International Journal of Embedded and Real-Time Communication Systems
Volume 12 • Issue 1 • January-March 2021

A Modified GA-Based Load Balanced Clustering Algorithm for WSN: MGALBC

Mohit Kumar, Cambridge Institute of Technology, India
Dinesh Kumar, Jaipur National University, India
Md Amir Khusru Akhtar, Usha Martin University, India

ABSTRACT

The prevalent applications of WSN have fascinated a plethora of research efforts. Sensor nodes have serious limitations such as battery lifetime, memory constraints, and computational capabilities. Clustering is an important method for maximizing the network lifetime. In clustering, a network is divided into virtual groups, and CHs send their data to the BS either directly or using multi-hop routing. CHs are some special nodes having more energy than normal nodes. In fact, these special nodes are also battery operated and consequently power constrained; thus, they play a vital role in network lifetime. Cluster formation is very important and improper design may cause overload. This paper presents a modified GA-based load balanced clustering (MGALBC) algorithm for WSN. It is better than GA-based load balanced clustering (GALBC) algorithm because it balances the load by considering the residual energy. The result shows that the proposed method is better than GALBC in terms of energy consumption, number of active sensor nodes and network life.

KEYWORDS

Chromosome, Cluster Head, Gene, Genetic Algorithm, Integer Programming, Load Balancing, Residual Energy, Sensor Node, Wireless Sensor Network

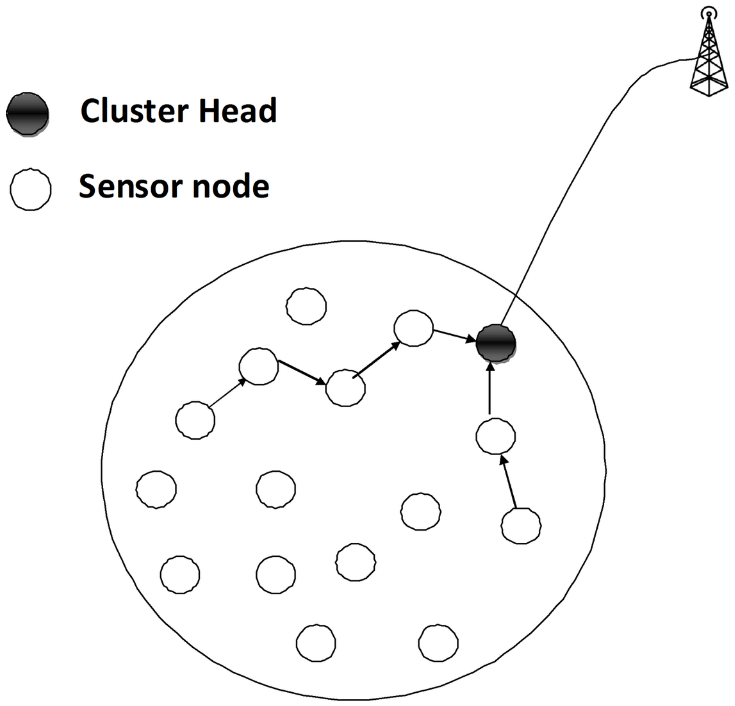
1. INTRODUCTION

Wireless sensor networks (WSN), are geographically distributed autonomous sensors which are deployed either arbitrarily or using some predefined provision. It is used to monitor the physical or environmental characteristics such as temperature, pressure, humidity, sound etc. shown in Figure 1. The collected data is cooperatively forwarded to the base station for application specific decisions. Furthermore sensor nodes have serious limitations in terms of battery lifetime, memory constraints and computational and communication capabilities.

Lots of works have been proposed in the field of energy efficient clustering and routing but they have serious limitations in terms of implementation complexity, load balancing, data fusion and the energy conservation.

DOI: 10.4018/IJERTCS.20210101.oa3

Figure 1. Sensor network



In a WSN most of the energy is consumed in transmission. In order to maximize network lifetime we divide the network into virtual groups and CHs send their data to the BS either directly or using multi-hop routing. We have several ways for the selection of CHs, here we have assumed our CHs are some special nodes having more energy than normal nodes. In fact these special nodes are also battery operated and consequently power constrained, thus play a vital role in network lifetime. Cluster formation is very important and improper design may cause overload. Thus, overloaded CHs increases latency in communication and consumes more energy in turn minimizes the performance of the network.

In literature, we have several methods for load balancing, such as GA based load balanced clustering problem for wireless sensor networks (GALBC) protocol (Kulia et al., 2013). They have used GA for minimizing the maximum load of each gateway. The proposed algorithm differs from the traditional GA because it generates children chromosomes that ensures better load balancing where as in traditional GA in which mutation point is selected randomly. The proposed strategy of generating initial population makes the proposed algorithm converges faster than the traditional GA but, it has a serious drawback. This method balances the load of the gateways without considering their residual energy. The proposed method is not practical because it forcibly balances the load and in turn chooses the incorrect node which may create network failure.

Our presented algorithm Energy Efficient Load Balanced Clustering Algorithm for WSN is a GA based load balanced clustering-based protocol which is better than GALBC because it balances the load by considering the residual energy. We have used the same method and design as GALBC but we have added residual energy as a constraints load balancing. We have conducted grievous simulations in our indigenous tool developed in C programming language. The result shows that our proposed method is better than GALBC in terms of energy consumption, number of active sensor nodes and network life. The rest of the paper is organized as follows. The related work is explained in Section 2. Section 3 describes the proposed model. Simulation results are presented in Section 4. Finally, Section 5 concludes the paper.

2. RELATED WORK

Lots of works have been proposed in the field of clustering and energy efficiency for WSNs. Liu X (2012) proposed a survey on clustering routing protocols in wireless sensor networks. Here, we are presenting some of the review and research work on this topic.

Low Energy Adaptive Clustering Hierarchy (LEACH) protocol (Heinzelman, 2000) is a popular TDMA based MAC protocol which improves the lifespan of WSN. LEACH protocol uses two phases namely set-up phase and steady phase. It balances the load of routing by dynamically rotating the workload of the CHs between the sensor nodes. On the other hand, the limitation of this approach is that it selects a node as CH without considering its residual energy. In addition to that in LEACH a CH communicates with the base station in a single hop.

Some of the algorithms (Liu et al., 2008; Ali et al., 2008; Al-Refai et al., 2011; Tyagi & Kumar, 2013; Kulia & Jana, 2012; Gupta et al., 2017; Han et al., 2017; Nayak & Vathasavai, 2017) have been proposed for clustering and routing to improve clustering protocol but it has serious connectivity issues with CHs.

