Swarm and Evolutionary Computation 12 (2013) 48–56

Contents lists available at ScienceDirect

Swarm and Evolutionary Computation

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

Regular Paper

A novel evolutionary approach for load balanced clustering problem for wireless sensor networks

Pratyay Kuila, Suneet K. Gupta, Prasanta K. Jana *

Department of Computer Science and Engineering, Indian School of Mines, Dhanbad 826004, India

ARTICLE INFO

Article history:
Received 27 November 2012
Received in revised form
2 March 2013
Accepted 8 April 2013
Available online 17 April 2013

Keywords:
Wireless sensor networks
Clustering
Load balancing
NP-hard problem
Evolutionary approach
Genetic algorithm

ABSTRACT

Clustering sensor nodes is an effective topology control method to reduce energy consumption of the sensor nodes for maximizing lifetime of Wireless Sensor Networks (WSNs). However, in a cluster based WSN, the leaders (cluster heads) bear some extra load for various activities such as data collection, data aggregation and communication of the aggregated data to the base station. Therefore, balancing the load of the cluster heads is a challenging issue for the long run operation of the WSNs. Load balanced clustering is known to be an NP-hard problem for a WSN with unequal load of the sensor nodes. Genetic Algorithm (GA) is one of the most popular evolutionary approach that can be applied for finding the fast and efficient solution of such problem. In this paper, we propose a novel GA based load balanced clustering algorithm for WSN. The proposed algorithm is shown to perform well for both equal as well as unequal load of the sensor nodes. We perform extensive simulation of the proposed method and compare the results with some evolutionary based approaches and other related clustering algorithms. The results demonstrate that the proposed algorithm performs better than all such algorithms in terms of various performance metrics such as load balancing, execution time, energy consumption, number of active sensor nodes, number of active cluster heads and the rate of convergence.

1. Introduction

Wireless sensor networks (WSNs) have attracted many researchers for their potential uses in various fields including disaster warning systems, environment monitoring, health care, safety, surveillance, intruder detection and so on [1,2]. A WSN is composed of a large number of tiny sensor nodes, which are randomly or manually deployed in a target area. The sensor nodes consist of sensing, data processing, and communicating components along with a power unit. The sensor nodes sense the target area to collect local information, process them and send it to a remote base station called sink. The sink is connected to the Internet for the public notification of the phenomena. The main bottleneck of the WSNs is the limited and irreplaceable power sources of the sensor nodes as they are operated on small batteries. Moreover, in many applications, it is almost impossible to replace the sensor nodes when their energy is exhausted. Therefore, energy consumption for the sensor nodes is the most challenging issue for the long run operation of WSNs [3–5].

Clustering is one of the most efficient techniques, which has been well researched for energy saving WSN. In a cluster based architecture (refer Fig. 1), the sensor nodes are grouped into distinct clusters with a leader, known as cluster head (CH) for each. Each sensor node belongs to only one cluster. The CHs collect and process the local data from their member sensor nodes and send it to the sink directly or via other CHs. A cluster based WSN has many advantages [6] as follows:

(1) It can reduce energy consumption significantly as only one representative (i.e., CH) per cluster needs to be involved in data aggregation and routing process.
(2) It can considerably conserve communication bandwidth as the sensor nodes need to communicate with their CHs only and can avoid exchange of redundant messages among them.
(3) The clusters can be more easily managed as they can localize the route set up and require small routing tables for the sensor nodes. This in turn improves the scalability of the network significantly.

However, in a cluster based WSN, CHs bear some extra work load contributed by their member sensor nodes as follows: (1) CHs communicate with all the sensor nodes within their cluster; (2) they perform data fusion to discard redundant and uncorrelated data sent by their member sensor nodes and finally (3) they send the processed data to the sink. Moreover, in many WSNs the CHs are usually selected amongst the normal sensor nodes which can die quickly owing to this extra work load. In this context, many researchers [7–11] have proposed the use of some special nodes

* Corresponding author. Tel.: +91 326 2235473.
E-mail addresses: pratyay_kuila@yahoo.com (P. Kuila), suneet.banda@gmail.com (S.K. Gupta), prasantajana@yahoo.com (P.K. Jana).

2210-6502/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.swevo.2013.04.002

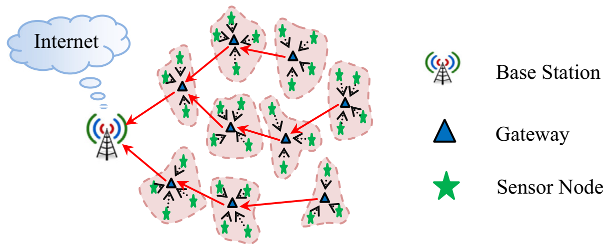

Fig. 1. A model of cluster based WSN with gateways; an arrow from a sensor node towards a gateway indicates that the sensor node has been assigned to the gateway.

called gateways or relay nodes, which are provisioned with extra energy. These gateways are treated as the cluster heads and responsible for the same functionality of the CHs. But 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. Therefore, improper cluster formation may cause some CHs overloaded. Such overload may increase latency in communication, consumes high energy of the CH and degrade the overall performance of the WSN. Therefore, load balancing of the CHs is the most important issue for clustering sensor nodes. Particularly, this is a pressing issue when the sensor nodes are not distributed uniformly. 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 of finding the optimal load balanced clustering for a large WSN seems to be very high by a brute force approach. In fact, load balanced clustering with unequal load of sensor nodes is a NP-hard problem [9]. Genetic Algorithm (GA) is one of the most suitable heuristics that can be applied for efficient load balanced clustering from such a large solution space.

