G Model
ASOC-2497; No. of Pages 12

ARTICLE IN PRESS

Applied Soft Computing xxx (2014) xxx–xxx

Contents lists available at ScienceDirect

Applied Soft Computing

journal homepage: www.elsevier.com/locate/asoc

A novel differential evolution based clustering algorithm for wireless sensor networks

Pratyay Kuila, Prasanta K. Jana*,[1]

Department of Computer Science and Engineering, Indian School of Mines, Dhanbad 826 004, India

ARTICLE INFO

Article history:
Received 11 July 2013
Received in revised form 30 April 2014
Accepted 1 August 2014
Available online xxx

Keywords:
Wireless sensor networks
Clustering
Differential evolution
Network life

ABSTRACT

Clustering is an efficient topology control method which balances the traffic load of the sensor nodes and improves the overall scalability and the life time of the wireless sensor networks (WSNs). However, in a cluster based WSN, the cluster heads (CHs) consume more energy due to extra work load of receiving the sensed data, data aggregation and transmission of aggregated data to the base station. Moreover, improper formation of clusters can make some CHs overloaded with high number of sensor nodes. This overload may lead to quick death of the CHs and thus partitions the network and thereby degrade the overall performance of the WSN. It is worthwhile to note that the computational complexity of finding optimum cluster for a large scale WSN is very high by a brute force approach. In this paper, we propose a novel differential evolution (DE) based clustering algorithm for WSNs to prolong lifetime of the network by preventing faster death of the highly loaded CHs. We incorporate a local improvement phase to the traditional DE for faster convergence and better performance of our proposed algorithm. We perform extensive simulation of the proposed algorithm. The experimental results demonstrate the efficiency of the proposed algorithm.

1. Introduction

The radical advances in micro-electro-mechanical-system (MEMS) and wireless communication technology enable the development of wireless sensor networks (WSNs). WSNs have attracted enormous attention for their potential applications in diverse areas such as disaster warning systems, environment monitoring, health care, safety, surveillance, intruder detection, etc. [1,2]. A WSN consists of a large number of tiny, low power and inexpensive sensor nodes, which are randomly or manually deployed over an unattended target area. The sensor nodes are equipped with sensing, processing and communication component along with a power unit. These sensor nodes periodically collect local information of the targets, process the data and finally send it to a remote base station (called sink). The sink is connected to the Internet for the public notice of the phenomena.

1.1. Motivation

The main constraint of a WSN is the limited and irreplaceable power source of the sensor nodes. Moreover, in many applications, it is almost impossible to replace the sensor nodes when their energy is completely depleted. Therefore, energy consumption for the sensor nodes is the most challenging issue for the long run operation of WSNs [3,4]. Various issues have been studied for this purpose that includes low-power radio communication hardware [5], energy-aware medium access control (MAC) layer protocols [6,7], etc. However, clustering is the most effective technique for energy saving of the sensor nodes.

In a cluster based architecture, the sensor nodes are divided into several groups called clusters. Each cluster has a leader known as cluster head (CH). All the sensor nodes sense local data and send them to their corresponding cluster head. The CHs then aggregate the local data and finally send it to the base station directly or via other CHs. The functionality of a cluster based WSN can be seen in Fig. 1. A cluster based WSN has the following advantages [8,9]:

(1) It enables data aggregation at cluster head to discard the redundant and uncorrelated data. It should be noted that the energy

* Corresponding author. Tel.: +91 0326 2235473.
E-mail addresses: pratyay_kuila@yahoo.com (P. Kuila), prasantajana@yahoo.com (P.K. Jana).
[1] *IEEE Senior Member*

http://dx.doi.org/10.1016/j.asoc.2014.08.064
1568-4946/

Please cite this article in press as: P. Kuila, P.K. Jana, A novel differential evolution based clustering algorithm for wireless sensor networks, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.08.064

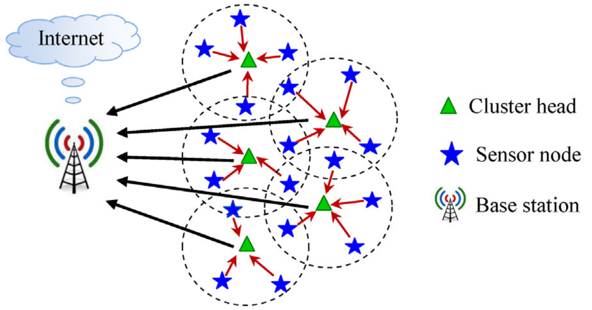

Fig. 1. A wireless sensor network model.

consumed to transfer one bit of data can be used in high volume of data aggregation. Thus it reduces energy consumption of the network by preventing to transmit high volume of redundant data rather than aggregated data.

(2) Routing can be more easily managed because only CHs need to maintain the local route set up of other CHs and thus require small routing information. This in turn improves the scalability of the network significantly.

(3) It also conserves communication bandwidth as the sensor nodes communicate with their CHs only and thus avoid exchange of redundant messages among themselves.

However, in a cluster based approach, CHs bear some extra work load owing to receiving the sensed data, data aggregation and communication with the BS. Moreover, in many WSNs, the CHs are usually selected amongst the normal sensor nodes which can die quickly for this extra work load. In this context, many researchers [9–14] have proposed the use of some special nodes called gateways or relay nodes, which are provisioned with extra energy. These gateways are treated as the cluster heads which are responsible for the same functionality of the CHs and therefore they can be used interchangeably for the rest of the paper. Unfortunately, the gateways are also battery operated and hence power constrained. Life time of the gateways is very crucial for the long run operation of the network. Moreover, if the sensor nodes are not properly assigned to the gateways for cluster formation, then some gateways may be overloaded with high number of sensor nodes. Such overload may lead initial death of these gateways and also increase latency in communication and degrade the overall performance of the WSN. Particularly, this is a pressing issue when the sensor nodes are not distributed uniformly. Therefore, proper assignment of the sensor nodes to the gateways is very crucial for balancing the energy consumption of the gateways. This is also worth noting that in order to balance the energy consumption of the gateways, some sensor nodes are assigned to a gateway which may be farther from it. As a result, their energy may be drained out due to long haul transmission and may die quickly. Therefore, while designing clustering algorithm, one should take care not only the energy consumption of the CHs but also energy consumption of the sensor nodes to increase the network life time. It is noteworthy that for a WSN, with n sensor nodes and m gateways, the number of possible clusters is m^n. This implies that the computational complexity to find out optimal cluster for a large WSN seems to be very high by a brute force approach.

In this paper, we address the following problem: given a WSN with n sensor nodes and m gateways find the optimal clusters of the sensor nodes surrounding the gateways so that the network lifetime is maximized. Differential evolution (DE) [15,16] is one of the most suitable heuristics that can be applied to solve this problem when the solution space is very large.

1.2. Our contribution

In this paper, we propose a novel differential evolution (DE) based clustering algorithm for WSNs called DECA. The main objective of the DECA is to prolong the network life time of the WSN by taking care of the energy consumption of the common sensor nodes and the gateways (i.e., CHs). By the network life time, we mean the time from the deployment of the WSN till the death of the first CH. The death of the first CH is delayed through balancing the energy consumption of the CHs which is implemented by the rate of energy consumption and residual energy. In order to find out fast and efficient solution, we introduce a local improvement phase in the traditional DE. This local improvement phase helps our DE based approach to converge faster than the traditional DE and the genetic algorithms (GAs) [9]. We perform extensive simulation of the proposed algorithm. The experimental results demonstrate the efficiency of the proposed algorithm in terms of network life, energy consumption and convergence rate. Our main contributions in this paper can be summarized as follows:

- A DE based clustering algorithm for WSNs to prolong network lifetime.
- An efficient vector encoding scheme for complete clustering solution.
- Mathematical derivation of the fitness function for DE based solution.
- Incorporation of a local improvement phase into the traditional DE for faster convergence and better performance of the algorithm.
- Simulation to demonstrate that the proposed algorithm is superior to existing protocols in terms of network life, number of dead sensor nodes, energy consumption of the network and convergence rate of the algorithm.

The rest of the paper is organized as follows. The related work is presented in Section 2. An overview of the differential evolution is given in Section 3. The system model is discussed in Section 4 which includes energy model, network model and used terminologies. The proposed algorithm and the experimental results are presented in Sections 5 and 6 respectively and we conclude our paper in Section 7.

2. Related works

A number of clustering algorithms [8,17,18] have been developed for WSNs. We present the review of such works based on heuristic and meta-heuristic approaches.

2.1. Heuristic approaches

LEACH [19] is a popular technique that forms clusters by using a distributed algorithm. It dynamically rotates the work load of the CH amongst the sensor nodes which is useful for load balancing. However, the main disadvantage of this approach is that a node with low energy may be selected as a CH which may die quickly. Moreover, the CHs send directly the packet to BS via single-hop communication which is impractical for WSNs with large coverage area. Therefore, a number of improved algorithms have been developed over LEACH such as PEGASIS [20], HEED [21], etc.

