JOURNAL OF INFORMATION SCIENCE AND ENGINEERING 36, XXXX-XXXX (2020)
DOI: 10.6688/JISE.202009_36(5).0013

CSO-Based Energy-Efficient Reliable Sectoring-Scheme in WSN

DHANASHRI NARAYAN WATEGAONKAR AND T. R. RESHMI
School of Computing Science and Engineering
Vellore Institute of Technology
Chennai, Tamil Nadu, postcode India
E-mail: dhanashrinarayan.2016@vitstudent.ac.in; reshmi.tr@vit.ac.in

The emphasizes on nature-inspired meta-heuristic techniques such as cat swarm optimization (CSO) and particle swarm optimization (PSO) for identifying the election of the optimal location sector head (SH) and invoking rapid communication is studied in the paper. The aim of this study is to use the intelligence of the behaviour of cats to solve the chain-based sectoring (partitioning) problem associated with a Nondeterministic Polynomial (NP) hard block. Here, a unique chain-based energy-efficient reliable sectoring scheme (EERSS) is proposed for electing an optimal number of SHs and is implemented for increasing the reliability of a network. The uniqueness of the proposed EERSS is the consideration of factors such as the receiving signal strength identification (RSSI) value, one hop away nodes from a sink, Euclidean distance, remaining energy, neighbour table, and distance between the SH node and the sink while electing a node as an SH. The CSO algorithm is applied to the sectoring scheme for obtaining the optimal number and location of the elected SHs. The unique CSO-based sectoring scheme results are compared with those of well-known optimization technique like PSO. The simulation results show that the CSO-based EERSS provides improved reliability, by consuming less energy and time when compared to the existing chain-based clustering schemes such as the original EERSS, PEGASIS (Power-Efficient Gathering in Sensor Information System) and PSO-based EERSS.

Keywords: static sectoring scheme, cat swarm optimization, packet delivery ratio, energy consumption, end to end delay

1. INTRODUCTION

Currently, applications like vehicle tracking system, gas detection system, movement detection system, *etc.* are in need of using sensor nodes. A sensor node is an entity used to sense the physical phenomenon of a network. The phenomenon can be temperature, pressure, humidity, body sensed values, gas, and soil. A sensor network is useful in numerous applications because of its highlighted features like its size, power, cost, simplicity, fast transmission, and broadcasting [1]. It is widely used in applications in health, home utilities, military, and industries. The most challenging part of a Wireless Sensor Network (WSN) used for research is its battery life or power consumption. To perform fast transmission of data with low energy consumption the networking schemes engaged cluster based working principles.

Clustering is defined as the grouping of some sensor nodes which are under the range or area of each other. Clustering can be performed using different strategies and different algorithms [18]. Various block-based, chain-based, tree-based, and grid-based clustering algorithms (CA) are used for grouping a given sensor network into different groups or

Received September 10, 2019; revised December 5, 2019; accepted December 27, 2019.
Communicated by Xiaohong Jiang.

clusters and thereby enhancing the performance. Block-based CA such as low-energy adaptive clustering-based hierarchy (LEACH) [21], hybrid energy-efficient distributed clustering (HEED) [22], and chain-based CA like power-efficient gathering in sensor information system (PEGASIS) [23] are proposed & proven to be effective methods for saving energy consumption during communication. The limitations of such conventional or classical algorithms are improved with some advanced algorithms such as energy-efficient hierarchical clustering (EEHC) [24], and energy-efficient low-energy adaptive clustering-based hierarchy (EE-LEACH). Another method to perform clustering is the use of a heuristic algorithm to select the cluster head (CH) optimally, as in the case of algorithms such as hyper hybrid clustering algorithm (HHCA), heuristic algorithm for clustering hierarchy (HACH) [25], and linked cluster algorithm (LCA). To extend this, there also exist some meta-heuristic and swarm intelligent (SI) algorithms that prove that the elected CH is located optimally and also use renowned optimization algorithms for finding the best CH *e.g.* PSO, CSO, and ACO (ant colony optimization). All optimization algorithms consider three parameters for electing CH such as maximum energy, short distance, and maximum node coverage. However, the selection of the optimal number of CHs in any clustering algorithm is an NP-hard problem [9] as it cannot aggregate the data by saving the energy consumption in polynomial time. Various optimization problems are also NP-complete; hence, they are difficult to solve. A perfect cluster is created when the parameters needed for clustering such as cluster size, delivery ratio, energy level and number of clusters are also optimized. These optimization algorithms (OAs) come with its own fitness function to find optimal CH selection where in those fitness functions satisfy the basic requirement for finding CH like energy, distance, and coverage.

The motivation behind the implementation of EERSS is optimal selection of nodes as a Sector Head (SH). The existing clustering approaches do not consider the distance between CH and sink node (SN) during the election CH. The other limitations such as redundant data transmission, unbalanced distribution of node density into a cluster are also not considered in the existing schemes. The contribution is to perform selection of SH using EERSS which considers the current energy level of the node, RSSI value of each node, distance between the one hop away node to SN, broadcast initiated path discovery (PD) & path selection (PS) in routing table and threshold-based election of SH in each iteration. Here, the EERSS which is deterministic algorithm for SH election is used as election of CH in each round consumes more energy and time in existing protocols.

The paper presents a novel approach for solving the optimal clustering problem using the CSO algorithm. The paper is organized as follows. Section 2 presents the description of the various optimization techniques used for comparative analysis. Section 3 explains the detail of the working of CSO technique in two scenarios. Section 4 presents a proposed clustering strategy called the sectoring scheme (EERSS). Section 5 describes the implementation of the proposed sectoring scheme using the CSO algorithm to show that the elected number of SHs is optimal. Section 6 discusses the results analysis of the proposed work in comparison with the existing works. Finally, the sections are concluded with a conclusion discussed.