Huruiala et al. (2010) have presented a GA based clustering and routing algorithm designed to extend the life of the network. It minimizes the energy consumption and latency by choosing the best nodes as cluster-heads. This algorithm uses a multi-objective genetic algorithm on the base station and then communicates with the network.

A clustering algorithm based on genetic algorithm has been proposed by Mehr, M. A. (2011). It uses different parameters such as residual energy, required energy needed to send a message and number of cluster heads to increase the network lifetime, but they do not consider any load balancing of CHs.

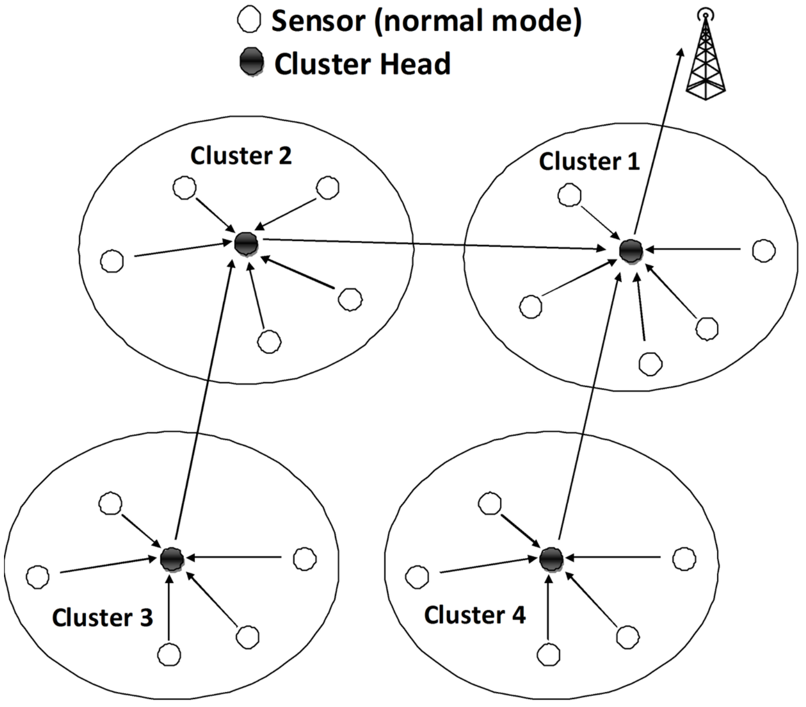
Kong et al. (2017) proposed a genetic algorithm based energy-aware routing protocol for a middle layer oriented wireless sensor network. The proposed design has two phases, selecting candidate middle layer phase and genetic algorithm phase. The author claims that the design lowers the traffic of the relay stations with full coverage.

Yuan et al. (2017) proposed a GA based, self-organizing network clustering (GASONeC) method that introduces a framework for dynamic optimization of wireless sensor node clusters. GASONeC uses residual energy, expected energy expenditure, distance to the base station, and the number of nodes for searching an optimal and dynamic network structure. This method enhances network life up to 43.44% because node density greatly affects the network longevity.

Kumar & Rai (2017) proposed an energy efficient and optimized load balanced localization method using CDS with one-hop neighborhood and genetic algorithm in WSNs. The proposed algorithm uses genetic algorithm for balancing and calculating the computational load among anchor nodes for location calculation. They have used an optimized backbone to locate the unknown nodes. This algorithm improves the network lifetime because it distributes the computational load efficiently among the anchor nodes.

Kulia et al. (2013) proposed a GA based clustering algorithm to solve load balancing problem. In this work they have used GA for minimizing the maximum load of each gateway. The proposed algorithm differs from the traditional GA because it generates children chromosomes that ensures better load balancing where as in traditional GA in which mutation point is selected randomly. Thus, the proposed strategy of generating initial population makes the proposed algorithm converges faster than the traditional GA. We have used this paper as our base paper.

Figure 2. Proposed Architecture



3. PROPOSED ARCHITECTURE

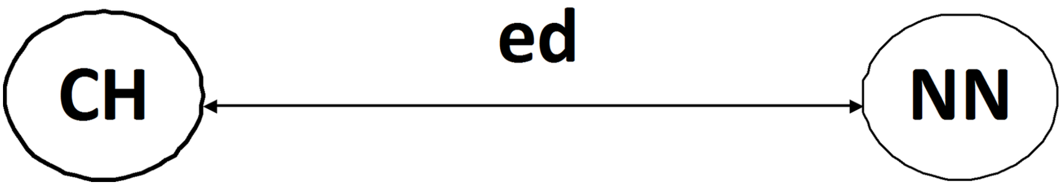
The proposed design is very simple and efficient, and meets the requirements of wireless micro sensor networks. This work involves a set of nodes (micro sensors) deployed manually or randomly into the target area. In our proposed WSN model there are there are two types of nodes in the network, sensor nodes and less energy constraint cluster heads. Sensor nodes are accountable to sense local data and send it to their respective CHs. While, the CHs receive the data from their member sensor nodes, aggregate the received data and forward them to their next-hop Cluster Head (CH) or towards the Base Station (BS). We are assuming the all sensor nodes are stationary after deployment. A scenario of the proposed architecture is given in Figure 2. In WSN individual nodes data are correlated to obtain a meaningful result using a high-level function of the data that describes the events occurring in the environment.

In this work we proposed a GA based method to minimize energy dissipation in wireless sensor networks. The method is cluster-based approach like LEACH. We have two phases in our work namely set-up phase and steady-state phase.

Set-up Phase: In the set-up phase one or more clusters are created on the basis of single hop neighbors or one hop distance. The one hop distance is computed on the basis of CH and normal node locations as shown in Figure 2. A cluster contains cluster head and sensor nodes (SN) / normal nodes (NN). It is similar to LEACH and the set-up phase is performed only one time. Normal nodes are assigned to the clusters based on their distances to the CHs. These normal nodes join into the clusters on the basis of one hop distance shown in Figure 3.

