α, and data are sent over the multiple paths using erasure coding.

By using above energy model, the proposed technique creates different paths to transfer data, and further energy consumption is calculated as the amount of energy used to transfer the total load (L_{total}). Different sensor activities need to be scheduled in an efficient manner to improve residual energy. Thus, TDMA is used to schedule the tasks of a subset of nodes into different groups with successive time slots. In the proposed technique, the entire WSN is divided into a number of different groups. Each group consists of parent nodes

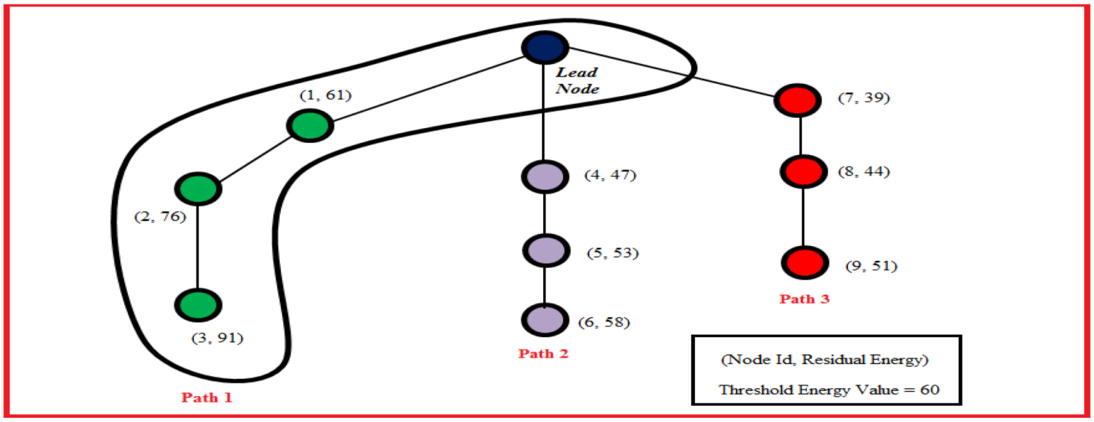

Figure 1. Selection of path with maximum residual energy.

(CHs) and children nodes (cluster members). Parent nodes collect the data from the children nodes and transfer that data to a BS for further processing (Figure 1). In this network model, load (L_i) is calculated for every group (G_i) as given in Eq. (7) for efficient load balancing:

$$L_i = \sum\nolimits_{[j=1]\,\varepsilon\,G_i}^{n} L_j. \tag{7}$$

Here n represents the total number of nodes used in a network. Load at particular sensor (j) is L_j, calculated using Eq. (8):

$$L_j = \frac{rS_n + L_{pc}}{P_n}, \tag{8}$$

where rS_n represents the packets that are generated by the sensors of the group (G_i), L_{pc} represents the number of packets received from the child nodes of the group (G_i), and P_n is the total number of packets transferred in one time slot. L_{total} is the total amount of load transferred in the entire network using TDMA, as in Eq. (9):

$$\text{Fitness} = L_{total} = \sum\nolimits_{i=1}^{n} L_i \tag{9}$$

Energy consumption is calculated as the amount of energy consumed to transfer the total load (L_{total}). In this technique, the residual energy is improved to transfer all the data in assigned time slots.

Network lifetime is defined as the time until the first node or group of nodes in a network runs out of energy, or the time (in terms of number of rounds) it takes for network disconnection to occur due to the failure of one or more nodes, as given by Eq. (10):

$$NL_n^n = min\, v \in V\, NLv \tag{10}$$

where NL_n^n is the network lifetime, with NL_v is the lifetime of node v, and V is the node set excluding the BS.

4. Proposed PSO-EELB

PSO is a very popular metaheuristic approach due to its implementation simplicity, as only a few parameters need tuning, and it has a faster convergence rate than other metaheuristic approaches. It is also very cheap in terms of computation in updating an individual, as it only needs two simple equations, as compared to GAs. To attain a quicker and efficient solution of clustering and routing problem, a metaheuristic approach such a PSO is highly desirable. In this section, the pseudocode of a PSO-EELB algorithm is presented for load balancing in WSNs.

This heuristic may lead the search into an infeasible state, because any node may be unselected. To lead the search back into its feasible state, the node must be released after data are transferred, so that if it still has more residual energy than the threshold value (considered as average residual energy), then it can move into the next iteration. The pseudocode of the proposed PSO-EELB algorithm is depicted in Figure 2.

A node list is then obtained from the participating nodes by selecting only nodes that fulfill node selection criteria. Once the node list has been obtained, a random feasible solution is initialized. The process of choosing the best heuristic from low-level heuristics is initiated. Each particle represents a node identifier with an initial solution in the solution space along with the evaluation function. A low-level heuristic is selected at each particle location and its fitness function (i.e. Fitness (L_{BP})) is computed. If Fitness (L_{BP}) is better than Fitness (G_{BP}),then G_{BP} takes the value of L_{BP}. The fitness value of the particle at the best global position is calculated next. The velocity and position of the selected particle is updated using Eq. (13). Then the fitness value is calculated for the new position and is compared with its previous calculated position. If it is better than the local best value, then the particle's current position is assigned to the local best value.

In the proposed PSO-EELB technique, an effective and efficient deterministic variant of the PSO algorithm is utilized, assuming limited computational resources. Basic PSO utilizes random coefficients to maintain swarm dynamic variety and needs extensive numerical computations to attain statistically convergent outcomes that are too computationally expensive. Therefore, an efficient deterministic approach has been utilized [46].

4.1. PSO formulations of PSO-EELB

As the proposed PSO-EELB algorithm is based on PSO formulations, a general model of PSO formulations utilized in this research work is presented below.