Compared to LEACH, PEGASIS promotes network lifetime. Rather than forming multiple clusters, PEGASIS forms chains of sensor nodes so that each sensor transmits and receives from a neighbor and only one node is selected from that chain as group head to convey data to the BS. However, it requires dynamic topology adjustment and the data delay is significantly high which

Please cite this article in press as: P. Kuila, P.K. Jana, A novel differential evolution based clustering algorithm for wireless sensor networks, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.08.064

is unsuitable for large-size networks. In [22], we have proposed a distributed cost-based energy balanced clustering and routing algorithm called CEBCRA where CHs are selected and clusters are formed in distributed way depending on some cost value of the sensor nodes. In the routing phase of this algorithm, the best neighbor relay node is selected by measuring the cost of each path toward the base station. However, it suffers from the connectivity problem of cluster heads.

To form clusters, Low et al. [10] have considered a *breadth-first search* (BFS) tree of the sensor nodes to find out the least loaded gateway for assigning a sensor node to a CH. The algorithm has the time complexity of $O(mn^2)$ for n sensor nodes and m CHs. For a large scale WSN, it seems that execution time of this algorithm is very high. Their algorithm also takes substantial amount of memory space for building a BFS tree for individual sensor node. In [11], we have proposed a load balanced clustering algorithm that runs in $O(n \log n)$ which is an improvement over [10]. Gupta and Younis [12] have proposed a clustering algorithm called LBC, where the authors define cardinality of a cluster as the number of sensor nodes associated with the cluster and attempts to minimize the variance of the cardinality of each cluster in the system. LBC takes $O(mn \log n)$ time in worst case. In [13], an energy efficient load-balanced clustering algorithm (EELBCA) have been proposed with $O(n \log m)$ time. EELBCA addresses energy efficiency as well as load balancing. EELBCA is a min-heap based clustering algorithm. A min-heap is build using CHs on the number of sensor nodes allotted to the CHs. However, the above algorithms [10–13] do not consider residual energy of the sensor nodes as well as CHs.

2.2. Meta-heuristic approaches

A number of meta-heuristic approaches have been reported for wireless sensor networks. Bari et al. [14] have proposed a GA based algorithm for data routing using relay nodes in a two-tire wireless sensor network. Selection of individuals is carried out using the Roulette-wheel selection method and the fitness function is defined by network lifetime in terms of rounds. For mutation operation, they select a critical node from the relay nodes, which dissipates the maximum energy due to receiving and/or transmitting data. Mutation is done by either replacing the next-hop node of this critical node by a new next-hop relay node or by diverting some incoming flow toward that critical node to other relay node. In [23], we have also proposed GA based routing algorithm called GAR where the overall communication distance from the gateways to the BS is minimized. However, it is different from [14] in respect of the following issues. For selection of individuals, tournament selection is used in contrast to Roulette-wheel selection. Fitness function is defined in terms of total distance covered in a round rather than network life time in terms of number of rounds. In the mutation phase, we select relay node that uses maximum distance to transmit the data to its neighbor in contrast to a critical node defined in [14]. However, both the algorithms as presented in [14] and [23] consider only routing of aggregated data from the gateways to the BS without considering data communication from the sensor nodes to the gateways within each cluster. Moreover, they have not considered residual energy of the nodes which may lead to serious energy inefficiency. In [24], Chakraborty et al. have presented a differential evolution based routing algorithm for more than a thousand relay nodes such that the energy consumption of the maximum energy-consuming relay node is minimized. However, the authors do not take care about the cluster formation. Some improper clustering may lead to serious energy inefficiency of the relay nodes.

Recently, we have proposed a GA based load balanced clustering algorithm for WSNs [9]. The algorithm forms clusters in such way that the maximum load of each gateway is minimized and it works for both of the equal and unequal load of the sensor nodes. The algorithm has faster convergence and better load balancing than the traditional GA. However, it does not take care of residual energy of the nodes. EAUCF (energy-aware unequal clustering with fuzzy) [25] is a distributed competitive unequal clustering algorithm. It makes local decisions for determining competition radius and electing CHs. In order to estimate the competition radius for tentative CHs, EAUCF employs both residual energy and distance to the base station parameters. However, selected CHs and their competition radius cannot assure complete coverage of the network. Enan and Bara [26] have presented an evolutionary aware routing protocol (EAERP) for dynamic clustering of wireless sensor networks. Here the authors have made an attempt to minimize the energy consumption throughout the network lifetime by choosing a set of CHs from the normal sensor nodes and all non-CH sensor nodes determine nearest CH to join. EAERP suffers same problem as LEACH, as some sensor node may became a CH which may not have sufficient energy. Moreover, EAERP requires re-clustering in each round to rotate the extra workload of CH. Unfortunately, being a centralized approach; EAERP requires whole network information in each round for re-clustering.

Hussain et al. [27] have presented a GA based hierarchical clustering algorithm to choose a set of cluster-heads from the normal sensor nodes as in [26] with different objective function. Huruiala et al. [28] have presented a GA based clustering and routing algorithm by choosing the optimal cluster-head and minimizing the transmission distance. Jin et al. [29] have presented a method to find the CHs based on a fitness function which minimizes the total transmission distance in network. Here, GA is used to select only cluster-heads. Each non-CH node uses a deterministic method to find its nearest cluster-head. Mudundi and Ali [30] have proposed a genetic clustering algorithm (GCA) for dynamic formation of clusters. Here, the goal is to increase lifetime of the network by minimizing the energy dissipation. The fitness function is developed by using the number of CH nodes and the Euclidian distance between all the nodes in each cluster to their CH with some weight value. However, their approach also forms the clusters by CH selection and joining of all the non-CH sensor nodes nearest to the CHs without any consideration of energy balancing of the CHs. Moreover, the method considers only the Euclidian distance between the cluster heads or BS and they did not attend any other parameters like residual energy, load balancing.

Particle swarm optimization (PSO) and ant colony optimization (ACO) are also used in WSNs and it can be found in [31–33]. Other evolutionary based approaches applied for WSNs can be found in [34–37] and their references inside them.

However, all the above evolutionary based clustering algorithms except [9] form the clusters by simply CH selection and allowing the non-CHs to join their nearest CHs. Moreover, they assume that sensor nodes are equally distributed. Therefore, if the non-CH sensor nodes join the nearest CH like LEACH then the CHs of densely deployed areas will be overloaded with higher number of member sensor nodes. Our DE based clustering algorithm presented in this paper is different from that of [9] with the following respects:

(1) Main objective of the proposed algorithm is to prolong the network life by preventing initial death of the gateways. Whereas in [9], we had proposed a novel GA based clustering algorithm which forms clusters in such way that the maximum load of each gateway is minimized.
(2) In cluster formation phase, we consider residual energy of the gateway along with energy consumption due to data processing and communication. Here, we also minimize the average cluster distances to reduce energy consumption of the sensor nodes. These two facts were not considered in [9].

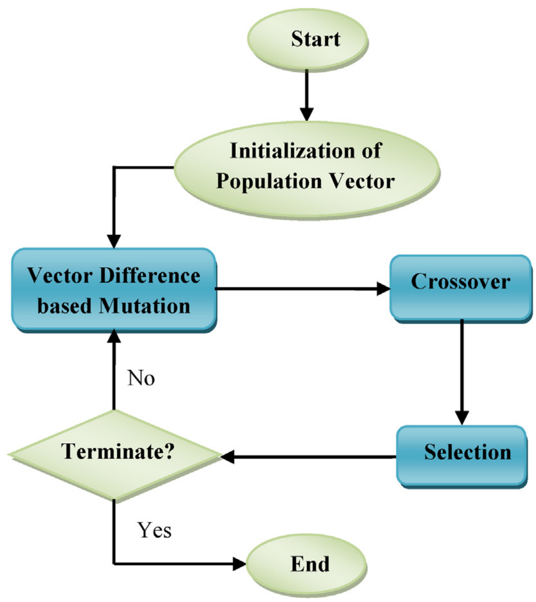

Fig. 2. Flowchart of differential evolution.

(3) The process of the vector initialization is fully different than the chromosome representation process in [9].

3. Overview of differential evolution

DE is a stochastic and population based evolutionary algorithm that is widely applied in solving many optimization problems. It consists of four stages, i.e., initialization of population vector, mutation, crossover and selection. DE begins with randomly generated real valued population vectors of some predefined population size (say P). The vectors are also known as genome or chromosome and each individual vector gives a complete solution to the multidimensional optimization problem. The dimension D of all the vectors is equal. Each individual vector is evaluated by a fitness function to judge the quality of the solution to the problem.

Once the population vectors are encoded, the algorithm iterates up to G (say) generation with the mutation, crossover and selection operation to enhance the quality of the population vectors. Therefore, the population vectors are likely to be changed over different generations. Finally, depending on the fitness function, the best vector is selected as the final solution vector. The various steps of a traditional DE are depicted in the flowchart as shown in Fig. 2. However, in the proposed work, we include one additional step called local improvement as discussed in Section 5, the justification of which is given in Section 5.5.