Let CH = (x_1, y_1) and NN = (x_2, y_2), then the Euclidean distance is computed using:

Figure 3. Euclidean distance (d) between CH and NN

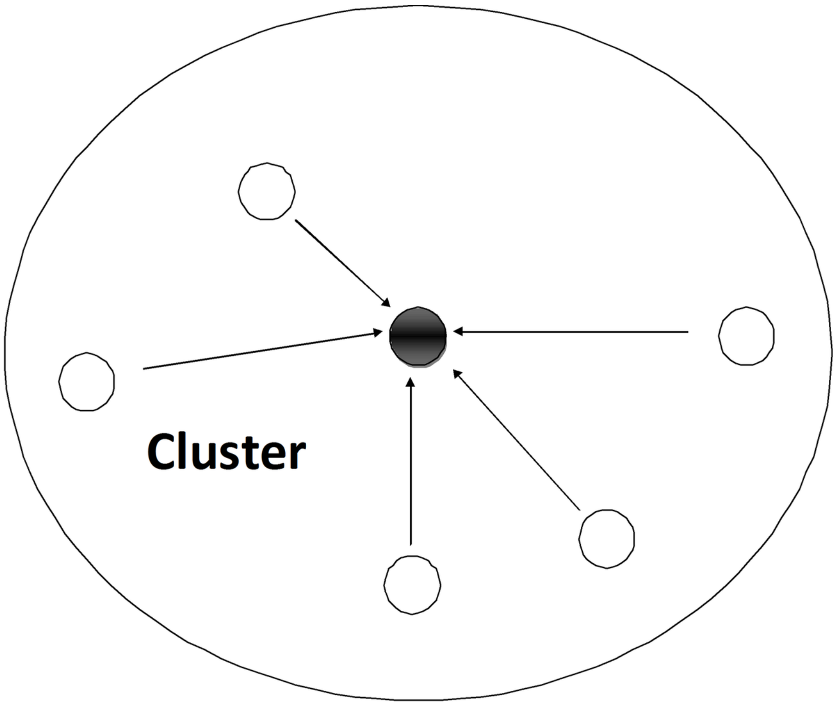


$$ed = \sqrt{\left(x_1 - x_2\right)^2 + \left(y_1 - y_2\right)^2} \tag{1}$$

A sensor is in the coverage area of a cluster if Euclidean distance is less than or equal to the threshold communication range. On the other hand, when a NN is within the coverage of two or more CHs, then the minimum Euclidean distance is the basis for selecting a NN as a member for the cluster. If two or more NNs are at the same distance from the NN, in that case the NN is added to any one of the CHs. A cluster with CH and NNs is shown in Figure 4, in which NNs or SNs are in the coverage area of CH and they are at one hop distance.

Steady-state Phase: In this phase, nodes communicate directly with their CHs using Time Division Multiple Access (TDMA) technology. TDMA allows multiple accesses and share same radio channel by dividing each channel into time slots to facilitate data transmissions. For communication with the base station the CH receives from all nodes, then fuses the data packets

Figure 4. Cluster



into one packet and sends it to the base station. If all CHs send their data to the BS, a round is completed.

3.1. Energy Model

In this paper we have used the same radio model for energy as discussed by Heinzelman (2000). This model takes the distance between the transmitter and receiver to choose the free space and multi-path fading channels are used. If the distance is less than a threshold dcross, then it choose free space (frs) model else the multipath (mpath) model. Thus, to transmit l-bit message over a distance d the energy required by the radio is given as follows:

$$E_{\Gamma}(l,d) = \begin{cases} lElec + 1\varepsilon frs\ d^2 : d < d_{cross} \\ lElec + 1\varepsilon mpath\ d^4 : d \geq d_{cross} \end{cases}$$

In order to receive this message, the radio expends:

$$E_{RX}(l) = lElec$$

The Eelec is the electronic energy required by the circuit, and it depends on several factors such as digital coding, modulation, and filtering of the signal. The parameters εfrsd2 and εmpathd4 are the energy required by amplifier in free space and multipath respectively and it depends on the required receiver sensitivity and noise.

3.2. Network Model

In this article, we assume a sensor network model similar to GALBC (Kulia et al., 2013), but we have added residual energy as a constraint for load balancing. It has following properties:

- There is a fixed base station located far away from the sensor nodes. In the study, we do not consider the energy consumption of the BS and assume that it has sufficient energy supply.
- The set of sensor nodes is denoted by $S= \{s_1, s_2\ldots, s_n\}$. All sensor nodes have same configuration and limited energy. Each node has the same starting energy and they are generating equal traffic in the network. All nodes are stationary.
- The set of Cluster Head is denoted by $CH= \{CH_1, CH_2, CH_3,\ldots, CH_m\}$.
- d_i denotes the traffic load contributed by the sensor node s_i, s_i Î S.
- CH_j denotes the set of CHs to which sensor node s_j may be assigned.
- E_i denotes lifetime of the cluster Head CH_i, calculated using Energy consumption per round *ECon_P_Round(CH$_i$)* and residual energy of CH_i *EResidual(CH$_i$)*:

Ei = *EResidual(Chi) / ECon_P_Round(CHi)*

- T denotes threshold energy required by a CH to participate in transmission.
- L_i be the load of the cluster head CH_i and maximum overload is denoted as:

$$Minimize\ L = \max\left\{L_i - E_i \middle| \forall CH_i \in CH\right\}$$

Now, we address the problem of traffic load and residual energy:

Goal: The objective function is to minimize the overall maximum load of the clusters on the basis of residual energy.

Let C_{ij} be a Boolean variable such that C_{ij} =1, if the sensor node s_i is assigned to the cluster head CH_j and C_{ij} =0, if it is not. Then the problem in terms of linear programming approach can be formulized as follows:

$$Minimize\ L = \max\left\{L_i - E_i \middle| \forall CH_i \in CH\right\}$$

subject to:

$$\sum_{CH_j \in CH} C_{ij} = 1 \middle| \forall s_i \in S \tag{2}$$

$$\sum_{s_i \in S} d_i \times C_{ij} \leq L = 1 \middle| \forall CH_j \in CH \tag{3}$$

$$Ei > T \tag{4}$$

The above constraints (2), (3) is similar to GALBC with an additional constraint (4). Here constraint (2) indicates a sensor node can be assigned to only one CH, constraint (2) describes that the total load all the sensor nodes assigned to a CH and should be less than or equal to the overall maximum load of the gateway and the third constraint denotes lifetime of CH_i should be greater than or equal to some threshold energy required by a CH to participate in transmission.