2. OPTIMIZATION TECHNIQUES

The process to apply optimization technique on any problem will be returning the best or optimal solution to that problem. A swarm is a group of some wing insects. The combination of a swarm and optimization provides a result by applying some strategies to a group of insects, observing their behaviour, and determining the best solution. The most recent technique of optimization is observing the behaviour of nature-inspired resources [7] (*i.e.* nature-inspired Swarm Intelligence (SI) algorithms). SI based algorithms, nature-inspired algorithms, and bio-inspired algorithms are a part of each other, as shown in Fig. 1. Different optimization techniques [15, 17] are used by introducing various innovations. Fig. 2 shows the different types of optimization techniques. They are evolutionary algorithms, bio-inspired algorithms, physic and chemical algorithms, swarm-based algorithms, human-based algorithms, and nature-inspired algorithms.

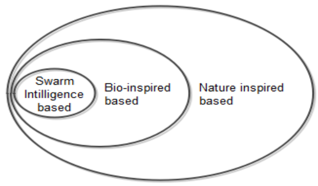


Fig. 1. Swarm optimization algorithms.

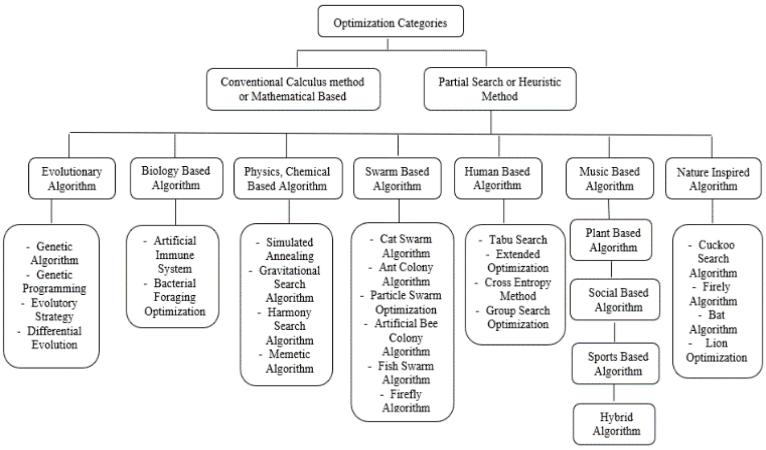


Fig. 2. Taxonomy of optimization techniques.

Table 1 describes the various types of OAs and the behavior of nature-inspired phenomenon. Each OA has its own characteristics that are used to solve a given problem and depending on this problem the algorithm may perform better than the other. In EERSS, the problem is to find the best node as SH from a pool of one hop away sensor nodes. The nodes need to be in sleep mode when event has not occurred. After the occurrence of the event, nodes come to the live mode and initiate or take part in the transmission. The parameters used for implementation of EERSS are explained in Section 1. It would be efficient to select an SH to whom a quick communication happens with path discovery and path selection. To satisfy the above parameters the OAs being used, PSO is one of the well-known OA, it is based on a stochastic optimization technique (random constraints) and works sequentially. PSO takes less memory for execution. It also provides optimized

Table 1. Types of swarm optimization algorithm (SOA).

SOA	Ref.	Year	Author	Behaviour of Swarm
Particle Swarm Optimiza-tion (PSO)	12	2009	Millie Pant *et al.*	It is motivated by the functioning of factual particles as a swarm. These particles are moved from place to place for searching the space using the particle position and velocity to find a local best-known position.
Ant Col-ony Opti-mization (ACO)	14	2010	V. Selvi *et al.*	It is motivated by the functioning of factual ants as a swarm. The purpose is to obtain the optimal or a good path from a graph based on the behaviour of ants, searching a path towards the colony.
Cat swarm Optimiza-tion (CSO)	13	2006	S.-A. Chu *et al.*	It is motivated by the functioning of factual cats as a swarm. The SM & TM of a cat for focusing on a target are used to obtain the global optimal position of a cat using parameters such as the position, velocity, and fitness func-tion.
Monkey search Optimiza-tion (MSO)	10	2016	C. E. Klein *et al.*	It is motivated by the functioning of factual monkeys as a swarm. It is based on the highland up-hilling process of monkeys. It can solve some problems such as the features of non-linearity and high dimensionality with a slow junc-tion rate.
Fish swarm Optimiza-tion (FSO)	11	2002	X.-L. Li *et al.*	It is motivated by the functioning of factual fish as a swarm. The search is performed inspired by the swimming motion of fish. It uses the fish movement for moving to-wards a positive gradient to take food and gain weight.

CH selection process. The working of PSO is like while selecting a node locally from a pool which shows as local best node. After election of node as local best then it works on finding node as a global best. It is an ongoing process until the global best node gets elected. The PD & PS is happing while finding the node as local best and global best. This is quite time consuming and if event does not occur then also process will continue to find best node till end. It is difficult to define the initial parameters in PSO and it faces the problem of premature convergence. As it takes more time to check each and every possibility to become a local and global best node. Because of these reasons it is not suitable to solve the EERSS problems. The ACO works on movement of ants in a particular way to reach at food. If any obstacle occurs in between the search it changes its path. The main idea behind ACO is to find shortest and cheapest path to reach at destination. Here its best nodes selection is not based on shortest path. Hence it is not suitable to solve problems of EERSS. CSO is optimistic in nature and performs parallel execution; it considers few of the cats from pool as a population of cats. It doesn't consider any random value to initiate the working of technique. The initial parameters are their population of cats. The working can be considered in two modes such as seeking mode (SM) and tracing mode (TM), where initial PD & PS happens in first mode and election of best SH and initiation of packet transmissions in second mode. If the event is not occurred for long time the cats goes to sleep state in SM. So CSO gives more optimal solution than PSO and ACO because of its own SM & TM.