This can be noted that DE has similar computational steps as the traditional evolutionary algorithms (EAs) such as GAs (genetic algorithms). However, unlike traditional EAs, the DE-variants perturb the current generation population members with the scaled differences of randomly selected and distinct population members. Therefore, no separate probability distribution has to be used for generating the offspring [15].

Various schemes [15,16] have been proposed for DE. In order to distinguish the schemes, the notation "$DE/x/y/z$" is used, where DE denotes the differential evolution, x specifies the vector to be mutated (which can be random or the best vector from the population), y represents the number of difference vectors which are used in the mutation operation and z denotes the crossover scheme (which can be binomial or exponential). Some popular existing DE schemes are $DE/best/1/exp$, $DE/rand/1/bin$, $DE/best/2/exp$, $DE/rand/2/exp$, etc. In the mutation phase of the DE, a donor vector is created for each member vector of the population called target vector. Then the crossover operation takes place between target vector and the donor vector. The donor vector exchanges its components with the target vector to create offspring which is called trial vector. The selection process determines which vector, the target or the trial vector from the current generation (G) survives in the next generation ($G+1$). The selection process keeps the population size constant over subsequence generation. Both of the target and the newly improved trial vectors are evaluated by the derived fitness function. The trial vector replaces the target vector if the trial vector has better fitness value; otherwise the target vector retained in the population.

4. System model

4.1. Energy model

The radio model for energy used in this paper is same as discussed in [19]. In this model, both the free space and multi-path fading channels are used depending on the distance between the transmitter and receiver. When the distance is less than a threshold value d_0, then the free space (fs) model is used, otherwise, the multipath (mp) model is used. Let E_{elec}, ε_{fs} and ε_{mp} be the energy required by the electronics circuit and by the amplifier in free space and multipath respectively. Then the energy required by the radio to transmit an l-bit message over a distance d is given as follows:

$$E_T(l,d) = \begin{cases} lE_{\text{elec}} + l\varepsilon_{fs}d^2 & \text{for} \quad d < d_0 \\ lE_{\text{elec}} + l\varepsilon_{mp}d^4 & \text{for} \quad d \geq d_0 \end{cases} \tag{4.1}$$

The energy required by the radio to receive an l-bit message is given by

$$E_R(l) = lE_{\text{elec}} \tag{4.2}$$

The E_{elec} depends on several factors such as digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\varepsilon_{fs}d^2/\varepsilon_{mp}d^4$, depends on the distance between the transmitter and the receiver and also on the acceptable bit-error rate. It should be noted that this is a simplified model. In general, radio wave propagation is highly variable and difficult to model.

4.2. Network model

We assume a WSN model where all the sensor nodes are randomly deployed along with a few gateways and once they are deployed, they become stationary. As mentioned in Section 1, the gateways are different from the normal sensor nodes with respect to communication and functional capabilities. A sensor node can be assigned to any gateway if it is within the communication range of the sensor node. Therefore, there are some pre-specified gateways onto which a particular sensor node can be assigned. Thus each sensor node has a list of gateways and it can be assigned to only one gateway amongst them. Similar to LEACH, the data gathering operation is divided into rounds. In each round, all sensor nodes sense local data and send it to their CH. Then CHs perform data aggregation to discard the redundant and uncorrelated data and send the aggregated data to the base station. Between two adjacent rounds, all nodes turn off their radios to save energy. All communication is over wireless link. A wireless link is established between two nodes only if they are within the communication range of each other. Current implementation supports TDMA [38] protocol to provide MAC layer communication. Gateways use slotted carrier-sense multiple access (CSMA) MAC [39] protocol to communicate with base station.

Various definition of the network life is given in the literature [40,41], such as this is the time until first node dies, the time until last node dies or the time until a desired percentage of nodes die. Moreover in some scenario [42], network life is considered as the period until the entire region is covered. In [43], the authors have defined the network life in several ways, e.g., N-of-N lifetime, K-of-N lifetime and m-in-K-of-N lifetime where, N-of-N lifetime defines the duration until first gateway dies, K-of-N lifetime presents the duration up to when K gateways out of N are alive and m-in-K-of-N lifetime presents the duration until all m supporting gateways and overall a minimum of K gateways are alive. In this paper, we have used lifetime of the network in terms of round from the start of the network operation until one gateway depletes its complete energy and stop functioning.

4.3. Terminologies

We use the following notations for the problem formulation as follows:

- The set of sensor nodes is denoted by $S = \{s_1, s_2, \ldots, s_n\}$.
- The set of gateways is denoted by $\zeta = \{g_1, g_2, \ldots, g_m\}$ where, $n > m$.
- $E_{\text{residual}}(g_i)$ denotes the remaining energy of g_i.
- $E_{\text{Gateway}}(g_i)$ denotes the total energy consumption of g_i in a single round. The gateways consume their energy to receive the sensed data by their member sensor nodes, data aggregation and finally to send the aggregated data to the base station. If g_i has n_i number of member sensor nodes, then the total energy consumption of g_i in a single round can be calculated as follows:

$$E_{\text{Gateway}}(g_i) = n_i \times E_R + n_i \times E_{\text{DA}} + E_T(g_i, BS) \tag{4.3}$$

  where E_R, E_{DA} and E_T are the energy consumption due to receiving, data aggregation and transmission to BS respectively.
- Let $L(i)$ denote the lifetime of the gateway g_i. If g_i has residual energy $E_{\text{residual}}(g_i)$ and consumes energy $E_{\text{Gateway}}(g_i)$ per round then $L(i)$ can be calculated as follows:

$$L(i) = \left\lfloor \frac{E_{\text{residual}}(g_i)}{E_{\text{Gateway}}(g_i)} \right\rfloor \tag{4.4}$$

- $dist(s_i, g_j)$ denotes the Euclidian distance between sensor node s_i and gateway g_j.
- $ComCH(s_i)$ is the set of all those gateways, which are within the communication range (R_S) of sensor node s_i. In other words,

$$ComCH(s_i) = \{g_j \mid dist(s_i, g_j) \le R_S \wedge g_j \in \zeta\} \tag{4.5}$$

  Therefore, s_i can be assigned to any one of the gateway from $ComCH(s_i)$, where $ComCH(s_i) \subseteq \zeta$.
- $AvegClusDist(g_j)$ denotes the average distances from all the member sensor nodes to g_j. Therefore, this is calculated after a complete assignment as follows:

$$AvegClusDist(g_j) = \frac{1}{n_j}\sum_{i=1}^{n}\{dist(s_i, g_j) \times \alpha_{i,j}\} \tag{4.6}$$

  where $\alpha_{i,j}$ is a Boolean variable such that

$$\alpha_{ij} = \begin{cases} 1, & \text{if sensor node } s_i \text{ is assigned to cluster head} \\ & g_j, \forall i, j : 1 \le i \le n, \quad 1 \le j \le m. \\ 0, & \text{otherwise.} \end{cases}$$

For efficient clustering, our objective is to maximize the network lifetime. This can be achieved by maximizing the lifetime of the gateways but reducing the energy consumption of the sensor nodes. In other words, Minimize $Z = \frac{W}{L}$ where $L = \min\{L(i), \forall g_i \in \zeta\}$ and $W = \max\{AvegClusDist(g_i), \quad \forall g_i \in \zeta\}$.

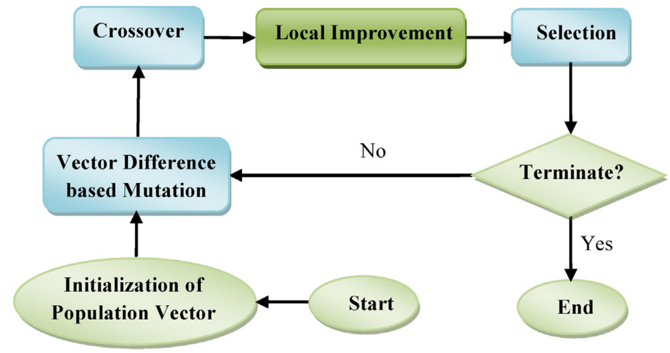

Fig. 3. Flowchart of proposed DE based clustering algorithm.

5. Proposed clustering algorithm

Network setup is performed in two phases: bootstrapping and clustering. During the bootstrapping process, all the sensor nodes and gateways are assigned unique IDs. Then the sensor nodes broadcast their IDs using CSMA/CA MAC [39] layer protocol. Therefore, the gateways within the communication range of these sensor nodes can collect the sensor IDs and finally send the local network information to the base station. Thus for each sensor node, the number of gateways within its communication range can be calculated by the base station. In clustering phase base station executes the clustering algorithm. When the clustering is over, all the gateways provide their IDs to their member sensor nodes by single hop communication. Then the gateways provide a TDMA schedule to their member sensor nodes for intra cluster communication. The flowchart of our proposed DE based clustering algorithm, DECA is depicted in Fig. 3. This is important to note that our approach has one additional stage of computation, i.e., local improvement. The justification of this extra stage is described in Section 5.5.