4.1.1. Basic formulation of PSO algorithm

The basic formulation of the PSO algorithm, as presented by Shi and Eberhart [47], is

$$\begin{cases} v_i^{k+1} = wv_i^k + c_1 r_1 \left(X_{i,pb} - X_k^i\right) + c_2 r_2 (X_{gb} - X_k^i) \\ X_i^{k+1} = X_i^k + v_i^{k+1} \end{cases} \tag{11}$$

The above equations represent the speed and position of the ith particle at the kth iteration, respectively; w is the inertia weight; c_1 and c_2 represent the social and cognitive learning rate, respectively; r_1 and r_2 denote two random numbers in the range [0, 1]; $X_{i,pb}$ is the personal best position ever found by the ith particle; and X_{gb} is the global best position ever found among all particles. The use of the constriction factor χ is necessary to ensure convergence of PSO [47–50]. Accordingly, the system in Eq. (11) is amended as follows:

$$\begin{cases} v_i^{k+1} = \chi\,[v_i^k + c_1 r_1 \left(X_{i,pb} - X_k^i\right) + c_2 r_2 \left(X_{gb} - X_k^i\right)] X_i^{k+1} = X_i^k + v_i^{k+1} \\ \chi = \dfrac{2}{\left|\sqrt{2 - \varphi - \sqrt{\varphi}^2 - 4\varphi}\right|}\ where\,\varphi = c_1 + c_2,\ \varphi > 4 \end{cases} \tag{12}$$

Typically, when the constriction method is used, φ value is set to 4.1, with $\chi = 0.729$, $c_1 = c_2 = 1.494$.

Algorithm 1: PSO Based Energy Efficient Load Balancing Algorithm

Input Data: Number of nodes and their values of residual energy, node id, location (distance from the BS in X and Y position), energy loss and energy loss ratio.

Result: Selection of effective paths for load balancing.

1. **Start**
2. Initialize Node List
3. Initialize Residual Energy Value
4. Initialize a random feasible solution and particles P_i, ∀i, $1 \leq i \leq N_p$. /*As described in section 4.1 */
5. Np = Population Size, R_V = Random Velocity, P_v = Particle Velocity, P_p = Particle Position, P_{op} = Population, G_{BP} = Global Best Position, L_{BP} = Local Best Position
6. **for** $i = 1$ *to* N_p **do**

    $P_v \leftarrow R_V()$

    $P_p \leftarrow$ Random_Position (N_p)

    $L_{BP} \leftarrow P_p$

    For each particle, calculate the value of fitness function

    if *Fitness* $(G_{BP}) \geq$ *Fitness* (L_{BP}) **then**

    $G_{BP} \leftarrow L_{BP}$

7. **while** *maximum iteration is not satisfied* **do**

    for $P \in P_{op}$ **do**

    $P_v \leftarrow$ UpdateVelocity (P_v, G_{BP}, L_{BP}) using (eq. 13)

    $P_p \leftarrow$ UpdatePosition (P_p, P_v) using (eq. 13)

    if *Fitness* $(P_p) \leq$ *Fitness*(L_{BP}) **then**

    $L_{BP} \leftarrow P_p$

    if $(L_{BP}) \leq$ *Fitness*(G_{BP}) **then**

    $G_{BP} \leftarrow L_{BP}$

8. **Return** (G_{BP})
9. **while** *there are un-selected nodes in the input queue* **do**

    for *every node in the node list* **do**

    get the next node from queue

    select the node based on fitness value

10. Repeat the process until the complete path is generated.
11. Store the path in a hash map; with index as key and path as value.
12. Modify *Ps, Po* by removing intermediate nodes of the path selected.
13. **if** all possible paths are generated then

    From (eq. 6), calculate the number of successful paths.

    Split the packet into n sub-packets using Erasure coding and send these sub-packets over multiple paths generated.

    End

    Else go to Step no. 6.

14. **End**

Figure 2. PSO-based energy-efficient load-balancing algorithm.

4.1.2. Deterministic formulation

In order to make the overall PSO more efficient for CPU-time expensive analyses, a deterministic algorithm is formulated by suppressing the random coefficients in Eq. (12), which becomes

$$\begin{cases} v_i^{k+1} = \chi \left[v_i^k + c_1 \left(X_{i,pb} - X_k^i \right) + c_2 (X_{gb} - X_k^i) \right] \\ X_i^{k+1} = X_i^k + v_i^{k+1} \end{cases} \tag{13}$$

The complexity of the proposed PSO-EELB algorithm is *O(npX)* for p number of paths having n number of nodes, and X denotes the number of iterations running on sink. The initialization of particle location and speed is performed using a deterministic and homogeneous distribution, according to Hammersley sequence sampling [51]. PSO parameters and their values for the proposed PSO-EELB are listed in Table 1.

Table 1. PSO parameters.

Parameter	Value
N_p	60
C_1	1.4962
C_2	1.4962
χ	0.7968

5. Cluster formation in the PSO-EELB approach

The cluster is formed by the BS on the basis of centralized clustering. For clustering, BS broadcasts an information collection message to all the nodes. After receiving the message, a node starts to send its information such as node id, location (distance from the BS in X and Y position), energy loss and energy loss ratio (velocity), and current energy to BS. Then BS initiates the clustering process steps as follows:

Step 1. Conversion of problem into the PSO space, in which the PSO particle has two dimensions such as position and velocity.