3. CAT SWARM OPTIMIZATION

The CSO algorithm [2-6] is divided into two different modes: 'seeking mode' and 'tracing mode'. The decision of each cat position is made using some important parameters like the fitness function, velocity of each dimension, and fitness value calculated by the fitness function in the resolution space. Each cat takes its own flag to represent its mode, (*i.e.* SM & TM). The quantity of the seeking flags and tracing flags is governed by the mixture ratio. The mixture ratio characterizes the proportion of the number of TM cats to that of the seeking mode cats. CSO offers improved performance by identifying the best global solution. The different modes or phases of the CSO algorithm are as follows:

(A) Initial set-up of cat placement: For the operation of CSO [8, 26], initially the scenario of how many cats, *i.e.* the population of cats, is needed to be decided. With the help of this population of cats, the CSO algorithm is executed for solving a particular problem. In all the iteration of CSO, a predefined number of cats need to be considered, as shown in Fig. 3. The characteristics assigned to each cat are position, *i.e.* its N-dimensional space in which the cat has to be located, velocity of each cat dimension, the fitness function to place the cat in each iteration, and flag to decide the mode of the cat. Finally, after considering all these characteristics, the output is provided in terms of the best place or position of one of the cats to be seated. The algorithm is executed until it reaches the next best position; otherwise it retains the previous one as the best position.

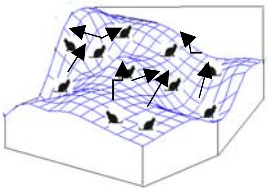


Fig. 3. Locating cats in an *N*-dimensional space.

Fig. 4. Seeking phase for sleeping or waiting.

Fig. 5. Tracing phase for chasing.

(B) The seeking mode of the CSO algorithm is used to represent the current position or condition of a cat. The condition includes sleeping, alive but observing the surrounding, and searching for the next residence to transfer to, as illustrated in Fig. 4. This mode works with the following four major parameters: Parameter 1: Seeking memory pool (SMP): The cats which are searching for the next place need to have an idea about which place is required to be shifted. The SMP provides an indication about its size for each cat, denoted as N_{SMP} with the neighbouring place of the cat. Therefore, as per the requirement and need, each cat takes the place from the SMP. Parameter 2: Seeking a range of the selected dimension (SRD): It represents a standard value as a range for a cat to select a new place. If a particular dimension is selected as a new place, then the difference between the new and old places should not be more than the SRD value. Parameter 3: Counts of the dimension to change (CDC): In all the iteration the cat can find its new best dimension. The CDC represents the value of the quantity of the dimension to be mixed. Here, the default value for CDC can be set. Parameter 4: Self-position considering (SPC): The SPC parameter represents the current position of each cat. It also directs that the current position of a particular cat will be the new position of a neighbouring cat. To provide a signal for the motion of the cat, the SPC initializes a Boolean value flag as true or false, which is represented as N_{SPC} which is either 0 or 1. If the signal is true then the particular cat sees a new neighbouring position; otherwise, it will be assigned a different position.

The working of SM with these four parameters is explained in the following steps in detail:

Step 1: 'i' copies of the current position Cat_t are created, where $i = N_{SMP}$. If the current value of the SPC is true, *i.e.*, $N_{SPC} = 1$, then $i = N_{SMP} - 1$ is set. It is represented in Eq. (1).

$$cat_{i\_copies} = \begin{Bmatrix} N_{SMP-1} & \text{if } N_{SPC} = 1 \\ N_{SMP} & \text{if } N_{SPC} = 0 \end{Bmatrix} \tag{1}$$

Step 2: Here, the copies created in the first step of SM, for each copy randomly $\pm N_{SRD}$ percent of the current value and replace the old one. This can be easily represented using Eq. (2).

$$P_{new} = (1 \pm N_{SRD} \times r_{(0,1)} \times P_{curr}) \tag{2}$$

P_{curr} = current position of the cat, N_{SRD} = seeking range of a particular dimension, P_{new} = new position of the cat, $r_{(0,1)}$ = random value that lies between 0 and 1.

Step 3: The fitness values (N_{FV}) of the entire candidate positions are determined.
Step 4: If the calculated N_{FV} of all the cats are not equal, then the selecting probability of each cat position using Eq. (3) is determined; otherwise, the selecting probability of each cat position is set as 1.

$$P_i = \frac{|FV_i - FV_{best}|}{|FV_{\max} - FV_{\min}|}, \quad \text{where } 0 < i < t \tag{3}$$

where P_i = probability of the current candidate, cat_t, FV_i = fitness value of cat_t. FV_{max} = maximum value of the fitness function, FV_{min} = minimum value of the fitness function, $FV_{best} = FV_{max}$ for the maximization problem, $FV_{best} = FV_{mix}$ for the minimization problem.
Step 5: Newly created solution is chosen to move and transfer the position of Cat_t.

(C) Tracing Mode: The tracing mode of the CSO algorithm is shown in Fig. 5. In this mode, a cat is trying to reach the target value or place for finding the best position. When a cat enters the tracing mode, it travels as per its individual velocity for each dimension. The working of the tracing mode of the CSO algorithm is explained using the following steps.

Step 1: The velocities for each dimension $v_{(t,d)}$ are updated using Eq. (4).

$$v_{(r,d)} = v_{(r,d)} + r_0 + c_0 \times (X_{(best,d)} - X_{(r,d)}), \text{ where } d = 1, 2, \ldots, N \tag{4}$$

$x_{(best,d)}$ is the position of the cat having the best fitness value, $x_{(t,d)}$ is the position of Cat_t, c_0 is a constant, and r_0 is a random value in the range of (0,1).