In this paper, we propose a new GA based clustering algorithm to solve the above load balancing problem. The algorithm forms clusters in such way that the maximum load of each gateway is minimized. The proposed algorithm differs from the traditional GA with the following respects:

- In the phase of initial population generation, we restrict the generation of initial population by considering the connectivity between the sensor nodes and their CHs. This is in contrast to fully randomized generation of chromosomes as used in the traditional GA.
- In the mutation phase, the mutation point is selected in such a way that it generates children chromosomes that ensures better load balancing. This is in contrast to traditional GA in which mutation point is selected randomly.
- The above two strategies of initial population generation and mutation make the proposed algorithm faster than the traditional GA.

Our proposed GA can work for both of the equal and unequal load of the sensor nodes. We perform extensive simulation of our algorithm and compare the results with those of a GA based clustering algorithm [12], simple GA and another evolutionary approach, i.e., Differential Evolution (DE) [13,14]. We also compare the results with two well known algorithms namely Load Balanced Clustering (LBC) [11] and Least Distance Clustering (LDC) [15]. The experimental results are measured with respect to various performance metrics such as load balancing, execution time, energy consumption, number of active sensor nodes, number of active CHs and the rate of convergence. The comparison clearly demonstrates that the proposed algorithm performs better than all such algorithms. Here onwards, we use gateways and CHs interchangeably.

The paper is organized as follows. The related work is presented in Section 2. The energy model is described in Section 3. The WSN model and problem formulation are described in Section 4. An overview of GA is given in Section 5. The proposed algorithm and the experimental results are presented in Sections 6 and 7 respectively and we conclude our paper in Section 8.

2. Related work

A number of clustering algorithms [6,16–18] have been developed for WSN. LEACH [19] is a well known clustering technique that forms clusters by using a distributed approach. However, the method has certain disadvantages. Firstly, a node with very low energy may be selected as a CH; secondly, the CHs use single-hop communication to send the data directly to the base station. As a result, they consume more energy. Therefore, a large number of improved algorithms have been developed over LEACH such as PEGASIS [20], HEED [21], TEEN [22], TL-LEACH [23], etc. Compared to LEACH, PEGASIS promotes network lifetime, but it requires dynamic topology adjustment and the data delay is significantly high which is unsuitable for large-size networks. On the other hand, the HEED periodically selects CHs based on the node's residual energy and proximity measure of the neighbor nodes or node degree. Bandyopadhyay and Coyle [24] presented a multi-hop hierarchical clustering algorithm, but their approach does not take into account the residual energy of the sensor nodes and the CH selection may result in faster death of some sensor nodes. To form cluster, Low et al. [9] have considered a BFS 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 is very high. Their algorithm also takes substantial amount of memory space for building a breadth-first search (BFS) tree for individual sensor node. In [10], we have proposed an algorithm that runs in $O(n \log n)$ which is an improvement over [9]. Gupta and Younis [11] have proposed a load balanced clustering algorithm called LBC, which takes $O(mn \log n)$ time in worst case. In [25], 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 cluster heads (CHs) on the number of sensor nodes allotted to the CHs. Other clustering algorithms developed for WSN can be seen in [26–28].

A few evolutionary based algorithms have been reported. Huruiala et al. [29] have presented a GA based clustering and routing algorithm by choosing the optimal cluster-head and minimizing the transmission distance. Chakraborty et al. [30] have developed a protocol called GROUP in which a chain is formed to communicate with the base station. In this work, the network lifetime is increased by allowing individual sensor nodes to transmit the message to the base station in non-periodical manner depending on their residual energy and location. Thus, the approach avoids forming greedy chains. In [7], Ataul Bari et al. 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 towards that critical node to other relay node. In [31], 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 [7] 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 [7]. However, both the algorithms as presented in [7] and [31] 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. In [12], Hussain et al. have presented a GA based hierarchical clustering algorithm to choose a set of cluster-heads from the normal sensor nodes. For the fitness parameter they have used (i) direct distance, (ii) cluster distance, (iii) cluster distance—standard deviation, (iv) transfer energy and (v) number of transmission. However, their method selects CHs from normal sensor nodes and form clusters without any load balancing. As a result, some CHs may quickly die and requires frequent re-clustering. In [32], Enan et al. have presented an evolutionary aware routing protocol (EARP) for dynamic clustering of wireless sensor networks. The objective is defined as the minimization of the total dissipated energy in the network, measured as the sum of the total energy dissipated from the non-CHs to send data signals to their CHs, and the total energy spent by CH nodes to aggregate the data signals and send the aggregated signals to the base station. EARP only elects CHs using the fitness function and each non-CH determines the cluster to which it belongs by choosing the CH that requires the minimum energy consumption. EARP suffers same problem as LEACH, as some sensor node may became a CH which may not have sufficient energy. In [33], Jin et al. 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. All the methods presented in [34] and [35] also consider only the Euclidian distance between the cluster heads or BS and they did not attend any other parameters like residual energy, load balancing, standard deviation. In [34], Mudundi and Ali have proposed a genetic clustering algorithm (GCA) for dynamic formation of clusters in WSN with the goal of increasing lifetime of 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. In [35] Yang et al. have presented a method called harmony search algorithm (HAS) for increasing the life time and reducing the energy consumption in a cluster based WSN. In the fitness function, they have used the maximum Euclidean distance of the nodes and the ratio of the energy of all the sensor nodes with the total current energy of the CH in the current round. In [36], authors have presented a clustering method to optimize the energy consumption of the sensor nodes. Their method of CH selection is based on several parameters such as residual energy, communication energy, number of CHs and the distance between CHs and their member sensor nodes. However, they do not consider any load balancing of CHs. Other evolutionary based approaches applied for WSNs can be found in [37], [38] and their references inside them.

But all of the above algorithms use GA only for CH selection. To the best of our knowledge, there is no evolutionary based clustering algorithm which considers load balancing of the CHs. Our GA based clustering algorithm presented in this paper, incorporates load balancing of the CHs and has the following advantages: (1) it works for both the equal and unequal loads of the sensor nodes and (2) it converges very fast in producing efficient results.