Step 2. Estimation of fitness value using fitness function:

The proposed fitness function for PSO-based clustering is to optimize the average distance and average energy of the member nodes and distance from the current CH and headcount. The fitness value is calculated for the particle by using Eq. (14):

$$\text{Fitness value} = Fv = \alpha_1 \times \frac{\sum_{i=0}^{n} d(\text{current node, member } i)}{n} + \alpha_2 \times \frac{\sum_{i=0}^{n} E(\text{member } i)}{n} + (1 - \alpha_1 - \alpha_2) \times \frac{1}{\text{No.of members covered by current node}}, \tag{14}$$

where α_1 and α_2 are weighting parameters (normalized values) and n denotes the number of members covered within the cluster.

Step 3. Generation of new particles from the initial solution. The formation of new particles from the old one is the generation of a new particle.

Step 3.1. Estimation of the new velocity. The current velocity of a taken particle is considered the rate at which the particle's position changes. Based on Eq. (13), the new velocity is calculated in Eq. (15) as follows:

$$\begin{aligned}\text{new velocity} \;&=\; \chi[\text{old velocity} + w_1\,(\text{localbestposition} - \text{currentbestposition}) \\ &\quad + w_2(\text{global best position current best position})]\end{aligned} \tag{15}$$

where χ denotes constriction factor, and w_1 and w_2 are basic PSO tuning parameters denoting social and cognitive learning rates, respectively.

Step 3.2. Based on Eq. (13), estimation of the new position of the particle is calculated in Eq. (16) as follows:

$$\text{new position} \;=\; \text{old position} \;+\; \text{new velocity} \tag{16}$$

Finally the new particle (new velocity and new position) arrives.

Step 4. Calculation of fitness value for new particles.

The fitness value of the new particles is estimated by using the fitness function in Step 2 with the new velocity and new position.

Step 5. Fitness values of the old and new particles are compared and the best one is selected for the next iteration.

Step 6. For each iteration, one best solution is selected as a local best solution. The particle that has maximum fitness value in the current iteration is selected as *lbest* solution.

Step 7. The local best solutions from all iterations of the particle that have optimal values among all solutions are selected as a global best solution *gbest*. The final solutions are decoded into clusters. The BS forms the cluster using PSO and broadcasts a cluster-announcement message to nodes that contains cluster information.

5.1. CH selection in the PSO-EELB approach

After clustering, each sensor node maintains a cluster list. It includes current cluster id, velocity, location and energy. Then the round procedure is initiated to perform CH selection by implementing a PSO algorithm.

Step 1. The members covered by the current node communicate with each other to select a CH, following the steps mentioned below.

Step 2. Estimation of fitness value using fitness function:

$$\begin{aligned}\text{Fitness value} \;&=\; Fv \;=\; \alpha_1 \times \frac{\sum_{i=0}^{m} d(\text{current node, member}\, i)}{n}\gamma + \alpha_2 \times \frac{\sum_{i=0}^{m} E(\text{member}\, i)}{n}\gamma \\ &+\; (1-\alpha_1-\alpha_2) \times \frac{1}{\text{No.of members covered by current node}},\end{aligned} \tag{17}$$

where $\gamma = \left\{ \begin{array}{l} 1,\text{ if member } i \text{ is covered by current node} \\ 0,\text{ else} \end{array} \right\}$, m is the number of members in the current cluster, α_1 and α_2 are weighing parameters (normalized values), and n denotes the number of members covered within the competition range.

Repeat steps 3 to 7 of cluster formation for CH selection (as discussed in Section 4.2). Finally, the particle that has a global best solution is chosen as a current CH.

5.2. Multihop intracluster and intercluster data transmission

Based on TDMA, the tasks of data collection from cluster members are scheduled by the CH, and data collection from CHs is scheduled by the BS for different groups/clusters within successive time slots. CHs gather the data from the cluster members and transfer that data to the BS for further processing. As the BS already carries every node's information, such as location, cluster id, cluster members, location, residual energy, and its CH, it creates disjoint multiple paths for each node towards its destination (CH or BS) using PSO, based on the fitness value given by Eq. (18):

$$\text{Fitness function} = \frac{d(si\,,\ sj)^2 + d(sj,\ SN)^2}{\max(d\,(si,\ sj)^2 + d\,(sj, SN)^2)} + (1-\omega)\times\frac{\max(E(j))E\,(j)}{\max\,(E\,(j))}, \omega\in[0,1] \tag{18}$$

where ω is a randomized tuning parameter and s_i and s_j are the source and destination nodes. After that, the sink assigns a TDMA time slot to each cluster, and the CHs in turn assign a TDMA time slot to each node in the cluster to send packets to it. Without waiting for the delivery of packets, the node is turned off and goes into sleep mode to minimize the energy consumption, as given by Eq. (19).

$$\text{TDMA time slot duration for each node} = \frac{\text{Cluster TDMA time slot duration}}{\text{Number of cluster members}} \tag{19}$$

A source node follows certain rules in order to send its data to the destination: let the distance limit for source and destination be d_0, as calculated in Eq. (20). If d (actual distance between source and the destination) is less than d_0, then data can be transmitted in a single hop from the source to the destination, in the form of direct communication. If $d > d_0$, a PSO-based algorithm presented as Algorithm 1 is utilized for data transmission to minimize energy consumption and to enhance the network life cycle.

$$d_0 = \frac{\sqrt{s}}{\sqrt{n}} \tag{20}$$

Here s is the area of the cluster and n denotes the number of nodes in the cluster.