Step 2: The velocity calculated by Eq. (4) needs to be checked to determine if it is in the range of the maximum velocity. However, if the new velocity is beyond the range, then it is set equal to the limit.
Step 3: The position Cat_t is updated using Eq. (5).

$$x_{(t,d)} = x_{(t,d)} + v_{(t,d)} \tag{5}$$

4. ENERGY EFFICIENT RELIABLE SECTORING SCHEME WITH CSO

A sectoring scheme or method is used to partition a given scenario into a different number of sectors. Here, sectoring performs the grouping of sensor nodes into one group with suitable characteristics. Initially, the partitioning is performed in the presence of a base station (BS) or SN. The BS decides which sensor node can be the SH for a particular number of iterations based on some parameters such as the RSSI value, distance, and remaining energy of all the one hop away (OHA) nodes. Initially, the BS collects this information only from the OHA nodes that form it. A short distance, large RSSI value, and more remaining energy must be calculated while selecting the OHA nodes from the sink. After the selection, the BS broadcasts a *HELLO* message to all the sensor networks to connect with all the nodes. This *HELLO* message is always transmitted with either of the selected OHA nodes to the network. The sensor nodes in a network, acknowledge the *HELLO* packet with their own information. This reverse communication occurs with either of the OHA nodes from the sink. This routing path of communication is to be selected for logical sectoring of the network, as shown in Fig. 6. These elected OHA nodes are used as the SHs for communication. In this study, the use of the OA is to prove that the elected number of SHs and location are optimal in nature.

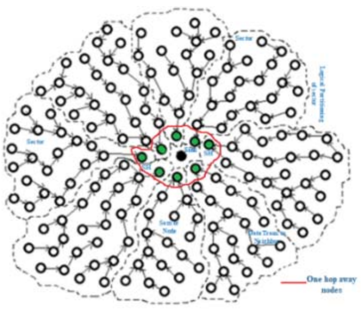


Fig. 6. Scenario of the sectoring scheme in WSN.

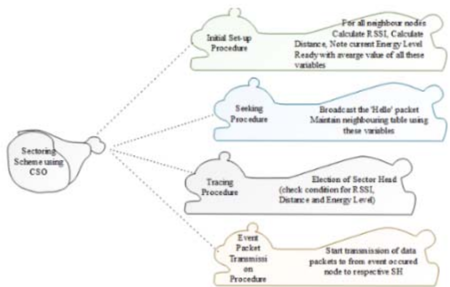


Fig. 7. Workflow of the CSOSS.

For this purpose, one of the best OAs, *i.e.* CSO, is used for the implementation of the sectoring scheme. Fig. 7 shows the working of sectoring scheme with CSO for implementation. The flowchart of Cat Swarm Optimization with Sectoring Scheme (CSOSS) where each step *i.e.* initial setup, seeking procedure, tracing procedure and event packet transmission procedure is shown in Fig. 8.

Step 1: Initial Setup CSOSS

Algorithm: Mechanism of Cat Swarm Optimization with Sectoring Scheme (CSOSS)

Input: pool of sensor nodes presents in a network.
Output: nodes has greater RSSI value and lesser distance from sink
1: **for** all neighbour nodes in network do
2: Compute distance of each node from the sink using Eq. (6).
3: Compute RSSI values all nodes from the sink using Eq. (7).
4: **end** for
5: Calculate average distance from the sink using Eq. (8).
6: Calculate average RSSI from the sink using Eq. (9).
7: return the nodes which satisfies condition in Eqs. (8) and (9).

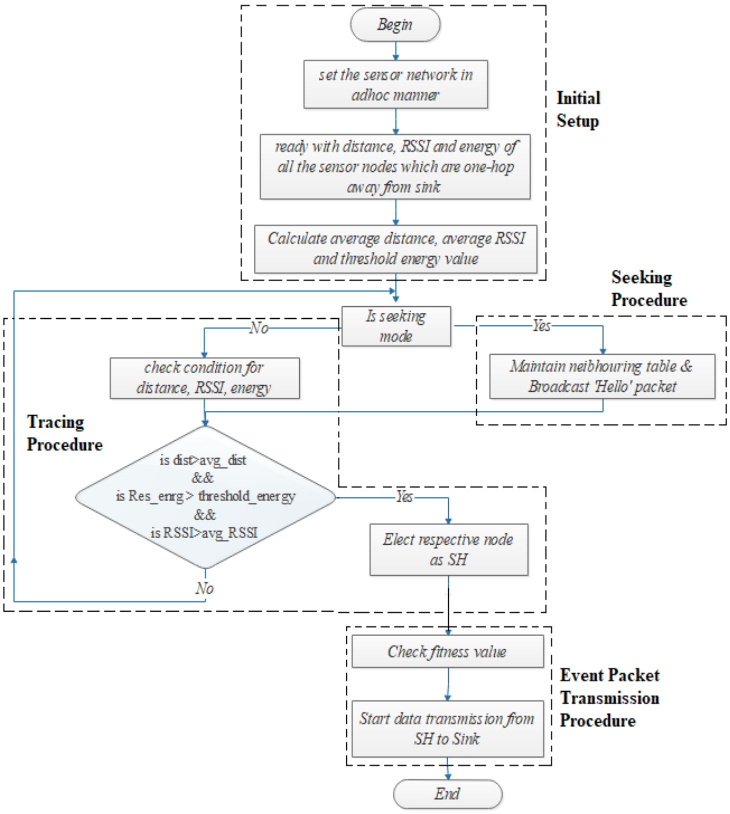


Fig. 8. Flowchart of execution of the sectoring scheme using CSO.

$$ED(node_i, SINK_{(s_0,s_1)}) = \sqrt{(x_i - s_0)^2 + (y_i - s_1)^2} \tag{6}$$

$$RSSI_{node_i} = \frac{R \times P}{Distance} \tag{7}$$

$$d_{avg} = \sum_{p=0}^{i_{node}} \sqrt{(x_0 - s_0)^2 + (y_0 - s_1)^2} + \sqrt{(x_1 - s_0)^2 + (y_1 - s_1)^2} + \ldots + \sqrt{(x_{i_{node}} - s_0)^2 + (y_{i_{node}} - s_1)^2} \tag{8}$$

$$RSSI_{avg} = \sum_{p=0}^{i_{node}} \frac{RSSI_0 + RSSI_1 + RSSI + ... + RSSI_{i_{node}}}{i_{node}} \tag{9}$$

$$[(RSSI_{node_i} > RSSI_{avg}) \; \&\& \; (ED(i_{node}, SINK) < d_{avg})] \tag{10}$$

where

$ED(node_i, SINK_{(s_0,s_1)})$ = Euclidian distance between the sensor node and the sink.