3. Energy model

We use the same radio model for energy as [19]. In this model, both the free space and multi-path fading channels are used, depending on the distance between the transmitter and receiver. If the distance is less than a threshold d_0, the free space (fs) model is used; otherwise, the multipath (mp) model is used. Thus, the energy required by the radio to transmit an l-bit message over a distance d is given as follows.

$$E_T(l,d)=\begin{cases} lE_{elec}+l\varepsilon_{fs}d^2 & \text{for} \quad d<d_0 \\ lE_{elec}+l\varepsilon_{mp}d^4 & \text{for} \quad d\geq d_0 \end{cases} \tag{3.1}$$

where, E_{elec} is the energy required by the electronics circuit, ε_{fs} and ε_{mp} are the energy required by amplifier in free space and multi-path respectively. The radio also expends energy to receive an l-bit message given by [19].

$$E_R(l)=lE_{elec} \tag{3.2}$$

The E_{elec} depends on several factors such as the digital coding, modulation, filtering, and spreading of the signal, whereas the amplifier energy, $\varepsilon_{fs}d^2$ or $\varepsilon_{mp}d^4$, depends on the distance between the transmitter and to the receiver and the acceptable bit-error rate.

4. WSN model and problem formulation

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. 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. The sensor nodes collect the local data and send it to their corresponding gateways. On receiving the data, the gateways aggregate them to reduce the redundant data within their cluster. 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. We use the following terminologies in the proposed algorithm:

(1) The set of sensor nodes is denoted by $S=\{s_1, s_2\ldots, s_n\}$.
(2) The set of gateways is denoted by $G=\{g_1, g_2\ldots, g_m\}$, $n>m$.
(3) The traffic load contributed by each sensor node is estimated prior the cluster formation.
(4) d_i denotes the traffic load contributed by a sensor node s_i, $s_i\in S$, $d_i\in\mathbb{Q}$ where $\mathbb{Q}$ is the set of rational numbers.
(5) G_j denotes the set of gateways to which sensor node s_j may be assigned, where $s_j\in S$ and $G_j\subseteq G$. For example, $G_r=\{g_1, g_3, g_7\}$ means that s_r can be assigned to any one of the gateways, g_1, g_3, g_7.
(6) Let L_i be the load of the cluster head g_i. Then the overall maximum load of each cluster head is $L=\max\{L_i|\forall g_i\in G\}$.

Now, we address the problem of clustering, where our main objective is to minimize the overall maximum load of the gateways. Let a_{ij} be a Boolean variable such that $a_{ij}=1$, if the sensor node s_i is assigned to the cluster head g_j and $a_{ij}=0$, if it is not. Then the optimization problem of load balanced clustering in terms of Integer Linear Programming (ILP) can be formulized as follows [10]:

Minimize $L=\max\{L_i|\forall g_i\in G\}$
Subject to

$$\sum_{g_j\in G_i} a_{ij}=1|\forall s_i\in S \tag{4.1}$$

and

$$\sum_{s_i\in S} d_i\times a_{ij}\leq L|\forall g_j\in G_i \tag{4.2}$$

The constraint (4.1) states that a sensor node can be assigned to one and only one gateway and (4.2) indicates that the total load of all the sensor nodes assigned to a gateway must not exceed the overall maximum load of the gateway.

5. An overview of Genetic Algorithm

Genetic Algorithm (GA) [39] is an adaptive heuristic method that is widely applied in solving many optimization problems [40,41]. It begins with a set of randomly generated possible solutions known as initial population. An individual solution is represented by a simple string or an array of genes called chromosome/individual. The length of each chromosome in a population is equal. Each individual is evaluated by a fitness function to judge its performance. The fitness function is chosen in such a way that an individual provides the result close to the optimal solution.

Once the initial population is generated, two randomly selected chromosomes (parents) are used to produce two child chromosomes by a process called crossover in which the parent chromosomes exchange their genetic information. To produce a better solution, the child chromosomes undergo mutation operation in which their lost genetic values are restored. Whenever the crossover and mutation are over, the fitness function of the child chromosomes is evaluated and their fitness values are compared with that of all the chromosomes of the previous generation. To ensure that the current generation produces better result, two chromosomes of the previous generation with poorest fitness values are replaced with the newly generated child chromosomes. The various steps of a simple GA are depicted in the flowchart as shown in Fig. 2. It is noteworthy that generation of initial population and mutation in the proposed algorithm differs from that of the traditional GA as discussed in Section 1.

6. Proposed algorithm

We first present our methodologies for chromosome representation, initial population generation and determination of fitness function followed by selection, crossover and mutation (that are repeatedly invoked) in the subsections as follows.

6.1. Chromosome representation

We represent the chromosome as a string of gateways which indicates the assignment of all the sensor nodes to their corresponding gateway as follows. The length of each chromosome is kept same as the number of sensor nodes. For a chromosome, if ith gene value is say j, then it implies that the sensor node s_i is assigned to the gateway g_j. Note that for any other gene position the same value j can be repeated as more than one sensor nodes can be assigned to the same gateway g_j. We illustrate it with the following example.

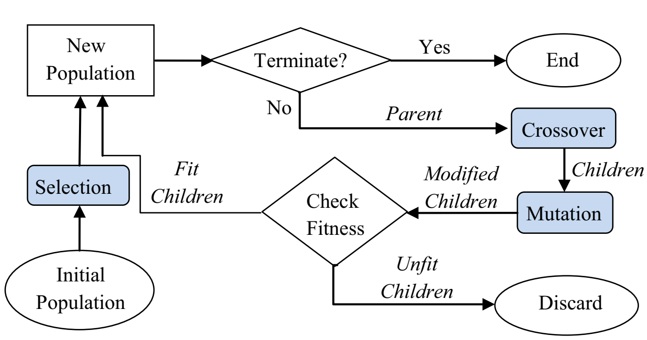

Fig. 2. Flowchart of genetic algorithm.