5.3. Energy model

An energy model designed in the physical layer is used for calculating energy loss in each sensor node while communicating with other nodes [52]. The two channel-propagation models used are the free space model (d^2 power loss) for the purpose of one-hop or direct transmission and the multipath fading channel model (d^4 power loss) for packet transmission via multihop. Thus, the energy exhausted for this kind of transmission of n-bit packet over distance d is calculated using Eq. (21):

$$E_{TX}\ (l,d)\begin{cases} lE_{elec} + l\epsilon_{fs}d^2, & d < d_0, \ or \\ lE_{elec} + lepsilon_{mp}d^4, & d \geq d_0, \end{cases} \tag{21}$$

where ϵ_{fs} is free space energy loss, ϵ_{mp} is multipath energy loss, d is the distance between a source node and a destination node, and d_1 is the crossover distance given by Eq. (22):

$$d_1 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{mp}}} \tag{22}$$

The energy spent for the radio to receive this message is

$$E_{RX}(l) = lE_{elec} \tag{23}$$

Thus, the transmission power and receiving power energy levels are designed in physical and MAC layer of the WSN.

6. Experimental setup and results

The performance evaluation of proposed PSO-EELB is done through the real test bed Coalesenses iSense network. The hardware is arranged around the iSense CoreModule3 with an IEEE 802.15.4 compliant radio, a 32-bit RISC (JN5148) controller running at 16 MHz, 512 KB of flash, and 128 kb of memory, with a highly accurate clock (typ. 6 ppm) and a switchable power regulator. The iSense hardware is augmented with modular operating and networking firmware based on object-oriented programming. The software components needed for programming iSense modules are Cygwin, ba-elf2 compiler, iSense firmware, iShell, Eclipse for C/C++, and iSense Gateway Module USB Driver for Windows. The modules included Zigbee-ready radio, which offers high data rates at ranges of up to 60 m while providing hardware AES encryption. Within a network, the 6LoWPAN protocol suite transmits IPv6 datagrams over the IEEE 802.15.4 radio interface. The various PSO parameters and their values for proposed approach are shown in Table 1. The parameters for the experimental setup are listed in Table 2. The BS is assumed to be situated in the center of the region.

Table 2. Experimental parameters and their values.

Parameters	Values
Number of nodes	100
Area of deployment	200 m × 200 m
Frequency	2.4 GHz
Initial energy of sensor nodes	2.0 J
Number of execution iterations	100
Communication range of node	60 m using Lucent WaveLan DSSS radio
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
Data packet size	4000 bits
Protocol used	802.11 MAC protocol
Traffic source type	Constant bit rate (CBR) sources
Sending rate	1–4 packets per second
Evaluation time	Periodic sample time 100 s to analyze the changes, 800 s for general performance analysis and energy consumption analysis.

6.1. Validation of the proposed technique

To validate the proposed PSO-EELB technique, some existing approaches (DESA [20], ACOLBR [24], GACCTR [30], and GSTEB [31]) that also perform energy-based load-balancing based on other metaheuristic methods, as discussed in Section 2, are selected. These techniques have been implemented in the above environment and the results are compared in terms of different parameters: energy consumption, throughput, convergence rate, number of data packets received, execution time, network lifetime, and number of active nodes to prove the effectiveness of the proposed technique.

Test case 1: Energy consumption versus number of nodes.

Energy consumption is calculated for the proposed PSO-EELB technique and the existing DESA, GSTEB, ACOLBR, and GACCTR techniques with different numbers of nodes (20–100). Each node consumes energy for communication and computation. Thus, as the increase in the number of nodes, energy consumption also increases. Energy consumption is calculated on the basis of the energy model presented in Section 4.5 and the parameter values are shown in Table 2. Energy consumption in PSO-EELB is lower than the other techniques at different number of nodes (Figure 3). The minimum value of energy consumption is 7.52 J at 20 nodes in PSO-EELB. Average energy consumption in PSO-EELB is 6.34%, 9.721%, 10.54%, 12.64%, and 16.66% less than ACOLBR, GSTEB, GACCTR, and DESA, respectively. The rationale behind this difference in performance is that in ACOLBR, a minimum spanning tree (MST) is used for intracluster routing, and so each time cluster reformation takes place, a MST must be generated. As the number of nodes increases, there is huge energy consumption due to network structure formation. In GACCTR, parent selection is performed based on the GA. For ensuring reliability, a trust function is calculated for every path through message exchange, which results in energy dissipation. Although GLBCA and GSTEB consume more or less the same amount of energy, it can still be claimed that the proposed algorithm performs better.

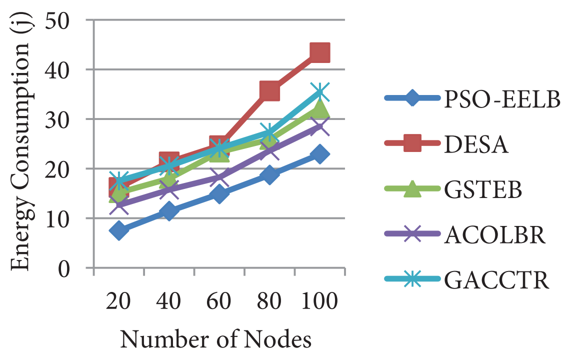

Figure 3. Energy consumption versus number of nodes.

Test case 2: Network lifetime versus number of nodes.

The value of network lifetime has been calculated for PSO-EELB, HLBS, and ELBS with different number of nodes based on Eq. (10). In the existing techniques, network lifetime decreases with increasing number of nodes (20 to 100). The rationale behind this performance is the communication overhead involved during topology formation. Network lifetime in PSO-EELB is longer than GACCTR, GSTEB, ACOLBR, and DESA at different numbers of nodes (Figure 4). Maximum network lifetime is 130 s at 20 nodes. Average network lifetime in PSO-EELB is 12.63%, 13.71%, 15.12%, and 18.75% longer than DESA, ACOLBR, GSTEB, and GACCTR, respectively.

Test case 3: Number of active nodes versus number of iterations.