3.3. Genetic Algorithm (GA)

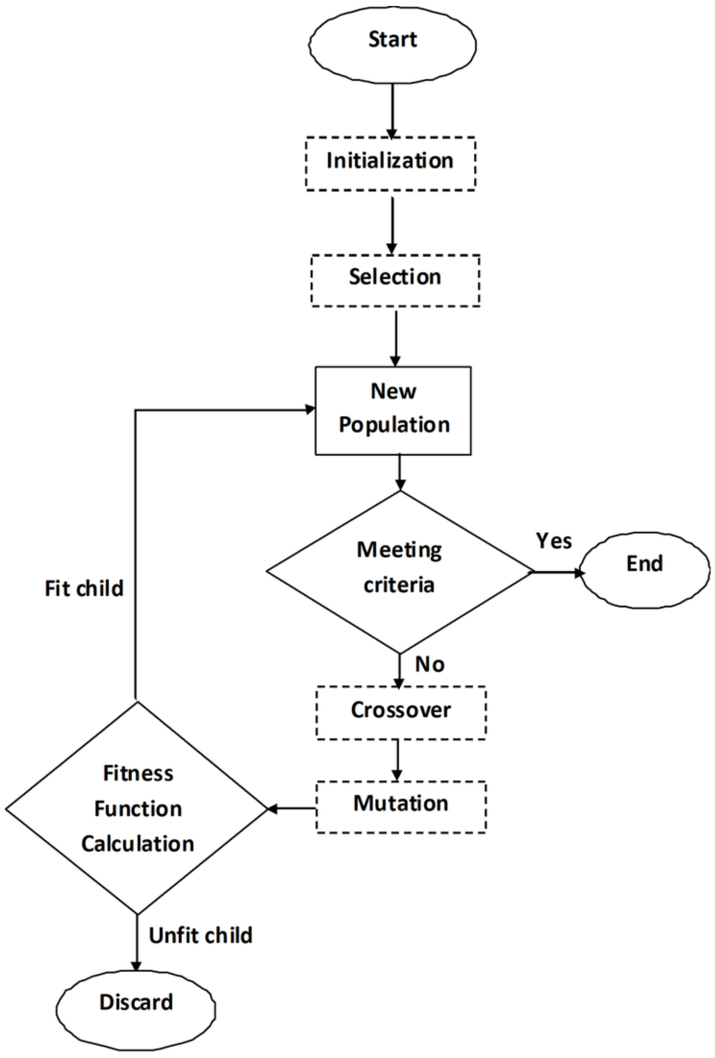
The genetic algorithm is a method generally used for solving constrained and unconstrained optimization problems. This method is based on the process of natural selection drives from biological evolution. The genetic algorithm starts with some known solutions known as initial population and repeatedly modifies the solutions. Firstly, this algorithm selects individuals/chromosomes from the initial population and use them generate the children for the next generation, and we get an optimal solution in successive generations

In each step it follows three basic rules to get the next generation from the current population:

- **Selection:** This rule selects the individuals commonly known as parents, used to create the population for the next generation.
- **Crossover:** This rule combines two parents to produce children for the next generation.
- **Mutation:** This rule pertain arbitrary changes to individual parents needed to form children.

We have shown the steps of a simple GA in the flowchart in Figure 5. The proposed algorithm differs from the traditional GA in terms of generation of initial population and mutation. This algorithm is a modified version of GALBC.

Figure 5. Flow Chart



3.4. Proposed Algorithm

Cluster formation is very important and improper design may cause overload. Thus, overloaded CHs increases latency in communication and consumes more energy in turn minimizes the performance of the network. The proposed algorithm balances the load by considering the residual energy. We have used the same method and design as discussed GALBC, but we have added residual energy as a constraint for load balancing.

3.4.1. Gene and Chromosomes

A chromosome is a collection of genes for example the string of CHs (Chakraborty et al., 2011). It is an order pair of Gene Indexes and Gene values. For example, if K_{ij} denotes a chromosome, then it implies that a sensor node s_i is assigned to CH_j. A chromosome is defined as an integer string of size m and each chromosome encodes an arrangement of CHs. If we have m number of genes then there is m! number of arrangements. Table 1 shows a chromosome containing 5 genes, the Gene Indexes are shown in First Row and Gene value is shown in Second Row.

3.4.2. Initialization

An initial population is generated from a randomly set of chromosomes which is a string of CHs. This step is similar to GABCL algorithm. We have used the same strategy for the generation of initial population because GALBC algorithm converges faster than the traditional GA. We have shown the initial population generation method in Figure 6. The method randomizes the genes of the normal and model chromosomes.

3.4.3. Selection

The selection process determines how we select chromosomes for reproduction on the basis of fitness value. In this step, we used roulette wheel sampling for selecting the chromosomes on the basis of fitness values. In this method a fixed point is selected on the wheel circumference when the wheel is rotated. The probability of choosing an individual chromosome using roulette wheel sampling method directly depends on its fitness value. The Select() method is shown in Figure 7.

3.4.4. Crossover

In this operation, more than one parent chromosomes are selected and one or more off-springs are produced using several methods such as one point, two point, and uniform crossover. In one-point crossover, we select a random crossover point and the tails of the two parents are swapped for new off-springs.

Table 1. An analogy to represent a string of CHs in terms of Gene and Chromosome

0	1	2	3	4	→ Gene Indexes
2	4	7	5	3	→ Gene Value

Figure 6. Initialize_Population() method

```
                    /* Initial population generation*/

void Initialize_Population(void){
  int chromosome;
  int gene;
  /* initialize normal chromosomes */
for(chromosome=0;chromosome<CHROMOSOME_COUNT;
++chromosome){
    for(gene=0; gene<GENE_COUNT; ++gene){
      currentGeneration[chromosome][gene] = rand()%TYPE;
    } }
  /* initialize model chromosome */
  for(gene=0; gene<GENE_COUNT; ++gene){
    modelChromosome[gene] = rand()%TYPE;
  }}
```

Figure 7. Select () method

```
                        /* Selection */
int Select(void){
  int chromosome;
  int runningTotal;
  int randomSelectPoint;
  runningTotal = 0;
  randomSelectPoint = rand() % (totalOfFitnesses + 1);
/*  Select  a chromosome */

for(chromosome=0;chromosome<CHROMOSOME_COUNT;
++chromosome){
    runningTotal += chromosomesFitnesses[chromosome];
    if(runningTotal >= randomSelectPoint)
 return chromosome; /*  return chromosome */
  }}
```

The process of crossover is given below:

Chromosome X: 1 0 0 1 0 0 1 1
Chromosome Y: 1 0 1 1 0 1 0 1

If we want crossover from the point || (one point crossover):

Chromosome A: 1 0 0 1 || 0 0 1 1
Chromosome B: 1 0 1 1 || 0 1 0 1

After crossover we get the following chromosomes:

New Chromosome A: 1 0 0 1 0 1 0 1
New Chromosome B: 1 0 1 1 0 0 1 1

3.4.5. Mutation

Mutation is a small random tweak at a selected gene in the chromosome, to obtain a new solution. In this step we have used the same technique as GALBC, but we have added additional residual energy as a constraint for load balancing. We select the cluster head from the chromosome having maximum optimum load computed using the given equation:

$$OL_i = L_i - E_i \tag{5}$$

In this process, we select a gene randomly and replace its CH on the basis of OL using equation 5. The crossover and mutation is presented using Re_produce() method shown in Figure 8.