We now present our methodologies for population vector initialization and determination of fitness function followed by mutation, crossover, local improvement and selection operation in the subsections as follows.

5.1. Initialization of the population vector

Our proposed scheme for vector representation is as follows. The vectors are represented in such a way that each vector indicates the complete assignment of all the sensor nodes to the gateways. We adopt the following notation for representing the ith vector of the population at the Gth generation as

$$\vec{X}_{i,G} = [x_{1,i,G}, x_{2,i,G}, x_{3,i,G}, \ldots, x_{N,i,G}] \tag{5.1}$$

where N is the dimension of the vector and value of the jth component, i.e., $x_{j,i,G}$ maps the assignment of the sensor node s_j to a gateway. We initialize each component with a randomly generated uniformly distributed number $Rand(0, 1]$, $0 < Rand(0, 1] \le 1$. The random number is generated independently for each component. Therefore, the jth component of this vector, i.e., $x_{j,i,G} = Rand(0, 1]$, $1 \le j \le N$ maps a gateway (say g_k) to which the sensor node s_j is assigned. The mapping is done as follows:

$$g_k = Index(ComCH(s_j), H) \tag{5.2}$$

where $Index(ComCH(s_j), H)$ is an indexing function that indexes the Hth gateway from $ComCH(s_j)$ and $H = \text{Ceiling}(x_{j,i,G} \times |ComCH(s_j)|)$.

This is important to note that the above vector representation is a part of the clustering algorithm. As mentioned above that the dimension of each vector is equal to the number of the sensor nodes, therefore, addition/deletion of any sensor nodes would change the vector dimension and it would require re-clustering.

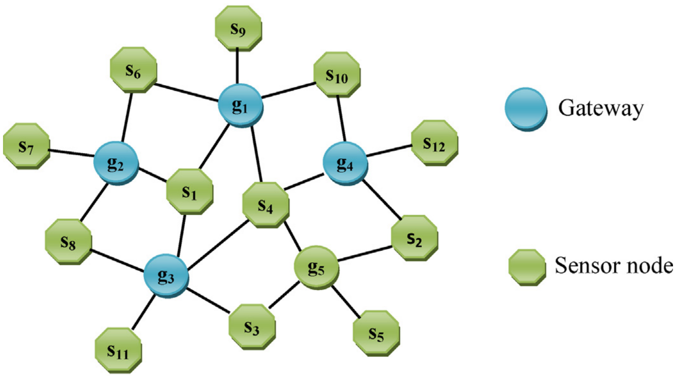

Fig. 4. A wireless sensor network with gateways.

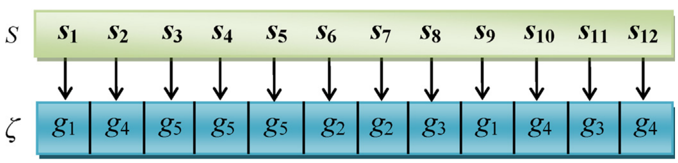

Fig. 5. Final assignment.

Illustration 6.1. Consider a WSN with 12 sensor nodes and 5 gateways, i.e., $S=\{s_1, s_2, \ldots, s_{12}\}$ and $\zeta=\{g_1, g_2, g_3, g_4, g_5\}$ as shown in Fig. 4. Therefore, the dimension of the vector is same as the number of sensor nodes, i.e., $D=12$.

The edges between the sensor nodes and the gateways indicate that the gateways are within communication range of the sensor nodes. It can be observed from Fig. 4 that the sensor node s_1 is connected with three gateways. In other words, $ComCH(s_1)=\{g_3, g_1, g_2\}$. The second column of Table 1 shows the gateways which are within communication range of the sensor nodes.

Now, for each element of the ith vector at the Gth generation, a random number is generated to initialize the element. Let, the generated random number for the first element is 0.46, i.e., $x_{1,i,G}=0.46$ as shown in fourth column of Table 1. Hence, Celling($x_{1,i,G} \times |ComCH(s_1)|$) = 2, therefore the 2nd gateway from $ComCH(s_1)$, i.e., g_1 is selected for assigning s_1 as shown in Table 1. In the same way all the sensor nodes are assigned to a gateway using the randomly generated vector. The final assignment of the sensor nodes to their corresponding gateways is shown in Table 1.

Therefore, the vector $\vec{X}_{i,G}=[0.46, 0.19, 0.39, 0.67, 0.86, 0.63, 0.24, 0.41, 0.34, 0.73, 0.63, 0.92]$ maps the assignment of the sensor nodes to their gateways shown in Fig. 5.

This is important to note that the different order of gateways under *ComCH* may produce another assignment which is also valid.

Table 1
Sensor nodes assignment from vector encoding.

Sensor nodes	$ComCH(s_j)$	$\lvert ComCH(s_j)\rvert$	$x_{j,i,G}$	Ceiling ($x_{j,i,G} \times \lvert ComCH(s_j)\rvert$)	Assigned gateway
s_1	$\{g_3, g_1, g_2\}$	3	0.46	2	g_1
s_2	$\{g_4, g_5\}$	2	0.19	1	g_4
s_3	$\{g_5, g_3\}$	2	0.39	1	g_5
s_4	$\{g_1, g_4, g_5, g_3\}$	4	0.67	3	g_5
s_5	$\{g_5\}$	1	0.86	1	g_5
s_6	$\{g_1, g_2\}$	2	0.63	2	g_2
s_7	$\{g_2\}$	1	0.24	1	g_2
s_8	$\{g_3, g_2\}$	2	0.41	1	g_3
s_9	$\{g_1\}$	1	0.34	1	g_1
s_{10}	$\{g_1, g_4\}$	2	0.73	2	g_4
s_{11}	$\{g_3\}$	1	0.63	1	g_3
s_{12}	$\{g_4\}$	1	0.92	1	g_4

5.2. Fitness function

Here our main objective is to maximize the network life. This can be possible only if we can make a balance of the life time of the gateways. The general principle behind the balancing of gateway life is as follows:

Principle: The gateway with lower residual energy should have lower rate of energy consumption per round than the gateways with higher residual energy.

Thus the life time of all the gateways can be balanced effectively. Now, we build the fitness function to evaluate the individual vector depending on the following parameters described as follows.

5.2.1. Standard deviation of lifetime of the CHs

The CHs consume their energy owing to extra work load from their member sensor nodes and life time of the CHs is very crucial for the long run operation of WSNs. Our main objective is to maximize the minimum life time of the CHs. This can be possible if we can make a balance of the lifetime of the CHs. Balancing the lifetime of the CHs mean that life time of all the CHs is more or less same. This can be possible by applying the above principle. The estimated energy consumption of the gateways and their lifetime can be calculated by the Eqs. (4.3) and (4.4) respectively. In order to judge the quality of the balanced lifetime of the CHs, we measure the standard deviation of the life. The standard deviation (σ_L) can help us to measure the quality of evenly distribution of the gateways life. It can be calculated as follows:

$$\sigma_L = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\{\mu_L - L(j)\}^2} \tag{5.3}$$

where $\mu_L = \frac{1}{m}\sum_{i=1}^{m} L(i)$ and m is the number of CHs. The shorter the σ_L, the higher is the fitness value. Therefore, the fitness function is reversely proportional to the σ_L. In other words,

$$Fitness \propto \frac{1}{\sigma_L} \tag{5.4}$$

5.2.2. Standard deviation of average cluster distance

Sensor nodes consume its maximum energy to transmit the sense data to its corresponding CH. In order to minimize the energy consumption of the sensor nodes, they should be assigned to their nearest CH. Here, our objective is to minimize the cluster distance of each cluster. Therefore, we have measured the standard deviation (σ_D) of the cluster distance and it can be calculated as

$$\sigma_D = \sqrt{\frac{1}{m}\sum_{j=1}^{m}\{\mu_D - AvegClusDist(g_j)\}^2} \tag{5.5}$$

where $\mu_D = \frac{1}{m}\sum_{i=1}^{m} AvegClusDist(g_i)$ and m is the number of CHs. The shorter the σ_D, the higher is the fitness value. Therefore, the fitness function is reversely proportional to the σ_D. In other words,

$$Fitness \propto \frac{1}{\sigma_D} \tag{5.6}$$

Eqs. (5.4) and (5.6) combinedly implies that

$$\text{Fitness} = K \times \frac{1}{\sigma_L} \times \frac{1}{\sigma_D} \tag{5.7}$$

where K is proportionality constant. It is noteworthy that the fitness value is used only comparison purpose. Therefore, the value of K does not hamper our objective. Without loss of generality, we assume that $K = 1$. Therefore,

$$\begin{aligned}
&\text{Fitness} = \frac{1}{\sigma_L} \times \frac{1}{\sigma_D} \\
&i.e., \quad \text{Fitness} = \frac{1}{\sqrt{\frac{1}{m}\sum_{j=1}^{m}\{\mu_L - L(j)\}^2}} \times \frac{1}{\sqrt{\frac{1}{m}\sum_{j=1}^{m}\{\mu_D - AvegClusDist(g_j)\}^2}} \\
&i.e., \quad \text{Fitness} = \frac{m}{\sqrt{\sum_{j=1}^{m}\{\mu_L - L(j)\}^2 \times \sum_{j=1}^{m}\{\mu_D - AvegClusDist(g_j)\}^2}}
\end{aligned} \tag{5.8}$$

Vectors are evaluated by the above fitness function. Higher is the fitness value, better is the chromosome.