The number of active nodes was calculated for PSO-EELB, GACCTR, GSTEB, ACOLBR, and DESA with an increasing number of iterations (1 to 100). A node is termed as active if its current remaining energy is above zero and there is at least one CH within its radius. As the number of iterations increases, the number of active nodes decreases due to energy dissipation. Network lifetime in PSO-EELB was longer than in GACCTR, GSTEB, ACOLBR, and DESA for different values of energy consumption with an increasing number of iterations (Figure 5). The maximum number of active nodes is 85 at 20 iterations. The number of active nodes in PSO-EELB is 15.98%, 14.22%, 12.97%, and 10.16% greater than GACCTR, ACOLBR, DESA, and GSTEB, respectively.

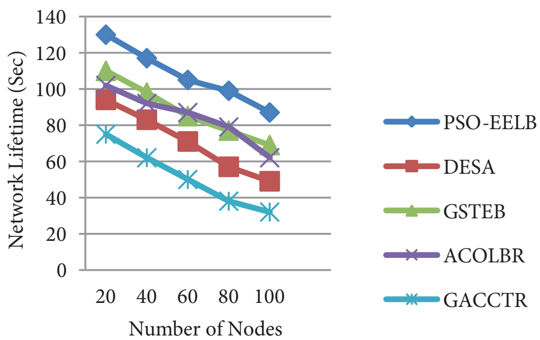

Figure 4. Network lifetime versus number of nodes.

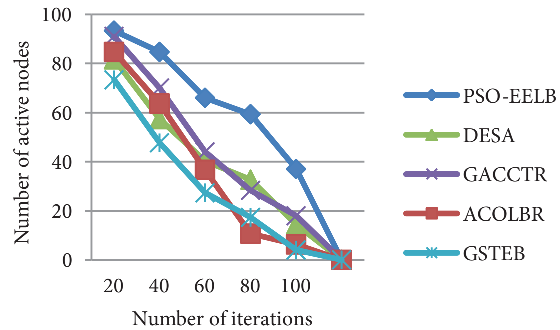

Figure 5. Number of active nodes versus number of iterations.

Test case 4: Received data packets versus residual energy.

Residual energy with respect to received data packets was calculated for PSO-EELB, GACCTR, GSTEB, ACOLBR, and DESA. As shown in Figure 6, the receiving rate of data packets decreases with decreasing residual energy. Initially, the maximum number of packets transfers at 8 J energy residual, but PSO-EELB, DESA, and ACOLBR receive almost the same number of data packets. At 7.5 J energy residual, PSO-EELB receives 1.87%, 16.11%, 17.39%, and 17.62% more data packets than ACOLBR, GACCTR, GSTEB, and DESA respectively. The maximum number of data packets received in PSO-EELB is 7.93 at 8 J energy residual, and the minimum number of data packets received in PSO-EELB is at 4 J. The average number of received data packets in PSO-EELB is 2.76%, 11.67%, 14.71%, and 21.59% more than ACOLBR, GACCTR, DESA, and GSTEB, respectively.

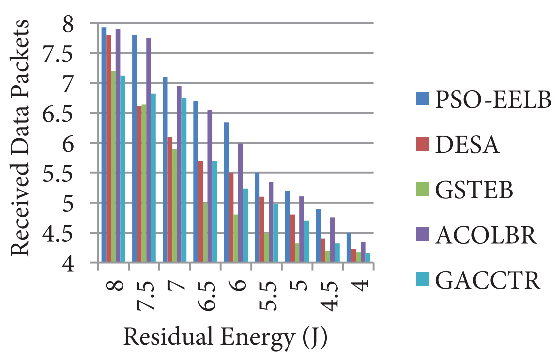

Figure 6. Received data packets versus residual energy.

Test Case 5: Throughput. Throughput is a ratio of the total amount of successfully transferred data to the total amount of time required to transfer data. It is calculated using Eq. (24).

$$\text{Throughput}=\frac{\text{Total amount of data transferred successfully}\,(\text{D})}{\text{Total amount of time required to transfer data}\,(T)} \tag{24}$$

The value of energy consumption was calculated for PSO-EELB, GACCTR, GSTEB, ACOLBR, and DESA with different numbers of nodes (20 to 100). Figure 7 shows the comparison of throughput of PSO-EELB, GACCTR, GSTEB, ACOLBR, and DESA, and it is clear that PSO-EELB performs better than the selected existing techniques. The maximum value of throughput at 20 nodes in PSO-EELB has 6.41%, 9.17%, 17.66%, and 18.22% more throughput than GSTEB, DESA, GACCTR, and ACOLBR, respectively.

Test case 6: Convergence curve.

Figure 8 plots the convergence rate for total load transmitted by GACCTR, GSTEB, DESA, ACOLBR, and the proposed PSO-EELB over a number of iterations; the proposed algorithm clearly demonstrates a faster convergence rate. Initially, the load is equal and randomly initialized. Therefore, the total initial load is very high at the 0th iteration. As the algorithm progresses, convergence is drastic and achieves global minima very quickly. The number of iterations required for the convergence ranges from 27 to 100.

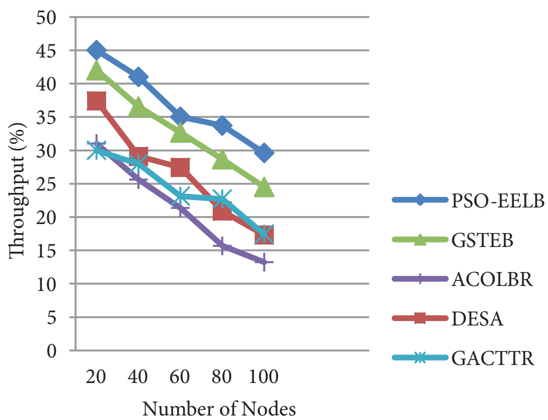

Figure 7. Throughput versus number of nodes.