3.4.6. Fitness Function

Evolutionary algorithms need a metric function in order to compare and evaluate the fittest candidate. We have modified the GALBC fitness function by adding the residual energy constraint. Our fitness

Figure 8. Re_produce() method

```
                    /* Crossover and Mutation*/

void Re_Produce(void){
  /* one point crossover */
  for(chromosome=0;       chromosome<CHROMOSOME_COUNT;
++chromosome){
    parent1 = Select();
    parent2 = Select();
    one_point_ crossover =  rand() % GENE_COUNT;
    for(gene=0; gene<GENE_COUNT; ++gene){
      /* copy over a single gene*/
      doMutation = rand() % (int) mutation_pace();
      if(doMutation == 0){
/* do mutation for the selected gene */
nextGeneration[chromosome][gene]= rand()%chromosomeType() ;
      } else {
        /* copy gene from  the parent */
        if (gene < crossoverPoint){
          nextGeneration[chromosome][gene] =
currentGeneration[parent1][gene];
        } else {
          nextGeneration[chromosome][gene] =
currentGeneration[parent2][gene];
        }}}}
```

function is constructed on the basis of the standard deviation (s) of the cluster head load and residual energy. Let us consider a WSN in which we have m number of CHs and n number of sensor node. Then the fitness function is given by:

$$\sigma = \sqrt{\frac{\sum_{j=1}^{m}\left(A - OL_j\right)^2}{m}} \tag{6}$$

where:

- A denotes average load calculated using the given function:

$$A = \sum_{i=1}^{n} d_i \,/\, m \tag{7}$$

- d_i denotes load of the sensor node s_i;
- OL_j is the optimum load of the CH_j.

The fitness function implies that if standard deviation is lower, it indicates a higher fitness value. The EvaluateChromosome() method is shown in Figure 9.

Figure 9. EvaluateChromosome() method

```
              /* Evaluate Chromosome Fitness */

int EvaluateChromosome(void){
  totalOfFitnesses = 0;

for(chromosome=0;chromosome<CHROMOSOME_CO
UNT; ++chromosome){
      /* Check chromosome's fitness */
   for(gene=0; gene<GENE_COUNT; ++gene){
     if( currentGeneration[chromosome][gene]
       == modelChromosome[gene] ){
      NextChromosome();
     }   }
   /*Calculate  total Fitness */
   chromosomesFitnesses[chromosome] =
     currentChromosomeFitness();
   totalOfFitnesses += currentChromosomeFitness();
   /* check for perfect generation */
   if(currentChromosomeFitness == fitness_point() ){
     return TRUE;
   }
  }  return FALSE;
}
```

4. SIMULATION RESULTS

We have conducted grievous simulations in our indigenous tool developed in C programming language. We have taken the result of GALBC and compared the result set with our proposed method. The result shows that our proposed method is better than GALBC in terms of energy consumption, number of active sensor nodes and sensor life.

4.1. Simulation Run

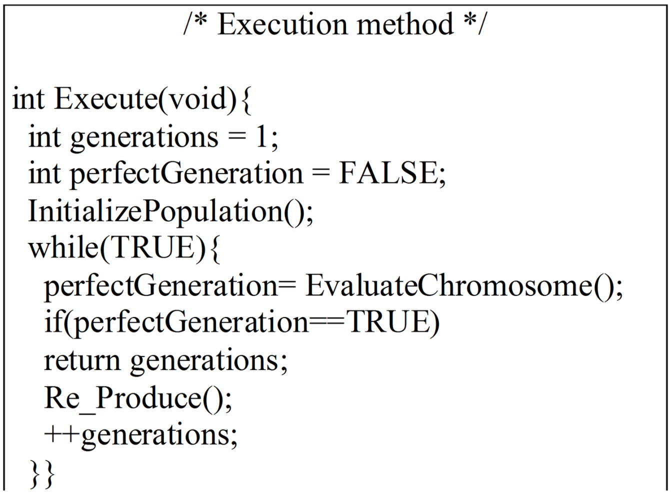
Our indigenous tool starts the simulation using the Execute() method shown in Figure 10. This method first initialize the generation and then call the InitializePopulation() method to get the initial population. Then it call the EvaluateChromosome() method to get the perfect generation. If perfect generation manages the load as per OL value then it stops otherwise it reproduce new generation by calling the Re_Produce() function.

The output of our tool is shown in Table 5. We have taken a WSN of 12 sensor nodes and 4 CH, i.e., S{s_1, s_2,…, s_{10}} and CH{CH_1, CH_2, CH_3, CH_4}. After that the select () method generates an individual from the initial population shown in Table 2. In this work we have calculated the optimum load (OL) shown in Table 3 using equation 5. Then a gene is selected on the basis of minimum OL shown in Table 3. GALBC replaces sensor node gateway on the basis of maximum load which is CH_3 or CH_4 in this example. Thus, we can see that it is not a perfect choice because of limited residual energy. But, our algorithm replaces a CH on the basis of OL and it is a good choice because of the additional energy constraint.

A positive value indicates that CH_4 has less energy.

Table 4 shows that GALBC algorithm replaces sensor nodes 11 and 12 to either 3 or 4 as per maximum load, but replacing CH on the basis of maximum load is not practical.

Figure 10. Execute() method



```
                /* Execution method */

int Execute(void){
  int generations = 1;
  int perfectGeneration = FALSE;
  InitializePopulation();
  while(TRUE){
    perfectGeneration= EvaluateChromosome();
    if(perfectGeneration==TRUE)
    return generations;
    Re_Produce();
    ++generations;
  }}
```

Table 2. Chromosome representation

S	1	2	3	4	5	6	7	8	9	10	11	12
CH	1	2	1	2	2	1	3	4	1	4	1	1

Table 3. Calculation of optimum load (OL)

CH	1	2	3	4
L	6	3	1	2
E	10	10	1	2
OL	-4	-7	0	0

Table 4. GALBC Output

S	1	2	3	4	5	6	7	8	9	10	11	12
CH	1	2	1	2	2	1	3	4	1	4	3 or 4	3 or 4

Table 5 shows that MGALBC algorithm replaces sensor nodes 11 and 12 to CH 2 on the basis of OL. Thus, our proposed algorithm is better than GALBC, because it replaces the correct node on the basis of OL.