Example 1. Consider a WSN of 15 sensor nodes and 4 gateways, i.e., $S=\{s_1, s_2,\ldots, s_{15}\}$ and $G=\{g_1, g_2, g_3, g_4\}$. So, the length of the chromosome of this network is 15. Fig. 3 shows a chromosome representation, where the gene value at position 7 is 2 and it implies that the sensor node s_7 is assigned to the gateway g_2. Similarly, s_8, s_9 and s_{10} are assigned to g_4, g_3 and g_2 respectively in this representation.

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

6.2. Initial population

The initial population is a randomly generated set of chromosomes. Each chromosome is a sequence of gateways. The valid chromosomes are generated in such a way that the value say j of the ith position gene is randomly selected such that $g_j \in G_i$. It can be noticed that our GA based approach does not depend on any particular algorithm for generating the initial population. It also does not attempt to find a solution which can give a reasonable load balancing. It should also be noted that in the initial population, all of the generated chromosomes represent a complete clustering solution. We illustrate the idea of generation of an initial population with the following example.

Example 2. Consider a WSN of 12 sensor nodes and 4 gateways, i.e., $S=\{s_1, s_2\ldots, s_{12}\}$ and $G=\{g_1, g_2, g_3, g_4\}$. Table 1 shows all the sensor nodes and their possible gateways to which the sensor nodes can be assigned.

Here, the length of the chromosome is twelve. For the 8th gene position, a number is generated randomly amongst 1, 2 or 4. This is because s_8 can be assigned to any one of the gateways g_1, g_2 or g_4 (refer Table 1). In the same way, the 4th gene position can be a randomly generated number amongst 1, 2, 3 or 4 and the 3rd gene position can be any of 2 or 3.

In this example, suppose s_8 selects the gateway g_4 amongst g_1, g_2 and g_4. Similarly, s_4 selects g_3, s_3 selects g_2 and so on. Then with this selection, an individual of the initial population is generated which is shown in Fig. 4. It should be note that the chromosome shown in this figure represents a complete clustering solution. This is because, the entire twelve sensor nodes are assigned to their corresponding gateway and also a sensor node is assigned to only one gateway.

Remark 6.1. The above strategy of generating initial population makes the proposed algorithm converge faster than the traditional GA. The rationale behind it is that in case of traditional GA, initial population is generated randomly and this may lead generation of

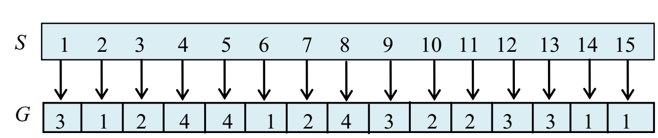

Fig. 3. Chromosome representation.

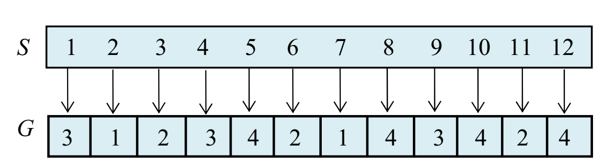

Fig. 4. Generated chromosome from Table 1.

Table 1
Sensors with the list of possible gateways.

Sensor(s_i)	Possible gateways (G_i)
s_1	$G_1=\{g_1, g_2, g_3, g_4\}$
s_2	$G_2=\{g_1, g_2\}$
s_3	$G_3=\{g_2, g_3\}$
s_4	$G_4=\{g_1, g_2, g_3, g_4\}$
s_5	$G_5=\{g_2, g_3, g_4\}$
s_6	$G_6=\{g_1, g_2, g_3, g_4\}$
s_7	$G_7=\{g_1, g_2, g_3\}$
s_8	$G_8=\{g_1, g_2, g_4\}$
s_9	$G_9=\{g_1, g_3\}$
s_{10}	$G_{10}=\{g_2, g_3, g_4\}$
s_{11}	$G_{11}=\{g_1, g_2, g_3, g_4\}$
s_{12}	$G_{12}=\{g_1, g_2, g_3, g_4\}$

many invalid chromosomes which are discarded. This results the process of selection slower.

6.3. Fitness function

We build a fitness function to evaluate the individual chromosomes of the initial population as follows. We note that the load balancing of the gateway not only minimizes the maximum load of a gateway but also concentrates on the load distribution among all the gateways. Therefore, we construct the fitness function on the basis of the standard deviation (σ) of the gateway load which gives even distribution of the load per cluster. If there are m gateways and n sensor nodes, the standard deviation of gateway load is given by

$$\sigma=\sqrt{\frac{\sum_{j=1}^{m}(\mu-W_j)^2}{m}}$$

where, $\mu(average\ load)=\sum_{i=1}^{n}d_i/m$, d_i is the load of the sensor node s_i and W_j is the overall load of the gateway g_j. The smaller the standard deviation, the higher is the fitness value. Therefore, the fitness function is chosen as the reciprocal of the standard deviation of the gateway load, i.e. $Fitness=1/\sigma$.

6.4. Selection

The selection process determines which of the chromosomes from the current population will mate (crossover) to create new chromosomes. For the selection process, we select some valid chromosomes with higher fitness value. The individuals with better fitness values have better chances of selection. There are several selection methods, such as Roulette-wheel selection, rank selection, tournament selection and so on. We use tournament selection in our method for selecting the chromosomes with best fitness values from the population. The selected chromosomes are applied to produce new child chromosomes (offspring) by the crossover operation as described in the following section.

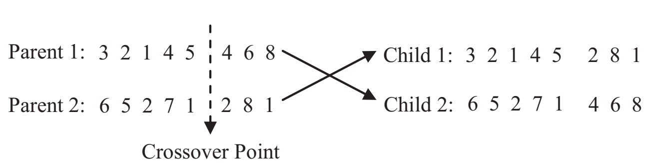

Fig. 5. Crossover operation.

6.5. Crossover

The crossover operation takes place between two randomly selected chromosomes. To produce the new offspring from the selected parent chromosomes, we use 1-point crossover whereby a point is chosen at random, and the two parent chromosomes exchange their information after that point. The whole process is shown in Fig. 5.