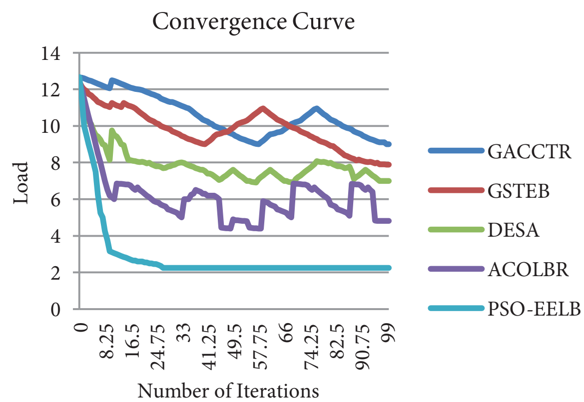

Figure 8. Convergence curve of load.

Apart from these test cases, the proposed PSO-EELB algorithm also performs well in terms of algorithmic complexity, as compared to GSTEB, GACCTR, DESA, and ACOLBR. The overall run-time complexity of the existing ACOLBR is $n(\log n) + nk^2$, where$n \leq k$in every case. In this case, for intracluster routing operations, the complexity is nk^2 and intercluster routing operations are $n(\log n)$, where k is the number of nodes in the cluster and n is the number of clusters. In the case of GACCTR, the overall complexity is $k(2n^2 g + ln + l + n^2)$, where g denotes the number of generations in the population, n represents the number of nodes in the grid, and k is the number of iterations. In DESA, the worst-case complexity is $O(kn^2)$; here k is the number of iterations of simulated annealing and $O(n)^2$ is the worst-case complexity of differential evolution. In GSTEB, the overall complexity of the algorithm is $[n^2p + Hkmn + s(Tnkm + H(H + nkm + nkC) + np^2]$. Here s is the number of iterations, H is the number of employed bees or food sources,n is the number of data objects in the dataset X, k is the number of clusters, m is the number of attributes, and p is the total number of categories for all attributes. The complexity of the proposed PSO-EELB algorithm is $O(npX)$ for p number of paths having n number of nodes, and X denotes the number of iterations running on sink, which proves that PSO-EELB performs far better as compared to the other techniques.

6.2. Statistical analysis

Coefficient of variation ($Coff.of\,Var$)[53] is used to analyze the statistical significance of the results. It is a statistical measure used to analyze data dispersion about the mean value and for comparing different means. It also provides an overall performance analysis of the technique being used for generating the statistics. It defines the data deviation as a proportion of its mean value, and is calculated using Eq. (25):

$$Coff.of\,Var. = \frac{SD}{M} \times 100, \tag{25}$$

where SD is standard deviation and M is mean. $Coff.\ of\,Var$ of energy consumption has been evaluated in the proposed load-balancing technique (PSO-EELB) and in the existing techniques (GACCTR, GSTEB, ACOLBR, and DESA) (Figure 9).

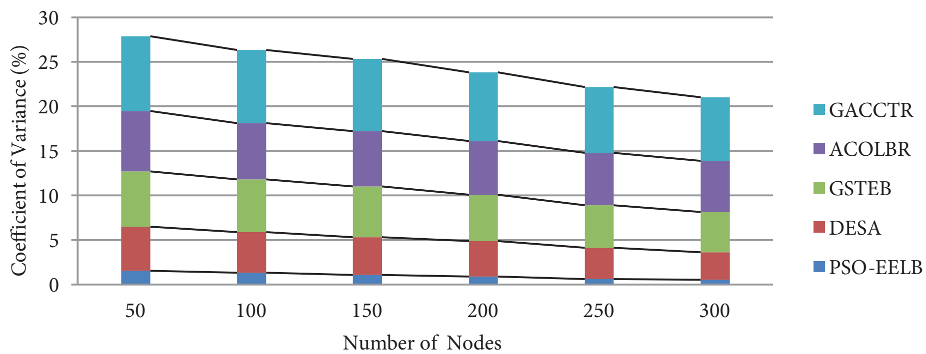

Figure 9. CoV for execution time of various load-balancing techniques.

$Coff.of\,Var$ is calculated for the energy consumption results attained by PSO-EELB, GACCTR, GSTEB, ACOLBR, and DESA. In PSO-EELB, $Coff.of\,Var$ ranged from 0.52% to 1.55%, proving its stability (Figure 9). A small value of $Coff.of\,Var$ indicates that PSO-EELB is more efficient at load balancing in situations where the number of nodes vary. $Coff.of\,Var$ decreases as the number of nodes increases. Statistical analysis illustrates that the PSO-EELB outperforms existing load-balancing techniques for large numbers of nodes.

7. Conclusions and future scope

A PSO-EELB technique for WSNs has been proposed in this paper. The main objective of this proposed work is to minimize energy consumption and improve network lifetime and throughput. In the proposed technique, the number of routing paths is identified and energy consumption of different nodes and paths is calculated. Based on PSO, multiple paths are selected and load balancing is performed by sending a packet (divided into subpackets) using erasure coding for data transfer at particular point of time. The effectiveness and usefulness of the proposed technique are determined based on the metrics designed. For real testbed evaluation, the Coalesenses iSense network was used for experimental performance evaluation. The results of real testbed evaluations demonstrate that PSO-EELB is effective in terms of energy consumption, throughput, network lifetime, number of active nodes, convergence rate, execution time, and number of packets received, as compared to existing load-balancing techniques (GACCTR, GSTEB, ACOLBR, and DESA) with different numbers of nodes and numbers of iterations. Further, statistical analysis demonstrates that PSO-EELB outperforms existing load-balancing techniques for large numbers of nodes. In the future, weight-based data aggregation schemes based on PSO-EELB using multiple sinks can be designed and implemented to further reduce energy consumption.