$RSSI_{node_i}$ = receiving signal strength identificatory of sensor node i,

$R \times P$ = receiving power

d_{avg} = average distance between each one hop away node from the sink to the node

$RSSI_{avg}$ = average $RSSI$ of each sensor node towards the sink and node.

Step 2: Seeking Mode for broadcasting a message

Step 2: Seeking Procedure

Input: set of one hop away nodes from sink (N)
Output: updated neighbour table
for every 3 seconds **do**
  for all N nodes **do**
    broadcast *HELLO* packet with the location and energy.
  end of for
end of for
nodes that are in the range of N nodes accepts the *HELLO* packet
The neighbour table is maintained.
if the node did not receive the *HELLO* packet from a neighbour within 3s
    the node entry is removed from the neighbour table.
else the expire time and energy of the neighbour node are updated.
end of if

Step 3: Tracing Mode for SH Selection

In the original/basic CSO algorithm, the TM is used for finding the best solution as per the velocity of the sensor node in every dimension. In this phase of sectoring scheme, consider the entire sensor nodes which are one hop away from the SN for the election of SH. List out the RSSI value, Euclidian distance, and energy level of each sensor node from the SN. Finally, by receiving all these values, the phase will get the sensor nodes which are nearer to the SN and that, having more energy level can be elected as an SH. After the selection of the SH, the SN broadcasts the message of the selection of the node as an SH to all other nodes for the formation of sectors. The downlink communication starts between the SN and sensor nodes. Each sensor node uses the shortest path for sending the packet to the next node. Once transmission occurs through any of the SH, respective SH transmits a reverse reply message which consists of an SH notification for finalizing the sectoring range, R_s. At this point, each sensor node considers the R_{max} values *i.e.* maximum transmission range. With the help of this, logical angles are drawn for each SH, for transmission. Further, the selection of probable SHs is done.

Step 3: Tracing Procedure

Input: neighbour table with energy, distance and RSSI value of nodes.
Output: set of elected SHs.
if $node\_energy \leq threshold\_value$

2: *node* = Invalid.
3: **else** for all valid nodes do
4: **if** node value ≥ the average distance, RSSI & energy (using Eq. (10))
5: node = SH
6: store R_s value of elected SH.
7: **else** node can't be sector head.
8: **while** (Position of SH = X_best)
9: **end of for**
10: **end of if**
11: **end of if**
12: sector formation using stored path.
13: perform data transformation once event occurred through elected SH.
14: **if** SH_*energy* ≤ *threshold_energy*
15: SH = invalid
16: **goto** 3
17: **end of it**

Step 4: Event packet transmission procedure

Initially, all the sensor nodes are in a sleep mode. Once the event occurs for a particular sector it activates the entire sector nodes from it. To perform this, activate the variable $Radio_{ON}$ and then start transmission between them. Finally, SH aggregates the packets sent by the event occurred node and sends it to the sink for the next action. The working of the implementation of the CSOSS is represented in Fig. 9. Each round consists the updated neighbouring table information in seeking procedure; in tracing procedure, the SH is selected with the formation of the sector, and in the event packet transmission procedure, transmission of data from the sensor node to SH and from SH to sink is performed.

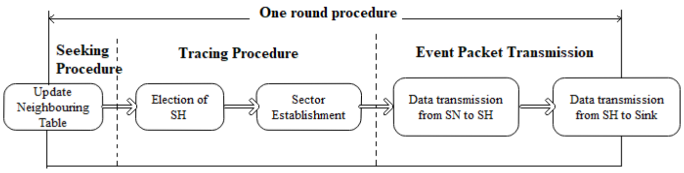


Fig. 9. One round procedure of the sectoring scheme using the CSO technique.

5. FORMATION OF OBJECTIVE FUNCTION

The objective function of a network is defined to set a target value with some predefined situation and parameters. Alternatively, we can also call the objective function as a fitness function (FF) of a network to represent our proposed network at a glance. The relationship of a desired output and FFs are closely related to each other. In short, FF representing the proposed network model is giving the best or optimal solution in particular circumstances. The objective function is used to represent the proposed model mathematically, to prove that the selection of parameters is optimal or best. In a proposed sectoring scheme, position of SH is one hop away from sink *i.e.* one level communication occurs. Similarly, the position of SH and SN is multi-hop *i.e.* multi-level communication takes place. The selection of SH in each iteration with optimality is not an easy task. To select an optimal SH, the FF is proposed in this section. In each iteration, the location of each SH is not defined. The location of SH is one-hop away from the SN. The selected SH using fitness function is of less distance SN, having more receiving strength for transmission, and having more residual energy to survive in a network. Here, we have proposed a fitness function of sectoring scheme using a CSO. f_{obj} gives how the fitness value is calculated as shown in Fig. 10 and calculated using Eq. (11). The fitness function is derived for achieving multiple objectives like more reliability, less delay and low energy consumption. The aim of f_{obj} is to optimize these parameters simultaneously. For this purpose, the objectives can be defined to solve multi-objective problem [27-30] of network. The f_1, f_2, and f_3 are three subfunctions used to measure distance, RSSI value, and energy consumption, respectively are mathematically formulated Eqs. (12)-(14). Using f_1, the network will find the distance using Euclidean formula of each node from the SN. By minimizing f_1 we will get the sensor nodes that are nearer to the SN. f_2 gives a sensor node that has the highest strength of the received signal for the transmission of data. f_3 provides a sensor node which is having more residual energy to transmit the data.