Lemma 6.1. The two child chromosomes produced by the above crossover operation is valid.

Proof. A valid chromosome is the one which corresponds to a cluster of the sensor nodes such that for each sensor node the assigned gateway is selected from their gateway list. As mentioned in Section 6.2, the chromosomes are generated in such a way that the value j of the gene position i is randomly selected so that $g_j \in G_i$. Therefore, for each sensor node the corresponding gateway is valid.

6.6. Mutation

The mutation is applied at a selected gene rather than randomly selected gene. Here, our main purpose is to balance the load of the gateways. We select that gateway from the chromosome that has maximum load. As more than one sensor nodes are assigned to the maximum loaded gateway, therefore its number appears in several positions of the chromosome. From all of these gene positions, we randomly select a gene and replace another gateway number to that gene. It means that we just replace a randomly selected sensor node from that maximum loaded gateway to other less loaded gateway. It can be noted that the replaced number at this gene position must be such that the sensor node is

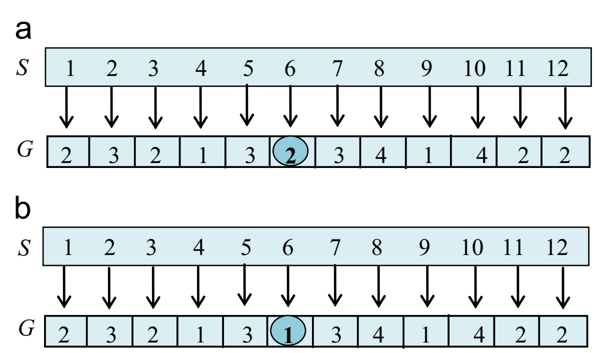

Fig. 6. Chromosome representation: (a) before mutation; (b) after mutation.

Table 2
Parameters of simulation.

Parameter	Value
Area	200×200 m^2
Base station location	(215, 100)
Sensor nodes	100–400
Gateways	15–30
Initial energy of sensor nodes	2.0 J
Number of simulation iterations	100
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

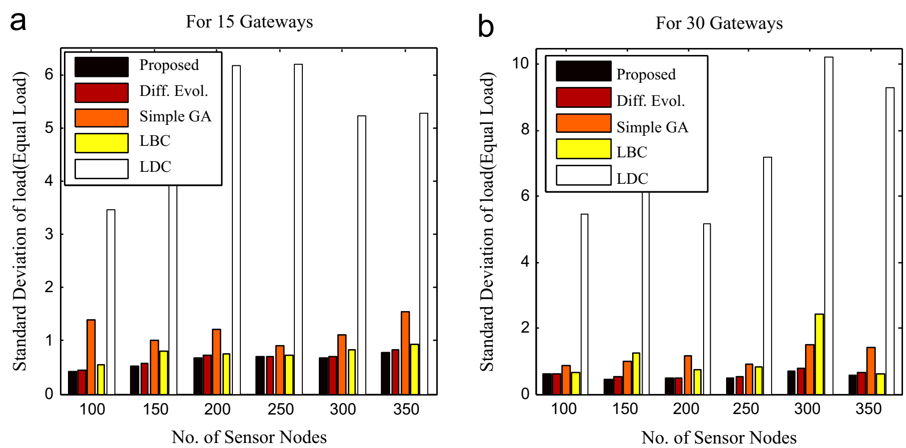

Fig. 7. Comparison of the proposed method with DE approach, simple GA, LBC and LDC in terms of load balancing for equal load of the sensor nodes: for (a) 15 gateways and (b) 30 gateways.

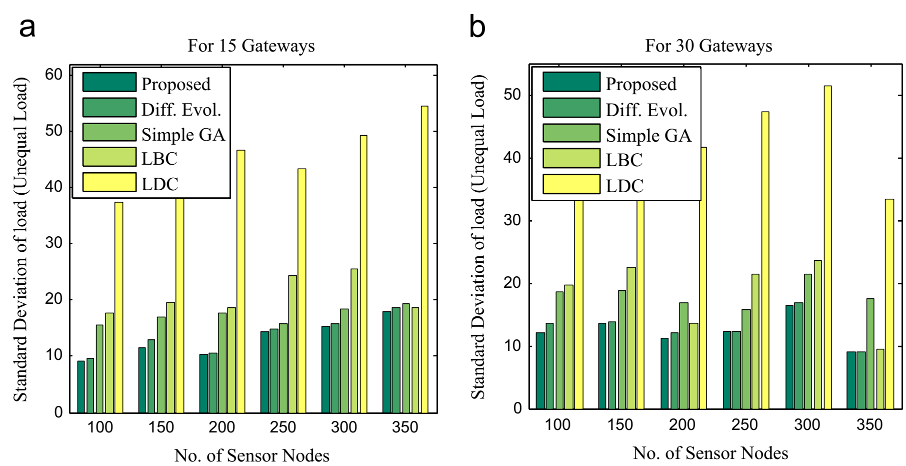

Fig. 8. Comparison of the proposed method with DE approach, simple GA, LBC and LDC in terms of load balancing for unequal load of the sensor nodes: for (a) 15 gateways and (b) 30 gateways.

replaced to another gateway from the sensor's gateway list. We illustrate the idea with the figurative example as follows.

Example 3. Consider the chromosome representation after crossover operation as shown in Fig. 6(a) for the same network scenario as Example 2. It is shown in Fig. 6(a) that the gateway g_2 is assigned with five sensor nodes and it is maximum. In mutation operation, the proposed method randomly selects a gene position, which is occupied by the gateway g_2 and replaces it by another gateway number. Note that the replaced gateway must be from the gateway list of the corresponding sensor node. Let a randomly selected gene position be 6 and it is occupied by the gateway number 2. Now, 2 is replaced by another gateway number from G_6 (refer Table 1). In this case, a randomly selected gateway number is 1, which is randomly selected amongst 1, 3 and 4 as per the gateway list G_6. The number 2 is not taken as it is the number which has to be replaced. Fig. 6(b) shows the resultant chromosome after mutation operation.