Acknowledgment

We would like to thank all the anonymous reviewers for their valuable comments and suggestions for improving the paper.

References

[1] Akyildiz IF, Su W, Sankarasubramaniam Y, Cayirci E. Wireless sensor networks: a survey. Comput Networks 2002; 38: 393- 422.

[2] Sha K, Gehlot J, Greve R. Multipath routing techniques in wireless sensor networks: a survey. Wireless Pers Commun 2013; 70: 807-829.

[3] Kim HY, Park HJ. An efficient gaming user oriented load balancing scheme for MMORPGs. Wireless Pers Commun 2013; 73: 289-297.

[4] Kacimi R, Dhaou R, Beylot AL. Load balancing techniques for lifetime maximizing in Wireless Sensor Networks. Ad Hoc Networks 2013; 11: 2172-2186.

[5] Anastasi G, Conti M, Di Francesco M, Passarella A. Energy conservation in wireless sensor networks: a survey. Ad Hoc Networks 2009; 7: 537-568.

[6] Chang JH, Tassiulas L. Maximum lifetime routing in wireless sensor networks. IEEE ACM T Network 2004; 12: 609-619.

[7] Carle J, Simplot-Ryl D. Energy-efficient area monitoring for sensor networks. IEEE Comput Soc 2004; 47: 40-46.

[8] Muhammad S, Caro Di GA, Farooq M. Swarm intelligence based routing protocol for wireless sensor networks: Survey and future directions. Inf Sci (Ny) 2011; 181: 4597- 4624.

[9] Kuila P, Jana PK. Energy efficient load-balanced clustering algorithm for wireless sensor networks. Technology 2012; 6: 771-777.

[10] Azharuddin MD, Kuila P, Jana PK. Energy efficient fault tolerant clustering and routing algorithms for wireless sensor networks. Comp El En 2015; 41: 177-190.

[11] Kuila P, Jana PK. A novel differential evolution based clustering algorithm for wireless sensor networks. Appl Soft Comput J 2014; 25: 414-425.

[12] Kuila P, Jana PK. Energy efficient clustering and routing algorithms for wireless sensor networks: particle swarm optimization approach. Eng Appl Artif Intel 2014; 33: 127-140.

[13] Zungeru AM, Ang LM, Seng KP. Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison. J Netw Comput Appl 2012; 35: 1508-1536.

[14] Hamid A, Shahzad W, Khan FA. Energy-efficient clustering in mobile ad-hoc networks using multi-objective particle swarm optimization. Appl Soft Comput 2012; 12: 1913-1928.

[15] Kuila P, Gupta SK, Jana PK. A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. Swarm Evol Comput 2013; 12: 48-56.

[16] Yao Y, Cao Q, Vasilakos AV. EDAL: An energy-efficient, delay-aware, and lifetime-balancing data collection protocol for heterogeneous wireless sensor networks. In: Proceedings of the 10th International Conference on Mob Ad-Hoc and Sensor Systems; 14–16 October 2013; Hangzhou, China: IEEE. pp. 182-190.

[17] Kuila P, Jana PK. Approximation schemes for load balanced clustering in wireless sensor networks. J Supercomput 2014; 68: 87-105.

[18] Fatma B, Bouabdallah N, Boutaba R. Load-balanced routing scheme for energy-efficient wireless sensor networks. In: Proceedings of the Global Telecommunications Conference; 30 November–4 December 2008; New Orleans, LA, USA: IEEE. pp. 1-6.

[19] Caliskanelli I. A bio-inspired load balancing technique for wireless sensor networks. PhD, University of York, Heslington, UK, 2014.

[20] Potthuri S, Shankar T, Rajesh A. Lifetime improvement in wireless sensor networks using hybrid differential evolution and simulated annealing (DESA). Ain Shams Eng J 2016; 7: 867-872.

[21] Ashok BN, Kumar R. Load balancing clustering in WSN using MABC. J Inform Technol 2014; 6: 389-398.

[22] Yue Y, Li J, Fan H, Qin Q. Optimization-based artificial bee colony algorithm for data collection in large-scale mobile wireless sensor networks. J Sensors 2016; 2016: 1-12.

[23] Abdolreza M, Gharavian D. An ant colony optimization based routing algorithm for extending network lifetime in wireless sensor networks. Wirel Netw 2015; 22: 2637-2647.

[24] Bi J, Li Z, Wang R. An ant colony optimization-based load balancing routing algorithm for wireless multimedia sensor networks. In: Proceedings of the IEEE International Conference on Communication Technology (ICCT); 11–14 November 2010; Nanjing, China: IEEE. pp. 584-587.

[25] Yang J, Lin Y, Xiong W, Xu B. Ant colony-based multi-path routing algorithm for wireless sensor networks. In: Proceedings of the International Workshop on Intelligent Systems and Applications; 23–24 May 2009; Wuhan, China: IEEE. pp. 1-4.

[26] Abdelmoniem AM, Ibrahim HM, Marghny HM, Hedar AR. Ant colony and load balancing optimizations for AODV routing protocol. Int J Sens Networks Data Commun 2012; 1: 1-14.

[27] He J, Ji S, Yan M, Pan Y, Li Y. Genetic-algorithm-based construction of load-balanced CDSs in wireless sensor networks. In: Proceedings of the Military Communications Conference; 7–10 November 2011; Baltimore MD, USA: IEEE. pp. 7-11.

[28] Revathi AR, Santhi B. Efficient clustering for wireless sensor networks using evolutionary computing. Indian J Sci Technol 2015; 8: 1-5.

[29] Kumar R, Kumar D. Multi-objective fractional artificial bee colony algorithm to energy aware routing protocol in wireless sensor network. Wirel Netw 2016; 22: 1461-1474.