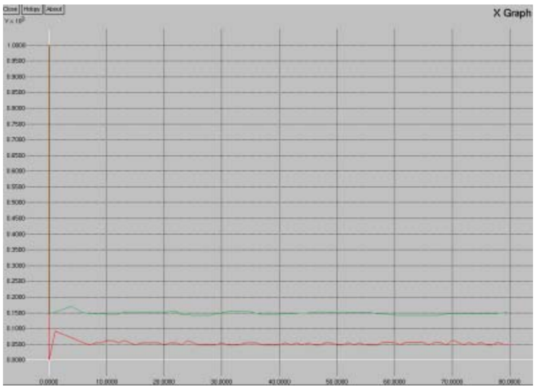


Fig. 10. Fitness function as a function of the number of rounds.

$$f_{obj} = c * (f_1 + f_2) + (1 - c) * f_3 \tag{11}$$

$$f_1 = \min_{j \in (1,n)} \left\{ \sum\nolimits_{\forall node \in hop_1} ED(node_i, SINK_{(s_0, s_1)}) \right\} \tag{12}$$

$$f_2 = \max_{j \in (1,n)} \left\{ \sum\nolimits_{\forall node \in hop_1} RSSI_{node_i} \right\} \tag{13}$$

$$f_3 = \min \sum\nolimits_{i=1}^{n} \left\{ \frac{E_{res}^{i}}{E_{res}^{SH}} \right\} \tag{14}$$

Where, $ED(node_i, SINK_{(s_0,s_1)})$ = Euclidean distance of ith node to SN

$RSSI_{node_i}$ = received signal strength indicator of ith node

E_{res_i} = residual energy of ith node, $E_{res_{SH}}$ = current energy level of SH

c = a constant value showing involvement of it to an objective function

ith *node* = All one hop away nodes from sink

6. PERFORMANCE EVALUATION

6.1 Simulation Parameters

The simulation of the EERSS algorithm is conducted and analysed in Network Simulator-2 (NS-2) [19-20]. Similar to EERSS, PEGASIS, PSO-based SS (PSO-EERSS), and CSO-based SS (CSO-EERSS) are implemented and the simulation results are observed. The scenario considers the network population as 100, reporting rate as 10 pkt/s and packet size as 50 bytes.

The comparison of basic sectoring scheme simulation results is done with the implementation of SSS with two OAs like PSO and CSO using packet delivery ratio (PDR*)*, energy consumption, and an end to end delay. The packet delivery ratio is measured using the number of packets received with the number of packets sent towards the network. PDR is an important factor to analyse the reliability of the network. Packets are lost because of the size of the buffer or if any congestion occurs in a network. The formula for calculation of PDR is shown in Eq. (15). Another parameter used for analysing the performance is the energy consumption of a sensor node. The total number of energies utilized for communication is to be measured using Eq. (16). A sensor node utilizing less energy for transmission and receiving of the packet is to be considered as a perfect network partitioning. It is the difference between the current energy level and the initial energy level of a sensor node. Eq. (17) represents the average end to end (E2E) delay required during transmission of a packet from the sender to receiver. It considers the successfully received packet at the destination. E2E delay is measured as the time required for successfully sending and receiving the packet at the destination.

$$Packet\ Delivery\ Ratio = \frac{\sum_{i=0}^{n} No\ of\ Packets\ Received}{\sum_{i=0}^{n} No\ of\ Packets\ sent} \times 100 \tag{15}$$

$$Energy\ Consumption = Current\ Energy\ Level - Initial\ Energy\ Level \tag{16}$$

$$Average\ End\ To\ End\ Delay = \sum_{i=0}^{n} \frac{Receiving\ Time - Sending\ Time}{n} \times 1000 \tag{17}$$

Here in these equations, n is a total number of sensor nodes that participated during the transmission of a packet from source to destination. The results are calculated using a chain based partitioning algorithm (EERSS), compared with the existing chain-based clustering protocol (PEGASIS), and provides the best OA used for getting an optimal solution in iteration (PSO & CSO). Basic EERSS is implemented using PSO and CSO techniques.