Table 5. MGALBC Output

S	1	2	3	4	5	6	7	8	9	10	11	12
CH	1	2	1	2	2	1	3	4	1	4	2	2

Table 6. GALBC Polygon

Frequency (GALBC)	
Class	**Count**
5-54	0
55-104	20
105-154	16
155-204	19
205-254	15
255-304	13
305-354	5
355-404	4
405-454	7
455-504	1
505-554	0
Polygon 2	
Lowest Score	55
Highest Score	465
Total Number of Scores	100
Number of Distinct Scores	85

4.2. Comparison

We have generated frequency polygons representing life distributions of sensor nodes, associated frequency tables, and class distributions. These data relate life of sensor nodes. Table 6 and Table 7 shows associated frequency tables, class distributions and additional information of 100 sensor nodes. We have gathered these data distribution using GALBC and MGALBC algorithm. The histogram shows that we have achieved better result than GALBC in terms of sensor life in turn network life.

Figure 11 shows the histogram representing the frequency distribution of 100 sensor nodes created using Social Science Statistics tool (Social Science Statistics, 2017).

Figure 12 and Figure 13 shows the Comparison of the proposed method with GA (Hussain et al., 2007) and GALBC (Kulia et al., 2013) in terms of active sensor nodes per round and energy consumption per round. Our proposed algorithm has more active sensors as well as consumes less energy than GALBC.

Figure 14 and Figure 15 shows the comparison of the proposed method with GA (Hussain et al., 2007) and GALBC (Kulia et al., 2013) in terms of minimum execution time and load of the sensor nodes. Our proposed algorithm takes minimum execution time compared to GA and GALBC as well as manages load better than GA and GALBC.

5. CONCLUSION

In this paper, we have modified the GA based load balanced clustering algorithm for WSN (GALBC). In GALBC, authors have used GA for minimizing the maximum load of each gateway.

Table 7. MGALBC Polygon

Frequency (MGALBC)	
Class	**Count**
5-54	0
55-104	1
105-154	21
155-204	19
205-254	21
255-304	10
305-354	12
355-404	6
405-454	5
455-504	5
505-554	0
Polygon 1	
Lowest Score	101
Highest Score	500
Total Number of Scores	100
Number of Distinct Scores	85
Lowest Class Value	5
Highest Class Value	554
Number of Classes	11
Class Range	50

GALBC converges faster than the traditional GA but, it has a serious drawback. This method balances the load of the gateways without considering their residual energy. GALBC is not a practical method because it forcibly balances the load and in turn chooses the incorrect node which may create network failure. Our proposed algorithm is better than GALBC because it balances the load by considering the residual energy. We have used the same method and design as discussed in GALBC but we have added residual energy as a constraints load balancing. We have shown that GALBC algorithm replaces a CH on the basis of maximum load, but our algorithm replaces a CH on the basis of optimum load (OL). We have conducted grievous simulation on 100 sensor nodes and gathered data using both algorithms. The result shows that our proposed method is better than GALBC in terms of energy consumption, number of active sensor nodes, and network life.