[30] Raha A, Naskar MK, Paul A, Chakraborty A, Karmakar A. A genetic algorithm inspired load balancing protocol for congestion control in wireless sensor networks using trust based routing framework (GACCTR). I J Comput Netw Inf Secur 2013; 9: 9-20.

[31] Kaur S, Gangwar RC. Hybrid GSTEB routing protocol using clustering and artificial bee colony optimization. In: Proceedings of the International Conference on Green Computing and Internet of Things; October 2015; Computer Society Washington, DC, USA: IEEE. pp. 661-666.

[32] Shu T, Krunz M. Coverage-time optimization for clustered wireless sensor networks: A power-balancing approach. IEEE/ACM Trans Netw 2010; 18: 202-215.

[33] Venkateswarlu M, Sekaran C, Kandasamy A. Node-link disjoint multipath routing protocols for wireless sensor networks: a survey and conceptual modeling. In: Thilagam PS, Pais AR, Chandrasekaran K, Balakrishnan N, editors. Advanced Computing, Networking and Security. Surathkal, India: Springer. pp. 405-414.

[34] Haenggi M. Energy-balancing strategies for wireless sensor networks. In: Proceedings of the International Symposium on Circuits and Systems (ISCAS'03); 25–28 May 2003; Bangkok, Thailand: IEEE. pp. 828-831.

[35] Muruganathan S, Ma D, Bhasin R, Fapojuwo A. A centralized energy-efficient routing protocol for wireless sensor networks. IEEE Commun Mag 2005; 43: 8-13.

[36] Bhardwaj M, Chandrakasan A. Upper bounds on the lifetime of wireless sensor networks. In: Proceedings of the IEEE International Conference on Communications; 11–14 June 2001; Helsinki, Finland: IEEE. pp: 785-790.

[37] Kim HY, Park HJ, Lee S. A hybrid load balancing scheme for games in wireless networks. Int J Distrib Sens Networks 2014; 10: 1-7.

[38] Jung S. Energy efficiency of load balancing in MANET routing protocols. In: Proceedings of the Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing; 23–25 May 2005; Washington, DC, USA: IEEE. pp. 476-483.

[39] Montoya GA, Donoso Y. Energy load balancing strategy to extend lifetime in wireless sensor networks. Proc Comp Sci 2013; 17: 395-402.

[40] Coleri S, Ergen M, Koo TJ. Lifetime analysis of a sensor network with hybrid automata modelling. In: Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA'02); September 28–30, 2002; NY, USA: ACM. pp. 98-104.

[41] Yamunadevi SP, Vairam T, Kalaiarasan C, Vidya G. Efficient comparison of multipath routing protocols in WSN. In: Proceedings of the International Conference of Computing Electronics Electrical Technology; 21–22 March 2012; Tamil Nadu, India: IEEE. pp. 807-811.

[42] Jiang G, Li B, Long Z, Zhang L. The design of energy-efficient optimal multipath routing protocol based on wireless sensor networks. In: Sambath S, Zhu E, editors. Frontiers in Computer Education. Berlin, Germany: Springer, 2012; 33: pp. 457-464.

[43] Chen G, Yu J. Particle swarm optimization algorithm. Inform Control 2005; 34: 318-324.

[44] Blough DM, Santi P. Investigating upper bounds on network lifetime extension for cell-based energy conservation techniques in stationary ad hoc networks. In: Proceedings of the 8th Annual International Conference on Mobile Computing and Networking; 23–28 September 2002; Atlanta, GA, USA: ACM MobiCom. pp. 183-192.

[45] Mutschlechner M, Li B, Kapitza R, Dressler F. Using erasure codes to overcome reliability issues in energy-constrained sensor networks. In: Proceedings of the 11th Annual Conference on Wireless On-demand Network Systems and Services; 2–4 April 2014; Obergurgl, Austria: IEEE. pp. 41-48.

[46] Serani A, Diez M, Leotardi C, Peri D, Fasano G, Iemma U, Campana EF. On the use of synchronous and asynchronous single-objective deterministic particle swarm optimization in ship design problems. In: Proceedings of the International Conference on Engineering and Applied Sciences Optimization; 4–6 June 2014; Kos Island, Greece.

[47] Shi Y, Eberhart RC. Parameter selection in particle swarm optimization. In: Proceedings of the Seventh Annual Conference on Evolutionary Programming; 25–27 March 1998; New York, NY, USA: Springer. pp. 591-600.

[48] Clerc M. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In: Proceedings of the Congress on Evolutionary Computation; 6–9 July 1999; Washington, DC, USA: IEEE. pp. 1951-1957.

[49] Eberhart RC, Shi Y. Comparing inertia weights and constriction factor in particle swarm optimization. In: Proceedings of the IEEE Congress on Evolutionary Computation; 16–19 July 2000; San Diego, CA, USA: IEEE. pp. 84-88.

[50] Eberhart RC, Shi Y. Particle swarm optimization: Developments, applications and resources. Evolutionary computation. In: Proceedings of the IEEE Congress on Evolutionary Computation (CEC); 27–30 May 2001; Seoul, Republic of Korea: IEEE. pp. 949-956.

[51] Wong TT, Luk WS, Heng PA. Sampling with Hammersley and Halton points. J Graph Tools 1997; 2: 9-24.

[52] Wang J, Yang X, Ma T, Wu M, Kim JU. An energy efficient competitive clustering algorithm for wireless sensor networks using mobile sink. I J Grid Distr Comput 2012; 5: 79-92.

[53] Singh S, Chana I. Q-aware: Quality of service based cloud resource provisioning. Comput Electr Eng 2015; 47: 138-160.