Remark 5.1. The above fitness function takes care of energy consumption of both the gateways and the sensor nodes. The first parameter (σ_L) of the fitness function (Eq. (5.8)), takes care of the energy consumption of the gateways as follows. The lower the value of σ_L better is the life time of the gateways (i.e., better is the life time of the network) and this is achieved following the above *principle*. The second parameter σ_D of the fitness function takes care of the energy consumption of the sensor nodes as follows. The lower value of σ_D which minimizes the maximum cluster distance helps to enrich the fitness value. It should be noted that the required transmitting energy of the sensor nodes exponentially proportional to the distances between sensor nodes and the gateways. Thereby, the second parameter reduces the overall energy consumption of the sensor nodes.

5.3. Mutation

We use the scheme $DE/best/1/bin$ for mutation and crossover operation. For each member vector of the population (called target vector), a donor vector is created through the differential mutation operation by employing the best vector from the population and two other distinct vectors which are generated randomly. Let $\vec{X}_{i,G}$, $\vec{X}_{\text{best},G}$, and $\vec{V}_{i,G}$ be the target vector, best vector and donor vector respectively. Then the mutation process can be expressed as

$$\vec{V}_{i,G} = \vec{X}_{\text{best},G} + F \cdot \vec{D}_{i,G} \tag{5.9}$$

where $r, s \in [1, P]$ with $i \neq r \neq s \neq best$, F is the scaling factor which typically lies in the interval $[0.4, 1]$ and $\vec{D}_{i,G} = (\vec{X}_{r,G} - \vec{X}_{s,G})$ is the difference vector of this mutation operation.

It is noteworthy that the above mutation operation which is used for standard DE may not work for our scenario unfortunately. This is because the algebraic subtraction operation of two components of the vectors may cause the resultant components of the difference vector negative. It should be noted that, in our scenario each component of the vectors must be within the range $(0, 1]$. Therefore, our algorithm should generate the components of the difference vector in such a way that it can satisfy the range. As $0 < x_{j,r,G} \le 1$ and $0 < x_{j,s,G} \le 1$, this can be possible only if we choose the component of difference vector as follows:

$$d_{j,i,G} = \begin{cases} 1 + (x_{j,r,G} - x_{j,s,G}) & \text{if } ((x_{j,r,G} - x_{j,s,G}) \le 0) \\ x_{j,r,G} - x_{j,s,G} & \text{Otherwise.} \end{cases} \tag{5.10}$$

The same type of problem may be arisen at the time of arithmetic addition of the mutation process (Eq. (5.9)). Again, as $0 < x_{j,\text{best},G} \le 1$ and $0 < x_{j,s,G} \le 1$, the components of the donor vector are generated as follows:

$$v_{j,i,G} = \begin{cases} (x_{j,\text{best},G} + F \times d_{j,s,G}) - 1 & \text{if } ((x_{j,\text{best},G} + F \times d_{j,s,G}) > 1) \\ x_{j,\text{best},G} + F \times d_{j,s,G} & \text{Otherwise.} \end{cases} \tag{5.11}$$

5.4. Crossover

After the mutation operation, crossover operation is performed in between donor vector and target vector to form a new child vector known as trial vector. Here, we use binomial cross over operation with a predefined crossover rate (say Cr). Let $\vec{U}_{i,G} = [u_{1,i,G}, u_{2,i,G}, u_{3,i,G}, \ldots, u_{N,i,G}]$ be the trial vector. Then the crossover operation can be outlined as follows:

$$u_{j,i,G} = \begin{cases} v_{j,i,G} & \text{if } (Rand[0, 1] \le Cr) \\ x_{j,i,G} & \text{Otherwise.} \end{cases} \tag{5.12}$$

To generate each of the components of the trial vector, a number is picked up randomly in between 0 and 1. For the jth component of the trial vector, if the randomly picked up number is less than or equal to the Cr, then jth component of the trial vector is taken same as the jth component donor vector; otherwise, it is taken same as jth component of target vector. The whole crossover process is shown in Fig. 6.

5.5. Local improvement

The local improvement step is added to improve the quality of the newly generated trial vector as follows. To perform the local improvement process, we select the gateway with minimum lifetime from the trial vector. The purpose of selecting this gateway is to enhance its lifetime and thus the network life can be prolonged by delaying the first death of the gateway. This can be possible if we can reduce the load of the gateway by reassigning a member sensor node from its cluster to any other cluster with higher lifetime

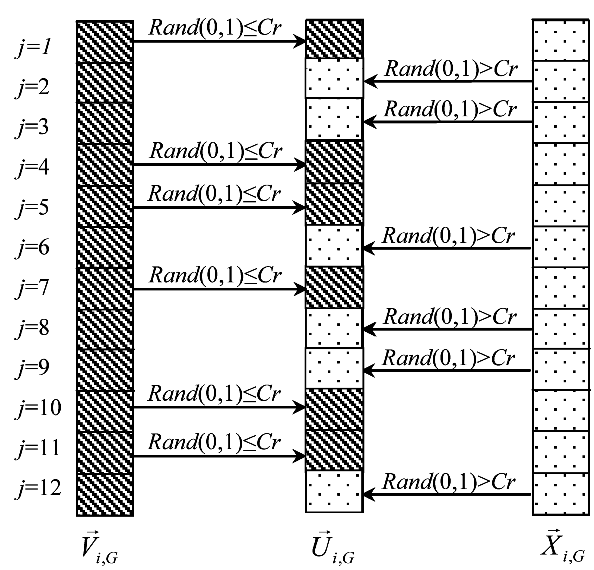

Fig. 6. Crossover operation.

gateway. To achieve this, a sensor node is randomly selected from the cluster of minimum life gateway and is reassigned to another gateway which is closer to it than the remaining gateways. This is illustrated as follows:

Illustration 5.2. Consider the WSN as shown in Fig. 7(a). Assume that this is the configuration of the WSN before the local improvement phase. Also assume that the life time of the gateways g_1, g_2 and g_3 are already calculated and they are $L(1) = 421$, $L(2) = 523$ and $L(3) = 641$. Therefore, the minimum lifetime gateway is g_1 with 421 rounds. To enhance the lifetime of g_1, a cluster member of g_1 is randomly picked up to reassign another gateway. Let, s_1 be the randomly picked up member sensor node of g_1. Now, s_1 is reassigned to any one of g_2 or g_3 whichever is nearest. Let g_3 be the nearest gateway of s_1. Therefore, s_1 is reassigned to g_3 and the current lifetime of the gateway g_1 is increased but lifetime of g_3 is decreased. Fig. 7(b) shows this situation of the network after the local improvement phase.

Remark 5.2. The local improvement phase enhances the performance of the algorithm as follows. We first select the gateway with lowest lifetime and then randomly pick a sensor node from the cluster of this gateway to reassign the sensor node into another gateway with higher life. Thus lifetime of the lowest life gateway is improved. In other words, first parameter (σ_L) of the fitness function (Eq. (5.8)) is improved. During sensor reassignment, the randomly selected sensor node is reassigned to the nearest gateway. Therefore, the energy consumption of the reassigned sensor node is also taken care to be minimum. Thus second parameter (σ_D) of the fitness function is also improved. Hence, the local improvement phase helps the trial vector to enhance its fitness and helps the proposed algorithm to converge faster and perform better than the traditional DE and GA.

5.6. Selection

The selection operation decides which vector from the target vector and newly generated trial vector will survive to the next generation. Both of the vectors are evaluated by the derived fitness function. The trial vector replaces the target vector if the trial vector has better fitness value; otherwise the target vector retained in the population. The selection process is described as

$$\vec{X}_{i,G+1} = \begin{cases} \vec{U}_{i,G} & \text{if Fitness}(\vec{U}_{i,G}) \geq \text{Fitness}(\vec{X}_{i,G}) \\ \vec{X}_{i,G} & \text{Otherwise.} \end{cases} \tag{5.13}$$

6. Experimental results

We performed extensive experiments on the proposed algorithm using MATLAB version R2012b and C programming language. The experiments were performed with diverse number of sensor nodes ranging from 100 to 500 and 15 to 50 gateways placed in a 200 m × 200 m area. Each sensor node was assumed to have initial energy of 2 J and each gateway, 10 J. In the simulation run, we used following parameter values same as in [9,19] as shown in Table 2.