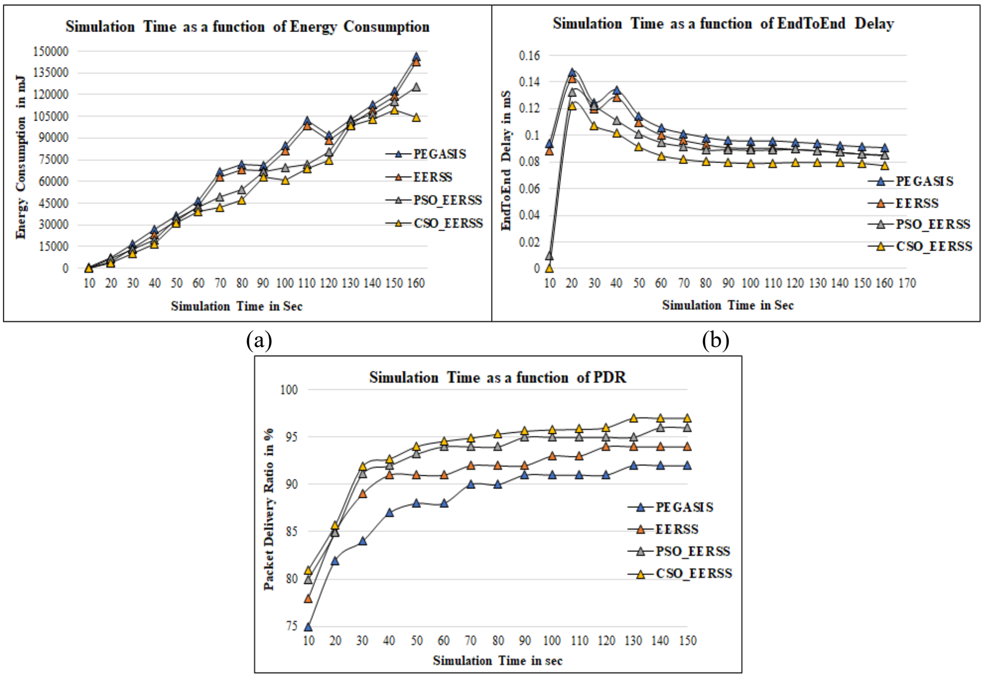
6.2 Simulation Results

6.2.1 Analysis of energy consumption

Fig. 11 (a) shows the analysis of energy consumptions with respect to the simulation time (ST). It has been observed that when packet transmission each protocol takes more energy as there is a change in ST. The energy required for transmission and receiving a packet along with communication needs more energy for PEGASIS than EERSS protocol because PEGASIS performs election of CH on random constraints wherever in a chain, the PS &PD doesn't work well in it. Whereas in EERSS, takes lesser energy because of the sector selection which helps to select SH who has less distance from sink and having more energy than threshold. And if any sector has come below the threshold energy then it automatically switches to other SHs. PSO based EERSS takes lesser than EERSS because of optimization applied on selection of best nodes locally and globally. Finally, CSO based EERSS takes as least energy as in SM the nodes take rest before occurrence of event.

6.2.2 Analysis of packet delivery ratio

Fig. 11 (b) shows the analysis of the PDR with respect to the ST. The PDR is analysed for a provided sensor network. CSO based EERSS protocol provides approximately 97% of reliability. Increase in 5% of packet delivery ratio as compared to the other three routing protocols. In CSO based EERSS, the larger number of packets getting transferred as routing path was decided initially for transmission. This algorithm starts the transmission of packets when PD & PS is completed. The neighbour table and routing table updated in SM of algorithm. So redefine path with best nodes and SH gives better PDR than other three protocols. As power consumption is less in CSOEERSS it helps to improve the lifetime of the network.



(c)

Fig. 11. (a) Energy consumption as a function of ST; (b) Packet delivery ratio as a function of ST; (c) Average end to end delay as a function of ST.

6.2.3 Analysis of average end to end delay

Fig. 11 (c) shows the analysis of E2E delay with respect to the ST. The PD, as well as the selection of SHs, are conducted in the initial step of EERSS algorithm, and hence, the packets are transmitted to the destination node quickly. The PD & PS maintain the updated neighbour and routing table. In CSO based EERSS, the SM preforms broadcasting of *HELLO* packet to the network. This helps to maintain quality path for transmission of event generated packets. As predefine route, well-known neighbours, healthy nodes make transmission easily. Because of this simulation results shows that CSO based EERSS takes less time when compared to the other three protocols during communication.

7. CONCLUSION

The optimization models make use of bio-inspired algorithms for getting an optimal solution for various problems. These algorithms analyse the behaviour of animals or insects and provide an optimal solution for the problems in real world. CSO is one of the appreciated OAs used to predict the finest global solution of an objective function. The proposed EERSS designed to overcome many of the limitations of the existing schemes. The EERSS algorithm considers the optimal parameters for electing the SHs such as RSSI value, distance, energy level *etc.* which makes it to perform well compared with other existing schemes. The scheme always ensures that the communication with sink node happens in one hop and formation of the sector is a one-time process by selecting or recovering the routing path. The performance of the proposed scheme is compared with the existing chain-based schemes such as PEGASIS. In addition, the other appreciated OA, PSO is also implemented in EERSS algorithm and the performance is analysed. The results analysis based on performance metrics such as packet delivery ratio, energy consumption, and average end to end delay shows that CSO based EERSS outperforms other schemes. The results confirm that CSO based EERSS provides better reliability as the packet drops are reduced and it also consumes less energy. Moreover, the delay in communication is drastically reduced in the proposed scheme. So, the proposed scheme emphasizes the need of multi-objective-based modelling for optimised solution of selecting SH and initiating communication with sink node. More exploration of multi-objective modelling is the future scope of the work.

REFERENCES

1. F. Akyildiz, S. Weilian, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, Vol. 40, 2002, pp. 102-114.
2. G. Ram, D. Mandal, R. Kar, and S. P. Ghoshal, "Cat swarm optimization as applied to time-modulated concentric circular antenna array: Analysis and comparison with other stochastic optimization methods," *IEEE Transactions on Antennas and Propagation*, Vol. 63, 2015, pp. 4180-4183.
3. L. Pappula and D. Ghosh, "Unequally spaced linear antenna array synthesis using multi-objective cauchy mutated cat swarm optimization," in *Proceedings of IEEE International Symposium on Antennas and Propagation*, 2017, pp. 313-314.