Figure 11. Frequency distribution of 100 sensor nodes

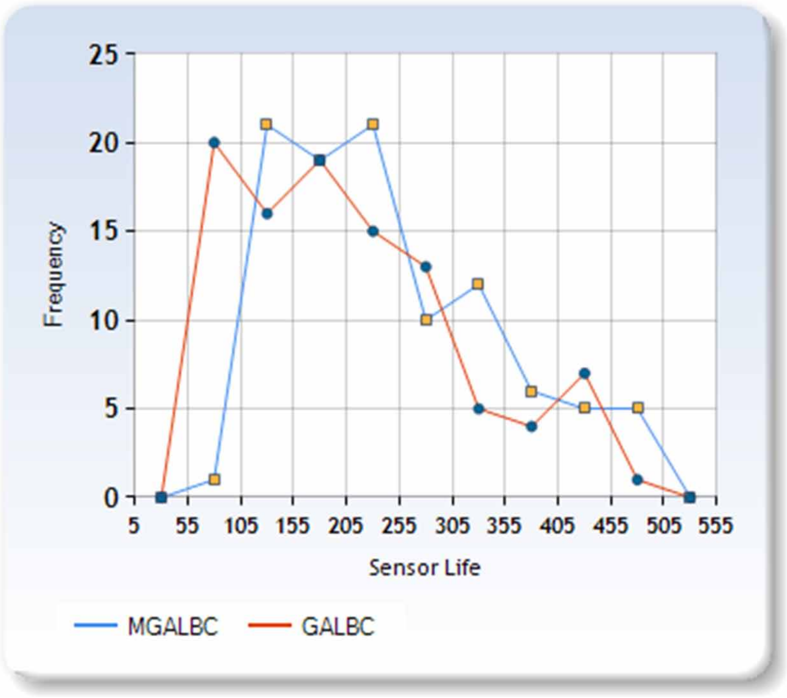


Figure 12. Active sensor nodes per round

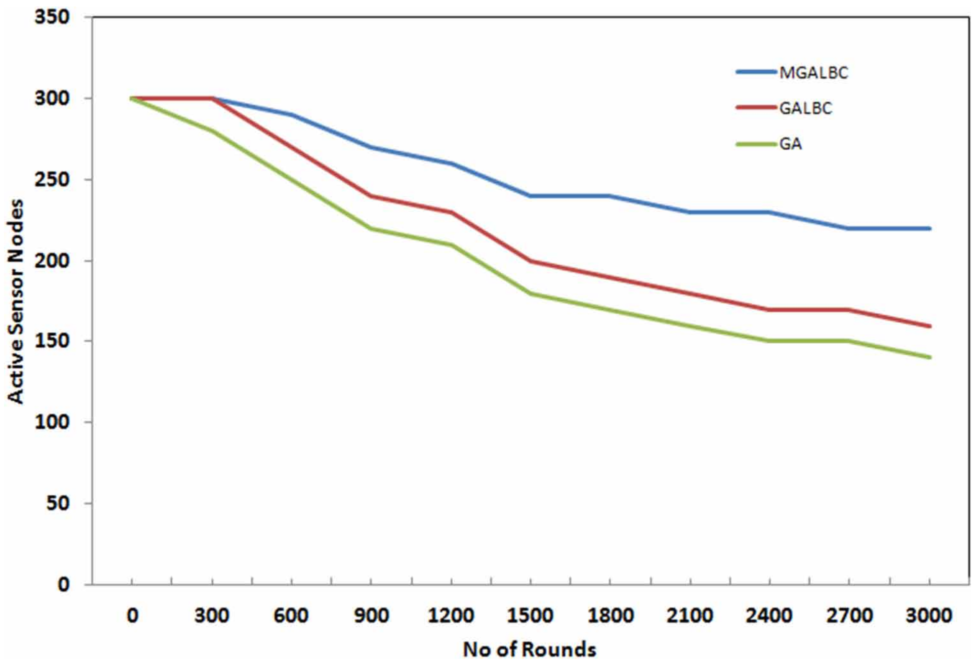


Figure 13. Energy consumption per round

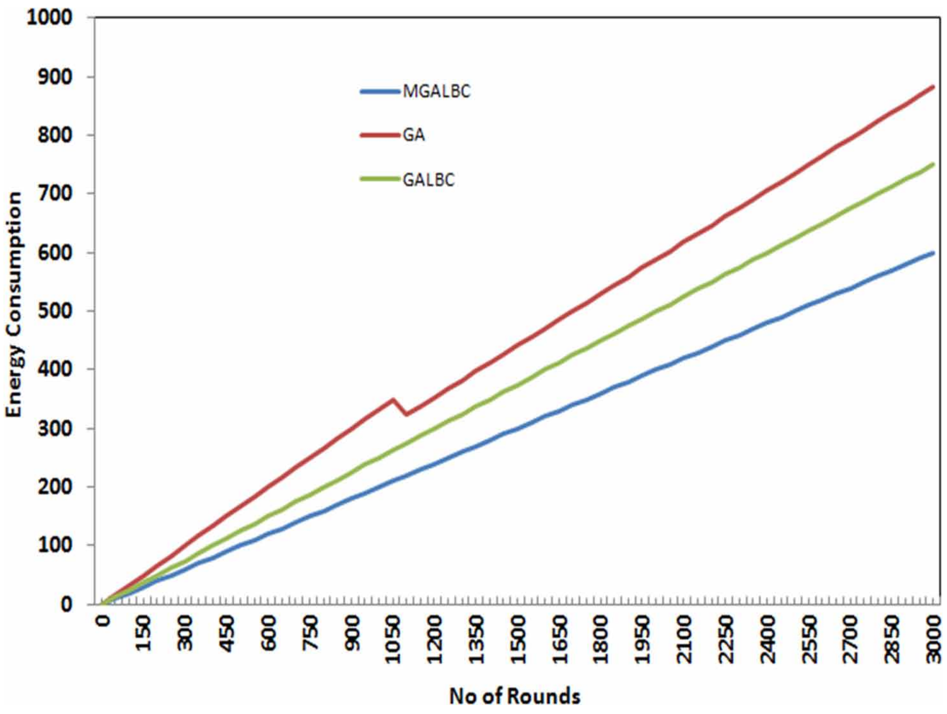


Figure 14. Execution Time

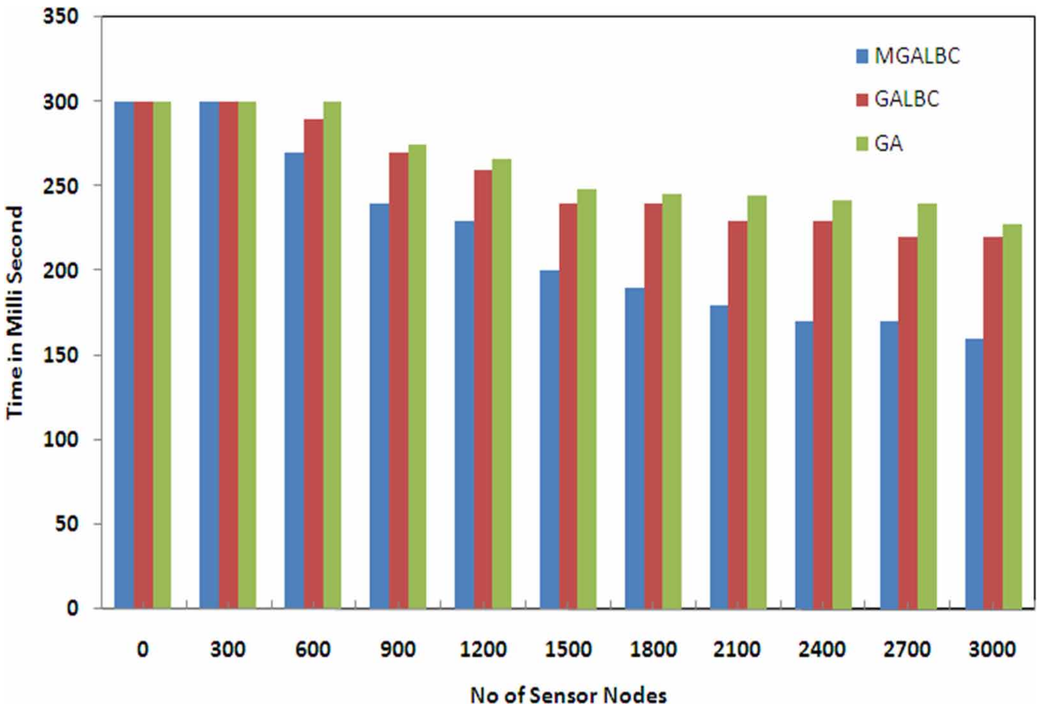
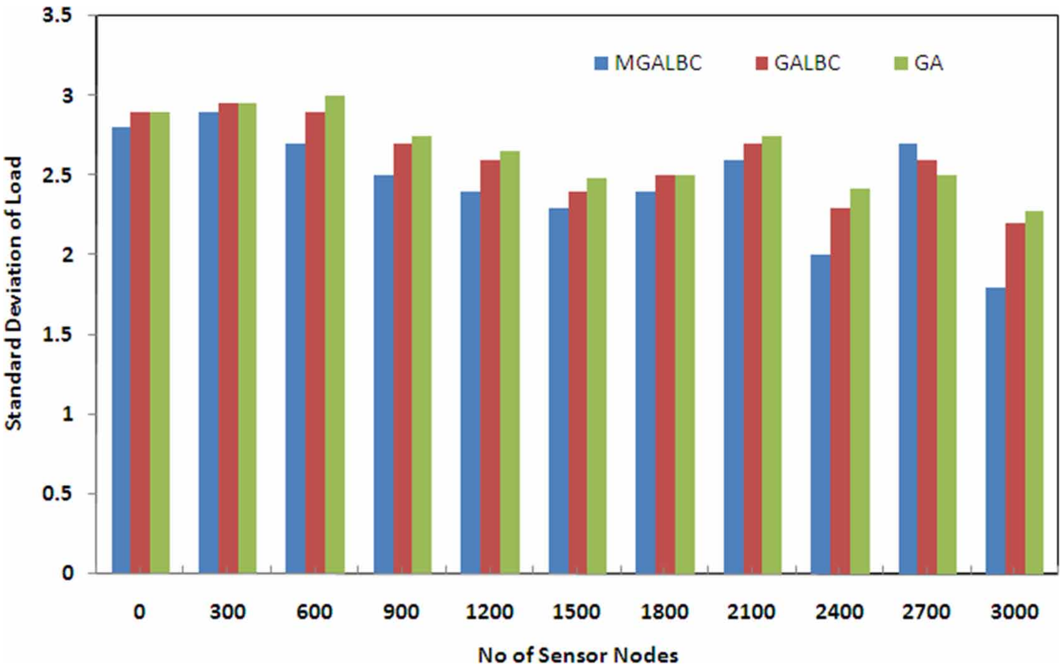