We considered two scenarios (WSN#1 and WSN#2) for the sake of simulation. Both of them have the sensing field of 200 m × 200 m area. For the WSN#1, the position of the base station was taken at (200, 100), i.e., in a side of the region and for the WSN#2, the position of the base station was taken at (100, 100), i.e., in the center of the region. To execute our proposed algorithm DECA, we considered an initial population of 100 chromosomes. In our simulation, crossover rate (Cr) and scaling factor (F) was taken as 0.7 and 0.5 respectively [44]. For termination criteria, we kept a fixed number of iterations like other DE-based works such as [44]. For the sake of comparison, we also ran GLBCA [10], LBC [12], EELBCA [13],

Table 2
Parameters of simulation.

Parameter	Value
Area	200 m × 200 m
Sensor nodes	100–500
Gateways	15–50
Initial energy of sensor nodes	2.0 J
Number of simulation iterations	200
Communication range	150 m
E_{elec}	50 nJ/bit
ε_{fs}	10 pJ/bit/m^2
ε_{mp}	0.0013 pJ/bit/m^4
d_0	87.0 m
E_{DA}	5 nJ/bit
Packet size	4000 bits
Message size	200 bits

traditional DE and GA with same fitness function as used by our proposed algorithm. In the traditional DE, the vector encoding schema, fitness function, population size crossover rate and all other parameters were taken same as that of the proposed DE, except the local improvement phase which is introduced in our proposed DE based algorithm. In the experiment using GA, we considered an initial population of 100 chromosomes. In the chromosome generation phase, same encoding schema was used according to our proposed algorithm. For crossover operation, we selected the best 10% chromosomes using Roulette-wheel selection and in our simulation, crossover rate was taken as 0.7 and mutation rate as 0.05.

First, we ran the algorithms for comparing lifetime of the network by varying the sensor nodes from 100 to 400 and the number of gateways for 30 and 50 on both the network scenarios, WSN#1 and WSN#2. Figs. 8 and 9 show the comparison of the DECA, traditional DE, GA and LBC in terms of network life in WSN#1 and WSN#2 respectively. It can be noted from Figs. 8 and 9 that DECA has better network lifetime than all the algorithms, i.e., LBC, GLBCA, EELBCA, traditional DE and the GA based clustering. The rationale behind this is that the DECA takes care of the CHs with lower residual energy by assigning lesser number of sensor nodes. Therefore it prevents initial death of the CHs and increases network lifetime, whereas the other clustering algorithms do not deal with the residual energy of the CHs. They balance the load of the CHs in terms of cardinality of the clusters. As a result, the CHs with lower residual energy die faster than the others. As the traditional GA and DE use same fitness function as DECA, they perform comparably better than the LBC, GLBCA and EELBCA.

Next we ran the algorithms to compare the balancing of lifetime of the gateways by varying the sensor nodes from 200 to 500 for 50 gateways on both the network scenarios, WSN#1 and WSN#2. Here, we calculate the duration between first gateway die (FGD) and last gateway die (LGD) in rounds. This is to be noted that lower the duration, better is the balancing of the lifetime. Fig. 10(a) and (b) shows the comparison of the DECA, LBC, GLBCA, EELBCA, traditional DE and GA in terms of balancing of lifetime of the gateways in WSN#1 and WSN#2 respectively. Note that DECA has better balancing than the LBC, GLBCA, EELBCA, traditional DE and GA based clustering algorithm as can be observed from Fig. 10.

Fig. 11(a) and (b) shows the comparison of the algorithms in terms of number of dead sensor nodes per round in WSN#1 and WSN#2 respectively. The algorithms are simulated for 300 sensor nodes and 30 gateways. It is clear that, the rate of death of the sensor nodes for DECA, traditional DE and GA is lesser than LBC and GLBCA in both of the scenarios. This is due to the fact that our derived fitness function takes care about the energy consumption of the normal sensor nodes by reducing the distances between sensor nodes and the gateways. It should be noted that, LBC and GLBCA balances the load of the gateways in terms of number of sensor nodes or cardinality of the cluster without considering

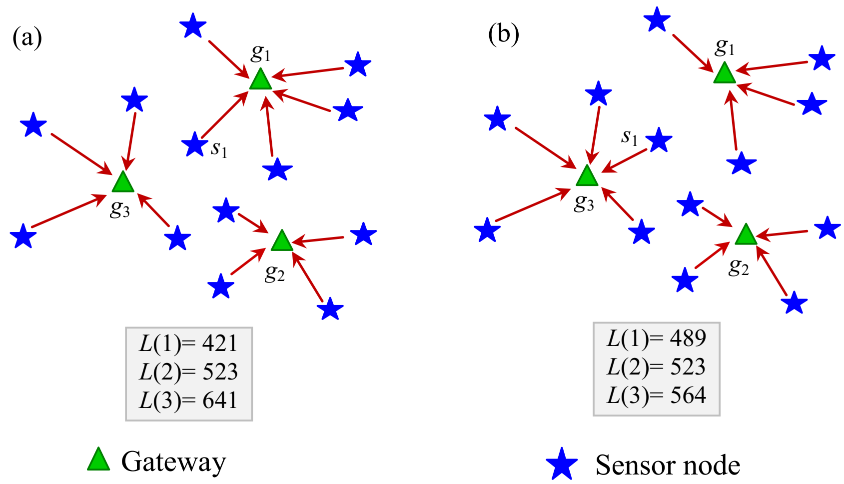

Fig. 7. A WSN of (a) before and (b) after local improvement.

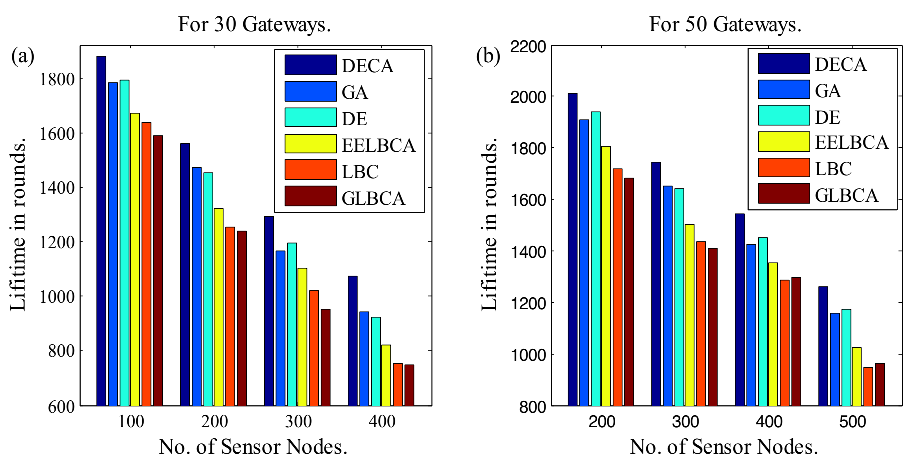

Fig. 8. Comparison in terms of lifetime for (a) 15 and (b) 30 gateways in WSN#1.

distance between sensor nodes and the CHs. To achieve this goal, some sensor nodes are assigned to a gateway which may be farther from it. As a result their energies are drained out due to long haul transmission and die quickly. However, EELBCA performs better than the other algorithms. This is because it assigns the sensor nodes to their CHs by considering the distance between the sensor nodes and their CHs. Although the sensor nodes consume lesser energy, the CHs die faster in EELBCA as residual energy of the CHs is not considered. Fig. 12(a) and (b) shows the comparison of energy (J) consumption of the network per round for 300 sensor nodes and 30 gateways in WSN#1 and WSN#2 respectively. Here, also EELBCA consumes lesser energy due to lesser energy consumption of sensor nodes. However, DECA outperforms the traditional DE, GA, LBC and GLBCA in this respect.

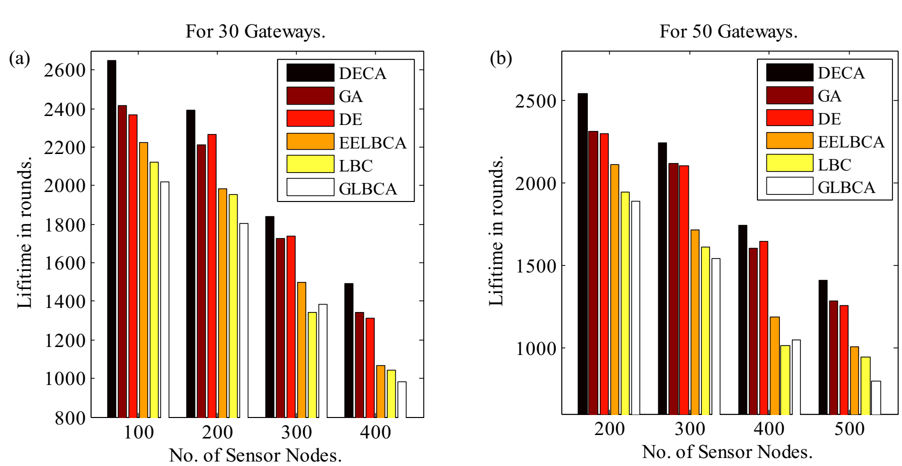

Fig. 9. Comparison in terms of lifetime for (a) 15 and (b) 30 gateways in WSN#2.