4. M. Bahrami, O. Bozorg-Haddad, and X. Chu, "Cat swarm optimization algorithm," *Studies in Computational Intelligence*, Book Chapter, 2017, pp. 9-18.
5. J. Wang, "A new cat swarm optimization with adaptive parameter control," *Genetic and Evolutionary Computing*, 2015, pp. 69-78.
6. L. Kong, *et al.*, "An energy-aware routing protocol using cat swarm optimization for wireless sensor networks," *Advanced Technologies*, *Embedded*, *and Multimedia for Human-Centric Computing*, Springer, Netherlands, 2014, pp. 311-318.
7. A. Slowik and H. Kwasnicka, "Nature inspired methods and their industry applications − Swarm intelligence algorithms," *IEEE Transactions on Industrial Informatics*, Vol. 14, 2018, pp. 1004-1015.
8. D. Chandirasekaran and T. Jayabarathi, "Cat swarm algorithm in WSN for optimized cluster head selection: a real-time approach," *Cluster Computing*, Vol. 22, 2017, pp. S11351-S1136.
9. N. M. A. Latiff, C. C. Tsimenidis, and B. S. Sharif, "Performance comparison of optimization algorithms for clustering in wireless sensor network," in *Proceedings of IEEE International Conference on Mobile Adhoc and Sensor Systems*, 2007, pp. ___.
10. C. E. Klein, E. H. V. Segundo, V. C. Mariani, and L. dos S. Coelho, "Modified social-spider optimization algorithm applied to electromagnetic optimization," *IEEE Transactions on Magnetics*, Vol. 52, 2016, pp. 1-4.
11. X.-L. Li, Z.-J. Shao, and J.-X. Qian, "Optimizing method based on autonomous animates Fish-swarm algorithm," *System Engineering Theory and Practice*, Vol. 22, 2002, p. 32.
12. M. Pant, R. Thangaraj, and A. Abraham, "Particle swarm optimization: Performance tuning and empirical analysis," *Foundations of Computational Intelligence*, Vol. 3, 2009, pp. 101-128.
13. S.-A. Chu, P.-W. Tsai, and J.-S. Pan, "Cat swarm optimization," in *Proceedings of Pacific Rim International Conference on Artificial Intelligence*, LANI 4099, 2006, pp. 854-858.
14. V. Selvi and Dr. R. Umarani, "Comparative analysis of ant colony and particle swarm optimization techniques," *International Journal of Computer Applications*, Vol. 5, 2010, pp. ______.
15. A. Wahab, M. Nadhir, S. Nefti-Meziani, and A. Atyabi, "A comprehensive review of swarm optimization algorithms," *PLoS One*, 2015, Vol. 10, p. e0122827.
16. A. Chakraborty and K. Arpan, "Swarm intelligence: A review of algorithms," *Nature-Inspired Computing and Optimization*, Springer, Berlin, 2017, pp. 475-494.
17. L. Xu, R. Collier, and G. M. P. O'Hare, "A survey of clustering techniques in WSNs and consideration of the challenges of applying such to 5G IoT scenarios," *IEEE Internet of Things Journal*, Vol. 4, 2017, pp. 1229-1249.
18. M. Amjad, M. K. Afzal, T. Umer, and B. Kim, "QoS-aware and heterogeneously clustered routing protocol for wireless sensor networks," *IEEE Access*, Vol. 5, 2017, pp. 10250-10262.
19. K. Fall and K. Varadhan, *The NS Manual*, The VINT Project, Berkley, USA 2009.
20. T. Issariyakul and E. Hossain, *Introduction to Network Simulator*, Springer, USA, 2009.
21. S. K. Singh, P. Kumar, and J. P. Singh, "A survey on successors of LEACH protocol," *IEEE Access*, Vol. 5, 2017, pp. 4298-4328.
22. O. Younis and S. Fahmy, "HEED: a hybrid, energy-efficient, distributed clustering

approach for ad hoc sensor networks," *IEEE Transactions on Mobile Computing*, Vol. 3, 2004, pp. 366-379.
23. A. R. Sankaliya, "PEGASIS: Power-efficient gathering in sensor information systems," *International Journal of Scientific Research in Science and Technology*, Vol. 1, 2015, pp. 108-112.
24. B. Jan, H. Farman, H. Javed, B. Montrucchio, M. Khan, and S. Ali, "Energy efficient hierarchical clustering approaches in wireless sensor networks: A survey," *Wireless Communications and Mobile Computing*, 2017, p. 14.
25. M. O. Oladimeji, M. Turkey, and S. Dudley, "HACH: Heuristic algorithm for clustering hierarchy protocol in wireless sensor networks," *Applied Soft Computing*, Vol. 55, 2017, pp. 452-461.
26. B. Santosa and M. K. Ningrum, "Cat swarm optimization for clustering," in *Proceedings of Conference on Soft Computing and Pattern Recognition*, 2009, pp. 54-59.
27. L. Pan, L. Li, R. Cheng, C. He, and K. C. Tan, "Manifold learning-inspired mating restriction for evolutionary multi-objective optimization with complicated pareto sets," *IEEE Transactions on Cybernetics*, Vol. ___, 2019, pp. _____.
28. M. Iqbal, M. Naeem, A. Anpalagan, N. N. Qadri, and M. Imran, "Multi-objective optimization in sensor networks," *Computer Networks*, Vol. ____, 2016, pp. 134-161.
29. L. Pan, C. He, Y. Tian, H. Wang, X. Zhang, and Y. Jin, "A classification-based surrogate-assisted evolutionary algorithm for expensive many-objective optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 23, 2019, pp. 74-88.
30. L. Pan, L. Li, C. He, and K. C. Tan, "A subregion division-based evolutionary algorithm with effective mating selection for many-objective optimization," *IEEE Transactions on Cybernetics*, Vol. ___, 2019, pp. _____.

Dhanashri Narayan Wategaonkar received the Bachelor in Computer Science and Engineering and completed her Master of Engineering with distinction in IT from MIT College of Engineering. She has 10 years of experience and working as an Assistant Professor in MITCOE, Pune, India and pursuing Ph.D. at VIT Chennai, India. Her research interest includes wireless sensor network.

T. R. Reshmi received her Ph.D. in 2015 under the Faculty of Information and Communication Engineering at Anna University, Chennai, India. She is currently working as a Senior Assistant Professor in VIT University, Chennai, India. She is an IPv6 Forum Certified Engineer (Silver) and a Cisco Networking Academy Instructor. She has authored many reputed journal research articles and book chapters. Her research area includes wireless networks, next generation networks, QoS, network security and service applications.