Lemma 6.2. The new chromosome produced by the above mutation process is valid.

Proof. According to the chromosome representation, the maximum loaded gateway lies in more than one gene position of the chromosome. At the time of mutation, the algorithm randomly selects the gene position which is occupied by the maximum loaded gateway number and replaces a new gateway number which is also randomly selected from the gateway list of the corresponding sensor node. Since all valid offspring are generated in crossover operation, the mutation operation cannot hamper the validity of these offspring by replacing a new gateway.

Remark 6.2. This can be noted that the above strategy of mutation makes our GA based approach to converge faster than the traditional GA. This is because, in case of traditional GA, mutation point is selected randomly. This may lead generation of poor chromosomes and slower convergence rate.

7. Experimental results

The proposed algorithms were experimented extensively using MATLAB (version 7.5) on an Intel Core 2 Duo processor with T9400 chipset, 2.53 GHz CPU and 2 GB RAM running on the platform Microsoft Windows Vista. For the experiments, we assumed a WSN scenario in which the sensor nodes were deployed along with the gateways in a 200×200 m^2 area. We ran the algorithms by varying the sensor nodes from 100 to 400 and the number of gateways from 15 to 30. Each sensor node is assumed to have an initial energy of 2 J and gateways have 10 J. In the simulation run, the typical parameter values were set same as [19] as shown in Table 2.

To execute our proposed algorithm, we considered an initial population of 200 chromosomes. For crossover operation, we

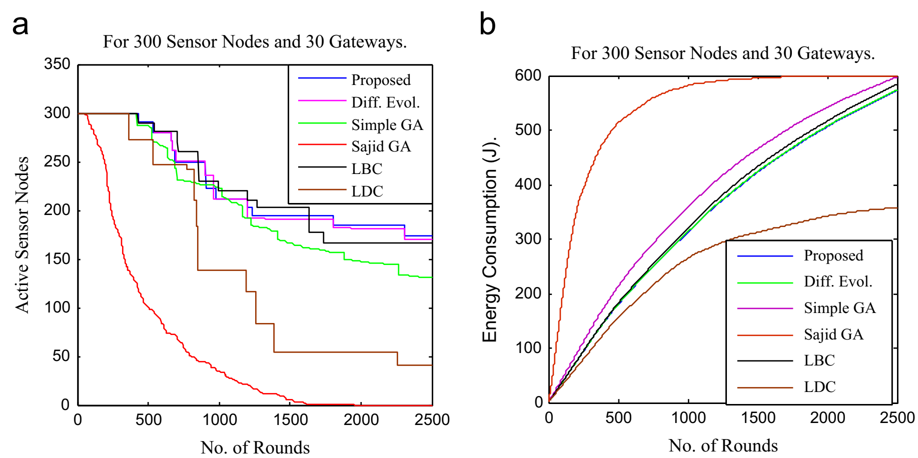

Fig. 9. Comparison of the proposed method with DE approach, simple GA, LBC, LDC and GA based clustering algorithm by Sajid in terms of (a) active sensor nodes and (b) energy consumption.

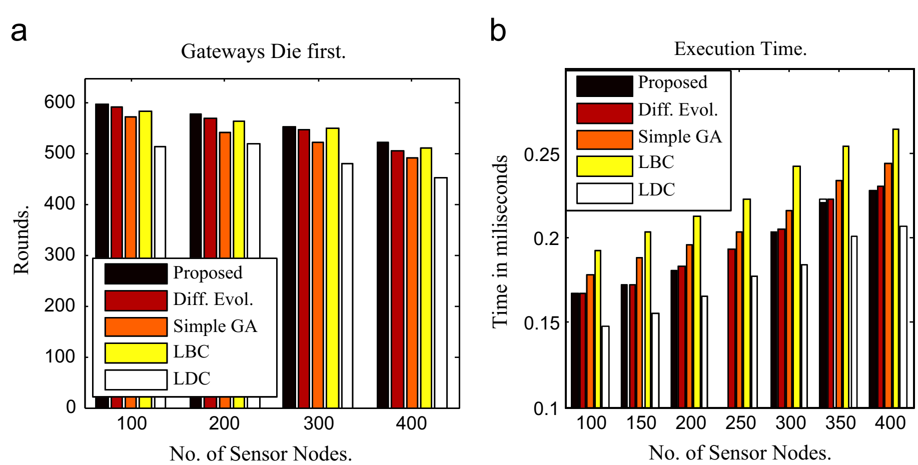

Fig. 10. Comparison of the proposed method with DE approach, simple GA, LBC and LDC in terms of (a) first gateway dies in the network and (b) execution time.

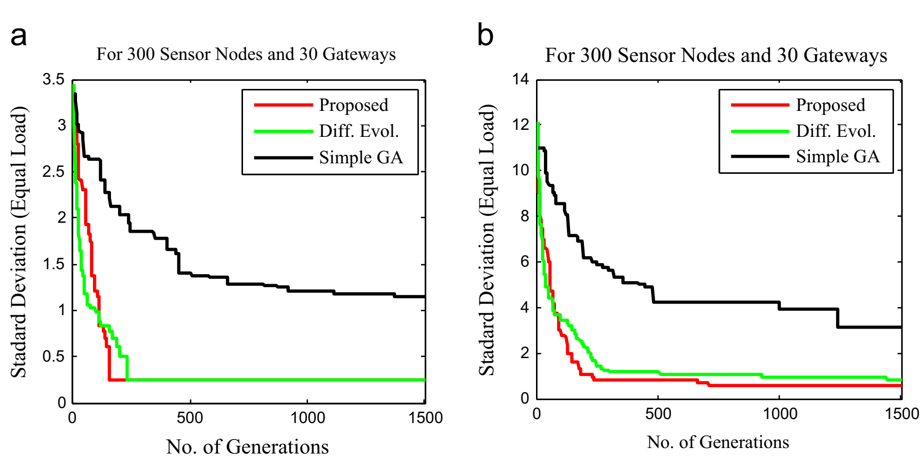

Fig. 11. Convergence rate of the proposed algorithm for (a) unequal load of the sensor nodes and (b) equal load of the sensor nodes.