Please cite this article in press as: P. Kuila, P.K. Jana, A novel differential evolution based clustering algorithm for wireless sensor networks, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.08.064

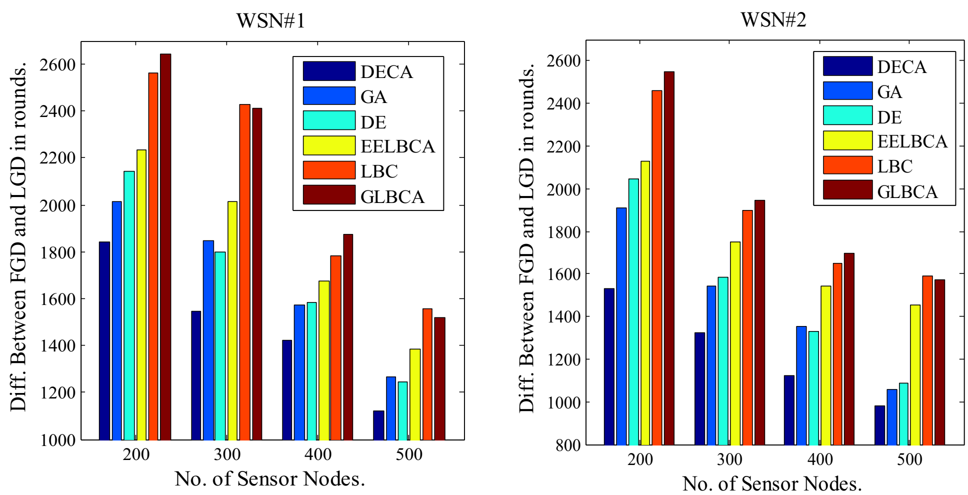

Fig. 10. Difference between FGD (first gateway die) and LGD (last gateway die) in rounds for (a) WSN#1 and (b) WSN#2.

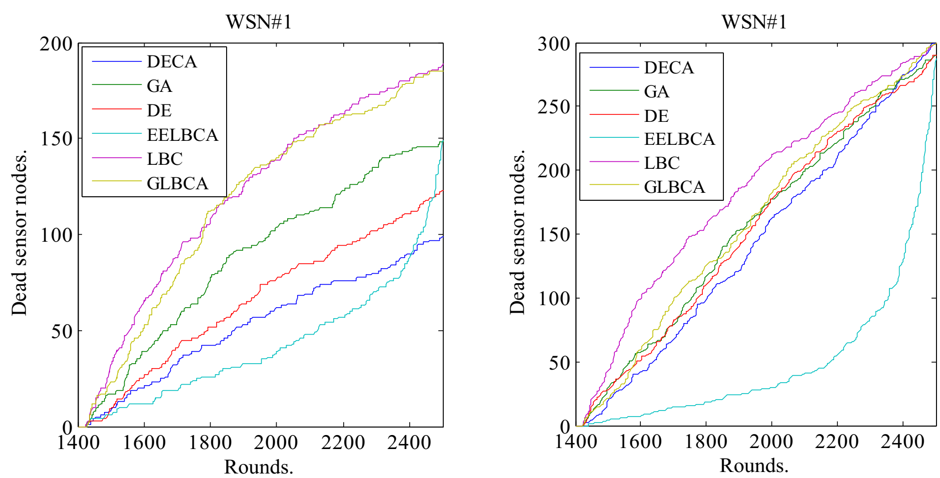

Fig. 11. Comparison in terms of dead sensor nodes for (a) WSN#1 and (b) WSN#2.

Note that, here our objective is not only to minimize the energy consumption of the network, but also maximize the life time. This is successfully achieved by taken care of the lifetime of the CHs which is crucial for extending network life. In the case of EELBCA, although sensor nodes are alive due to comparably less energy consumption, they may become uncovered due to faster death of the CHs.

Fig. 13(a) and (b) shows the comparison of the convergence rate of DECA, traditional DE and the GA in WSN#1 and WSN#2

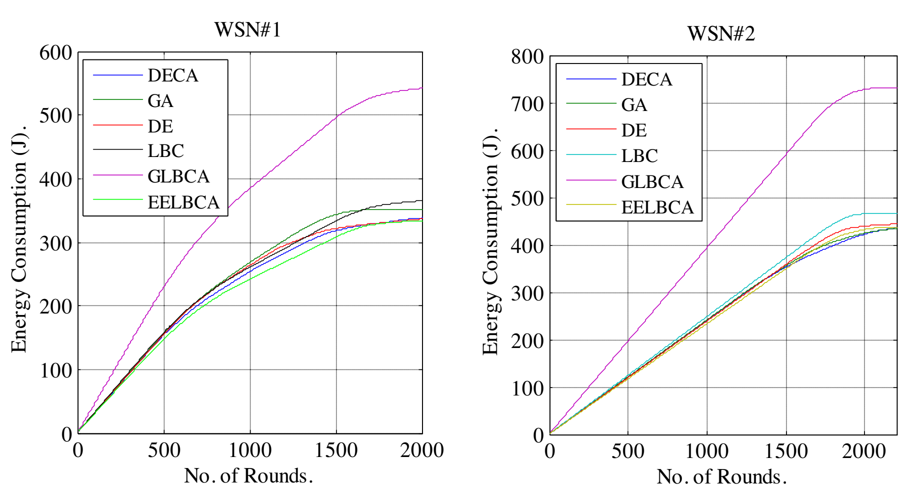

Fig. 12. Comparison in terms of energy consumption for (a) WSN#1 and (b) WSN#2.

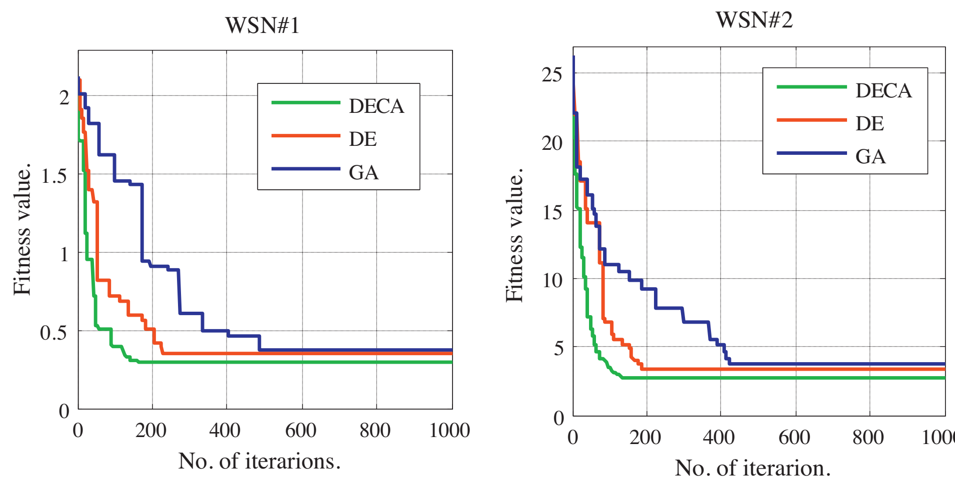

Fig. 13. Comparison in terms of convergence for (a) WSN#1 and (b) WSN#2.

respectively. We ran the algorithms for 500 sensor nodes and 50 gateways. Fig. 13 clearly shows the faster convergence of our proposed algorithm due to the local improvement phase, which is incorporated with the traditional DE for faster convergence.