Figure 15. Standard Deviation of Equal Load



REFERENCES

Ali, M. S., Dey, T., & Biswas, R. (2008). Aleach: advanced leach routing protocol for wireless micro sensor networks. *International Conference on Electrical and Computer Engineering, (2008) ICECE 2008.*

Al-Refai, H., Al-Awneh, A., Batiha, K., Ali, A. A., & Rahman, Y. M. E. (2011). Efficient Routing Leach (Er-Leach) Enhanced On Leach Protocol In Wireless Sensor Networks. *International Journal of Academic Research*, *3*(3).

Chakraborty, A., Mitra, S. K., & Naskar, M. K. (2011). A genetic algorithm inspired routing protocol for wireless sensor networks. *International Journal of Computational Intelligence Theory and Practice*, *6*(1), 1–8.

Gupta, S. K., Kuila, P., & Jana, P. K. (2017). Energy efficient routing algorithm for wireless sensor networks: A distributed approach. In *Communication and Computing Systems: Proceedings of the International Conference on Communication and Computing Systems (ICCCS 2016)*, *Gurgaon, India*, *9-11* (p. 207). CRC Press.

Han, R., Yang, W., Wang, Y., & You, K. (2017). DCE: A Distributed Energy-Efficient Clustering Protocol for Wireless Sensor Network Based on Double-Phase Cluster-Head Election. *Sensors (Basel)*, *17*(5), 998. doi:10.3390/s17050998

Heinzelman, W. B. (2000). *Application-specific protocol architectures for wireless networks* (Doctoral dissertation). Massachusetts Institute of Technology.

Huruială, P. C., Urzică, A., & Gheorghe, L. (2010). Hierarchical routing protocol based on evolutionary algorithms for wireless sensor networks. In *Roedunet International Conference (RoEduNet), 2010 9th* (pp. 387-392). IEEE.

Hussain, S., Matin, A. W., & Islam, O. (2007). Genetic algorithm for hierarchical wireless sensor networks. *JNW*, *2*(5), 87–97. doi:10.4304/jnw.2.5.87-97

Kong, L., Pan, J. S., Snášel, V., Tsai, P. W., & Sung, T. W. (2017). An energy-aware routing protocol for wireless sensor network based on genetic algorithm. *Telecommunication Systems*, 1–13.

Kuila, P., Gupta, S. K., & Jana, P. K. (2013). A novel evolutionary approach for load balanced clustering problem for wireless sensor networks. *Swarm and Evolutionary Computation*, *12*, 48–56. doi:10.1016/j.swevo.2013.04.002

Kuila, P., & Jana, P. K. (2012). An energy balanced distributed clustering and routing algorithm for wireless sensor networks. PDGC2012, 220–225. doi:10.1109/PDGC.2012.6449821

Kumar, G., & Rai, M. K. (2017). An energy efficient and optimized load balanced localization method using CDS with one-hop neighborhood and genetic algorithm in WSNs. *Journal of Network and Computer Applications*, *78*, 73–82. doi:10.1016/j.jnca.2016.11.013

Liu, Y., Gao, J., Jia, Y., & Zhu, L. (2008). A cluster maintenance algorithm based on leach-dchs protocol. *International Conference on Networking, Architecture, and Storage*, 165–166. doi:10.1109/NAS.2008.59

Liu, X. (2012). A survey on clustering routing protocols in wireless sensor networks. *Sensors, 12*(8), 11113-11153.

Mehr, M. A. (2011). Design and implementation a new energy efficient clustering algorithm using genetic algorithm for wireless sensor networks. *World Academy of Science, Engineering and Technology*, *52*, 430–433.

Nayak, P., & Vathasavai, B. (2017). Energy Efficient Clustering Algorithm for Multi-Hop Wireless Sensor Network Using Type-2 Fuzzy Logic. *IEEE Sensors Journal*, *17*(14), 4492–4499. doi:10.1109/JSEN.2017.2711432

Social Science Statistics. (2017). https://www.socscistatistics.com/descriptive/polygon/Default.aspx

Tyagi, S., & Kumar, N. (2013). A systematic review on clustering and routing techniques based upon LEACH protocol for wireless sensor networks. *Journal of Network and Computer Applications*, *36*(2), 623–645. doi:10.1016/j.jnca.2012.12.001

Yuan, X., Elhoseny, M., El-Minir, H. K., & Riad, A. M. (2017). A genetic algorithm-based, dynamic clustering method towards improved WSN longevity. *Journal of Network and Systems Management*, *25*(1), 21–46. doi:10.1007/s10922-016-9379-7

Mohit Kumar is an Assistant Professor in the Department of Computer Science & Engineering, Cambridge Institute of Technology, Tatisilwai, Ranchi, Jharkhand, India. His research interest includes wireless sensor networks, natural language processing, deep learning and feature engineering. He is the author of over 08 peer-reviewed publications. He is pursuing Ph.D from Jaipur National University. He received his M. Tech degree from Jaypee University, Solan (H.P) and B.E degree from Pune University.

Dinesh Kumar is an Assistant Professor in the Department of Computer Science, Jaipur National University, India.

Mohammad Amir Khusru Akhtar is an Associate Professor in the Faculty of Computing and Information Technology, Usha Martin University, Ranchi, Jharkhand, India. His research interest includes mobile ad-hoc network, natural language processing, deep learning and feature engineering. He is the author of over 20 peer-reviewed publications. He received his Ph.D from the Department of Computer Science & Engineering at Birla Institute of Technology, Mesra, Ranchi, India in 2015. He received his M. Tech degree from the Department of Computer Science & Engineering at Birla Institute of Technology, Mesra, Ranchi, India in 2009.