selected the best 10% chromosomes using tournament selection. In our simulation, crossover rate was taken as 0.7 and mutation rate as 0.05. The algorithm was run for 75–500 iterations. For the comparison purpose, we also ran three evolutionary based approaches, i.e., Sajid GA [12], Simple GA, Differential Evolution (DE) [13], [14]. We also ran two other related clustering algorithms, i.e., LBC [11] and LDC [15]. In the experiment using DE, we created the donor vector by selecting the best chromosome from the population and used the same mutation operation as applied in our proposed method. The crossover rate (CR) of the DE was taken 70%. In the simple GA, the fitness function, population size, crossover rate, mutation rate and all other assumptions were taken same as that of the proposed GA.

In order to judge the quality of the load balancing, we calculated the standard deviation of the gateway loads and plotted against the number of sensor nodes. The standard deviation of the gateway load gives even distribution of the load per gateway. First, we ran the algorithms for equal load of the sensor nodes by varying the sensor nodes from 100 to 350 and the number of gateways for 15 and 30. It can be observed that our proposed method produces better load balancing for equal load of the sensor nodes than others as shown in Fig. 7(a) and (b). As the algorithm by Hussain et al. [12] does not consider any load balancing of the CHs, we have not compared its results in these figures.

We next ran the algorithms for unequal load of the sensor nodes, the comparison results of which are depicted in Fig. 8 (a) and (b). The proposed method performs well for unequal load for this case too.

Fig. 9(a) shows comparison of number of active sensor nodes per round. A sensor node is considered as active if its existing energy is not zero and also there must be at least one gateway within its communication range. Sometimes few CHs die quickly for improper load balancing. As a result, few sensor nodes are unable to find any CH within their range, though the sensor nodes still may have some existing energy. In our scenario, this type of sensor nodes is also considered as inactive. It is observed that our proposed algorithm outperforms simple GA, LDC and the GA based clustering algorithm [12] in terms of number of active sensor nodes. It also performs better than DE and LBC in terms of number of active sensor nodes as shown in Fig. 9(a). Fig. 9(b) shows the energy consumption of the network per round. As LDC assign the sensor nodes to their nearest CH, it consumes less energy than other algorithms but all high loaded CHs die quickly and many sensor nodes became inactive. It can be noted from Fig. 9(b) that our proposed algorithm performs better than simple GA, DE, LBC and the GA based clustering [12] in terms of energy consumption.

Fig. 10(a) shows the comparison of the gateways which die first. It can be observed that our method always performs better than the others. In this case LDC always performs very poorly. This is due to the fact that LDC does not consider load balancing of the gateways. Therefore, the high loaded gateway dies very quickly in LDC. We also obtained the execution time of the algorithms in which the proposed algorithm is better than DE, simple GA and LBC in terms of execution time as clear from Fig. 10(b). LDC always shows better execution time than all the methods. This is due to simply assigning a sensor node to the gateway which is nearest to it. Thus LDC requires less computation time than others.

Fig. 11 shows the comparison of the convergence rate of our proposed GA, simple GA and the DE approach. We ran our algorithm for 300 sensor nodes and 30 gateways. Fig. 11(a) and (b) shows the convergence rate for equal and unequal load of the sensor nodes respectively which clearly shows the faster convergence of our algorithm.

8. Conclusions

In this paper, we have presented a GA based load balanced clustering algorithm for WSN. The algorithm has been described with proper chromosome representation, generation of initial population, selection process, followed by the crossover and mutation operations. The experimental results have shown that the performance of the algorithm is better than the GA based clustering algorithm, simple GA, Differential Evolutionary approach, Load Balanced Clustering (LBC) and the Least Distance Clustering (LDC) algorithm in terms of load balancing of the gateways for equal as well as unequal load of the sensor nodes. It is observed that the proposed algorithm provides superior performance in terms of energy consumption, number of active sensor nodes, rate of convergence and the execution time.