7. Conclusions

In this paper, we have presented a DE based clustering algorithm for wireless sensor networks. The proposed approach has introduced an efficient vector encoding scheme and an extra phase called local improvement to improve the performance of the proposed clustering algorithm. We have also derived an efficient fitness function for extending network life time significantly. The fitness function takes care of energy consumption of both the gateways and the sensor nodes. The experimental results have shown that the proposed algorithm converges faster than the traditional DE and GA. We have also shown that it performs better than the exiting algorithms, i.e., the traditional DE and GA, LBC and GLBCA in terms of network life, energy consumption and number of dead sensor nodes. However, it performs inferior to the other existing algorithm namely EELBCA in terms of energy consumption and number of dead sensor nodes. In the proposed scheme, we have assumed that the gateways can directly communicate with the base station which may not be realistic for a large area network. Our future research will be in the direction of designing DE based energy aware clustering with multi-hop routing between the CHs and the base station.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38 (4) (2002) 393–422.
[2] M. Sousa, W. Lopes, F. Madeiro, M. Alencar, Cognitive LF-Ant: a novel protocol for healthcare wireless sensor networks, Sensors 12 (8) (2012) 10463–10486.
[3] G. Anastasi, M. Conti, M. Di Francesco, A. Passarella, Energy conservation in wireless sensor networks: a survey, Ad Hoc Netw. 7 (3) (2009) 537–568.
[4] E. Lattanzi, E. Regini, A. Acquaviva, A. Bogliolo, Energetic sustainability of routing algorithms for energy-harvesting wireless sensor networks, Comput. Commun. 30 (14) (2007) 2976–2986.
[5] H. Benton, C. Calhoun, D. Denis, Naveen V., F. Daniel, D.D. Finchelstein, S. Wentzloff, A.W. Cho, A.P. Chandrakasan, Design considerations for ultra-low energy wireless microsensor nodes, IEEE Trans. Comput. 54 (6) (2005) 727–740.
[6] M.A. Yigitel, O.D. Incel, I. Ersoy, QoS-aware MAC protocols for wireless sensor networks: a survey, Comput. Netw. 55 (8) (2011) 1982–2004.
[7] A.A. Alhameed Alkhatib, G.S. Baicher, MAC layer overview for wireless sensor networks, in: International Conference on Computer Networks and Communication Systems, vol. 35, 2012, pp. 16–19.
[8] A.A. Abbasi, M. Younis, A survey on clustering algorithms for wireless sensor networks, Comput. Commun. 30 (14) (2007) 2826–2841.
[9] K. Pratyay, S.K. Gupta, P.K. Jana, A novel evolutionary approach for load balanced clustering problem for wireless sensor networks, Swarm Evol. Comput. 12 (2013) 48–56.
[10] C.P. Low, C. Fang, J.M. Ng, Y.H. Ang, Efficient load-balanced clustering algorithms for wireless sensor networks, Comput. Commun. 31 (4) (2008) 750–759.
[11] K. Pratyay, P.K. Jana, Improved load balanced clustering algorithm for wireless sensor networks, in: ADCONS 2011, Lecture Notes in Computer Science, vol. 7135, 2012, pp. 399–404.
[12] G. Gupta, M. Younis, Load-balanced clustering of wireless sensor networks, in: IEEE International Conference on Communications, ICC'03, vol. 3, IEEE, 2003, pp. 1848–1852.
[13] K. Pratyay, P.K. Jana, Energy efficient load balanced clustering algorithm for wireless sensor networks, Proc. Technol. 6 (2012) 771–777.
[14] A. Bari, S. Wazed, A. Jaekel, S. Bandyopadhyay, A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks, Ad Hoc Netw. 7 (4) (2009) 665–676.
[15] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Trans. Evol. Comput. 15 (1) (2011) 4–31.
[16] R. Storn, K. Price, Differential evolution – a simple and efficient heuristic for global optimization over continuous spaces, J. Glob. Optim. 11 (4) (1997) 341–359.
[17] O. Boyinbode, H. Le, M. Takizawa, A survey on clustering algorithms for wireless sensor networks, Int. J. Space-Based Situated Comput. 1 (2) (2011) 130–136.
[18] C. Jiang, D. Yuan, Y. Zhao, Towards clustering algorithms in wireless sensor networks – a survey, in: IEEE Wireless Communications and Networking Conference, 2009, WCNC 2009, IEEE, 2009, pp. 1–6.
[19] W.B. Heinzelman, A.P. Chandrakasan, H. Balakrishnan, An application specific protocol architecture for wireless microsensor networks, IEEE Trans. Wirel. Commun. 1 (4) (2002) 660–670.
[20] S. Lindsey, C. Raghavendra, K.M. Sivalingam, Data gathering algorithms in sensor networks using energy metrics, IEEE Trans. Parallel Distrib. Syst. 13 (9) (2002) 924–935.
[21] O. Younis, S. Fahmy, HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks, IEEE Trans. Mob. Comput. 3 (4) (2004) 366–379.
[22] P. Kuila, P.K. Jana, An energy balanced distributed clustering and routing algorithm for wireless sensor networks, in: 2012 2nd IEEE International Conference on Parallel Distributed and Grid Computing (PDGC), IEEE, 2012, pp. 220–225.
[23] S.K. Gupta, P. Kuila, P.K. Jana, GAR: an energy efficient GA-based routing for wireless sensor networks, in: International Conference on Distributed Computing and Internet Technology 2013, LNCS, vol. 7753, Springer, 2013, pp. 267–277.
[24] U.K. Chakraborty, S.K. Das, T.E. Abbott, Energy-efficient routing in hierarchical wireless sensor networks using differential-evolution-based memetic algorithm, in: 2012 IEEE Congress on Evolutionary Computation (CEC), IEEE, 2012, pp. 1–8.
[25] H. Bagci, A. Yazici, An energy aware fuzzy approach to unequal clustering in wireless sensor networks, Appl. Soft Comput. 13 (2013) 1741–1749.
[26] A.K. Enan, A.A. Bara, Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks, Swarm Evol. Comput. 1 (4) (2011) 195–203.
[27] S. Hussain, A.W. Matin, O. Islam, Genetic algorithm for hierarchical wireless sensor networks, J. Netw. 2 (5) (2007).
[28] P.-C. Huruiala, A. Urzica, L. Gheorghe, Hierarchical routing protocol based on evolutionary algorithms for wireless sensor networks, in: 2010 9th Roedunet International Conference (RoEduNet), IEEE, 2010, pp. 387–392.

Please cite this article in press as: P. Kuila, P.K. Jana, A novel differential evolution based clustering algorithm for wireless sensor networks, Appl. Soft Comput. J. (2014), http://dx.doi.org/10.1016/j.asoc.2014.08.064

[29] S. Jin, M. Zhou, A.S. Wu, Sensor network optimization using a genetic algorithm, in: Proceedings of the 7th World Multiconference on Systemics, Cybernetics and Informatics, 2003, pp. 109–116.
[30] S. Mudundi, H.A. Ali, A new robust genetic algorithm for dynamic cluster formation in wireless sensor networks, in: Proceedings of Wireless and Optical Communications, Montreal, Quebec, Canada, 2007.
[31] A.M. Zungeru, L.-M. Ang, K.P. Seng, Classical and swarm intelligence based routing protocols for wireless sensor networks: a survey and comparison, J. Netw. Comput. Appl. 35 (5) (2012) 1508–1536.
[32] M. Saleem, A. Gianni, M.F. Di Caro, Swarm intelligence based routing protocol for wireless sensor networks: survey and future directions, Inform. Sci. 181 (20) (2011) 4597–4624.
[33] R.V. Kulkarni, G.K. Venayagamoorthy, Particle swarm optimization in wireless-sensor networks: a brief survey, IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 41 (2) (2011) 262–267.
[34] S. Sengupta, S. Das, M. Nasir, V. Athanasios, W.P. Vasilakos, An evolutionary multiobjective sleep-scheduling scheme for differentiated coverage in wireless sensor networks, IEEE Trans. Syst. Man Cybern. C: Appl. Rev. 42 (6) (2012) 1093–1102.
[35] F.V.C. Martins, E.G. Carrano, E.F. Wanner, R.H.C. Takahashi, G.R. Mateus, A hybrid multiobjective evolutionary approach for improving the performance of wireless sensor networks, IEEE Sens. J. 11 (3) (2011) 545–554.
[36] M. Saleem, I. Ullah, M. Farooq, *BeeSensor*: an energy-efficient and scalable routing protocol for wireless sensor networks, Inform. Sci. 200 (2012) 38–56.
[37] A.M. Zungeru, L.-M. Ang, K.P. Seng, Termite-hill: performance optimized swarm intelligence based routing algorithm for wireless sensor networks, J. Netw. Comput. Appl. 35 (6) (2012) 1901–1917.
[38] M.O. Diaz-Anadon, K.K. Leung, TDMA scheduling for event-triggered data aggregation in irregular wireless sensor networks, Comput. Commun. 34 (17) (2011) 2072–2081.
[39] P. Baronti, P. Pillai, W.C. Vince, S.C. Chook, Y. Alberto Gotta, F. Hu, Wireless sensor networks: a survey on the state of the art and the 802.15, 4 and ZigBee standards, Comput. Commun. 30 (7) (2007) 1655–1695.
[40] I. Dietrich, D. Falko, On the lifetime of wireless sensor networks, ACM Trans. Sens. Netw. 5 (1) (2009) 5.
[41] R. Madan, S. Cui, S. Lall, A. Goldsmith, Cross-layer design for lifetime maximization in interference-limited wireless sensor networks, IEEE Trans. Wirel. Commun. 5 (11) (2006) 3142–3152.
[42] M. Cardei, D.Z. Du, Improving wireless sensor network lifetime through power aware organization, Wirel. Netw. 11 (3) (2005) 333–340.
[43] J. Pan, Y.T. Hou, L. Cai, Y. Shi, S.X. Shen, Topology control for wireless sensor networks, in: Proceedings of the Ninth Annual International Conference on Mobile Computing and Networking, 2003, pp. 286–299.
[44] Z. Cai, W. Gong, C.X. Ling, H. Zhang, A clustering-based differential evolution for global optimization, Appl. Soft Comput. 11 (1) (2011) 1363–1379.