References

[1] I.F. Akyildiz, et al., Wireless sensor networks: a survey, Computer Networks 38 (4) (2002) 393–422.
[2] Y. Jennifer, et al., Wireless sensor network survey, Computer Networks 52 (2008) 2292–2330.
[3] A. Giuseppe, et al., Energy conservation in wireless sensor networks: a survey, Ad Hoc Networks 7 (2009) 537–568.
[4] C.Y. Chong, S.P. Kumar, Sensor networks: evolution, opportunities, and challenges, Proceedings of the IEEE 91 (8) (2003) 1247–1256.
[5] L. Emanuele, et al., Energetic sustainability of routing algorithms for energy-harvesting wireless sensor networks, Computer Communication 30 (2007) 2976–2986.
[6] A.Hamed Abbasi, Y.A. Mohamad, Survey on clustering algorithms for wireless sensor networks, Computer Communications 30 (2007) 2826–2841.
[7] Ataul Bari, et al., A genetic algorithm based approach for energy efficient routing in two-tiered sensor networks, Ad Hoc Networks 7 (4) (2009) 665–676.
[8] J. Tang, et al., Relay node placement in large scale wireless sensor networks, Computer Communications 4 (29) (2006) 490–501.
[9] C.P. Low, et al., Efficient load-balanced clustering algorithms for wireless, Computer Communications 31 (2008) 750–759.
[10] K. Pratyay, K.J. Prasanta, Improved load balanced clustering algorithm for wireless sensor networks. ADCONS 2011, Lecture Notes in Computer Science 7135 (2012) 399–404.
[11] G. Gupta, M. Younis, Load-balanced clustering of wireless sensor networks, in: Proceedings of IEEE International Conference, ICC '03, vol. 3 (2003) pp. 1848–1852.
[12] S. Hussain, et al., Genetic Algorithm for hierarchical wireless sensor networks, Journal of Networks 2 (5) (2007) 87–97.
[13] S. Das, P.N. Suganthan, Differential evolution: a survey of the state-of-the-art, IEEE Transaction on Evolutionary Computation 15 (1) (2011) 4–31.
[14] R. Storn, K. Price, Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces, Journal of Global Optimization 11 (1997) 341–359.
[15] B. Ataul, et al., Clustering strategies for improving the lifetime of two-tiered sensor networks, Computer Communications 31 (2008) 3451–3459.
[16] B. Olutayo et al., A survey on clustering algorithms for wireless sensor networks, in: Proceedings of 13th IEEE International Conference on Network-Based Information Systems (2010) pp. 358–364.
[17] J. Congfeng et al., Towards clustering algorithms in wireless sensor networks—a survey, in: Proceedings of IEEE Wireless Communications and Networking Conference (WCNC) (2009) pp. 1–6.
[18] D.J. Dechene et al., Wireless sensor networks—a survey of clustering algorithms for wireless sensor networks. Project Report, Department of Electrical and Computer Engineering, The University of Western Ontario, Canada (2006).
[19] Heinzelman, et al., Application specific protocol architecture for wireless microsensor networks, IEEE Transactions on Wireless Communications 1 (4) (2002) 660–670.
[20] S. Lindsey, C.S. Raghavendra, PEGASIS: power efficient gathering in sensor information systems, in: Proceedings of the IEEE Aerospace Conference, vol. 3 (2003) pp. 1125–1130.
[21] O. Younis, S. Fahmy, HEED: a Hybrid, Energy-Efficient, Distributed clustering approach for Ad Hoc sensor networks, IEEE Transaction on Mobile Computing 3 (2004) 366–379.
[22] A. Manjeshwar, D.P. Agrawal, TEEN: a routing protocol for enhanced efficiency in wireless sensor networks, in: Proceedings of 15th International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing, San Francisco (2002) pp. 2009–2015.
[23] V. Loscri et al., A two-level hierarchy for low-energy adaptive clustering hierarchy, in: Proceedings of the IEEE Vehicular Technology Conference, vol. 3 (2005) pp. 1809–1813.
[24] S. Bandyopadhyay, E.J. Coyle, An energy efficient hierarchical clustering algorithm for wireless sensor networks, in: Proceedings of IEEE INFOCOM, vol. 3 (2003) pp. 1713–1723.
[25] P. Kuila, P.K. Jana, Energy efficient load-balanced clustering algorithm for wireless sensor network. ICCCS 2012, Procedia Technology 6 (2012) 771–777.
[26] Kuila P., Jana P.K., An energy balanced distributed clustering and routing algorithm for wireless sensor networks, in: Proceedings of IEEE Xplore International Conference, PDGC 2012 (2012) pp. 220–225.
[27] J. Shujuan, L. Keqiu, LBCS: a Load Balanced Clustering Scheme in wireless sensor networks, in: Proceedings of Third International Conference on Multi-media and Ubiquitous Engineering (2009) pp. 221–225.
[28] M. Ziyadi et al., Adaptive clustering for energy efficient wireless sensor networks based on ant colony optimization, in: Proceedings of the Seventh Annual Communication Networks And Services Research Conference (2009) pp. 330–334.
[29] P.C. Huruiala et al., Hierarchical routing protocol based on evolutionary algorithms for wireless sensor networks, in: Proceedings of the Roedunet International Conference (RoEduNet) (2010).
[30] A. Chakraborty, et al., A genetic algorithm inspired routing protocol for wireless sensor networks, International Journal of Computational Intelligence Theory and Practice 6 (1) (2011).

[31] K.G. Suneet, et al., GAR: an energy efficient ga-based routing for wireless sensor networks, Lecture Notes in Computer Science 7753 (2013) 267–277.
[32] A. Enan, et al., Energy-aware evolutionary routing protocol for dynamic clustering of wireless sensor networks, Swarm and Evolutionary Computation 1 (2011) 195–203.
[33] S. Jin et al., Sensor network optimization using a genetic algorithm, in: Proceedings of 7th World Multiconference on Systemics, Cybernetics and Informatics, Orlando, 30 March–2 April (2003) pp. 109–116.
[34] S. Mudundi, H.H. 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).
[35] X.S. Yang, Harmony search as a metaheuristic algorithm, in: Z.W. Geem (Ed.), Music-Inspired Harmony Search Algorithm: Theory and Applications; Studies in Computational Intelligence, vol. 191, Springer, Berlin, 2009, pp. 1–14.
[36] A.M. Moslem, Design and implementation a new energy efficient clustering algorithm using genetic algorithm for wireless sensor networks, World Academy of Science, Engineering and Technology 52 (2011) 430–433.
[37] S. Sengupta, S. Das, Md. Nasir, A.V. Vasilakos, W. Pedrycz, An Evolutionary Multiobjective Sleep-Scheduling Scheme for Differentiated Coverage in Wireless Sensor Networks, IEEE Transactions on Systems, Man, and Cybernetics-Part C 42 (6) (2012) 1093–1102.
[38] 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 Sensors J. 11 (3) (2011) 545–554.
[39] D.E. Goldberg, Genetic Algorithms: Search Optimization and Machine Learning, Addison Wesley, Massachusetts, 2007.
[40] F.A. Buthainah, A.A. Hamza, Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA), World Academy of Science, Engineering and Technology 38 (2008) 296–302.
[41] H.A. Zakir, Genetic algorithm for the traveling salesman problem using sequential constructive crossover operator, International Journal of Biometrics & Bioinformatics (IJBB) 3 (6) (2010) 96–105.