Journal of Engineering and Applied Sciences 13 (11): 3927-3934, 2018
ISSN: 1816-949X

Particle Swarm Optimization Based Approach for Reduction of Energy Consumption and Time Period in Wireless Sensor Network

[1]R. Sivaranjani and [2]A. Senthil Kumar
[1]Department of Computer Technology,
[2]Department of Computer Application, Hindusthan College of Arts and Science, Coimbatore, India

Abstract: During the previous decades, Wireless Sensor Networks (WSN) got ton of attraction from the scientific and industrial society. WSNs are composed of giant range of tiny resource affected devices recognized as sensors. Energy could be a important issue in WSN. Energy efficient cluster is an eminent improvement downside that has been studied extensively to prolong the lifespan of the network. This study demonstrates the programming formulation of this downside followed by a projected algorithmic program with Cat Swarm Optimisation (CSO) approach. The clustering technique is expressed by taking into thought of energy saving of nodes. The projected algorithmic program is experimented wide and results are evaluated with existing ways to point out their mastery in term of energetic nodes, energy out flow, packet delivery ratio and output of network. Simulation results shows that our projected algorithm outstrip the other existing algorithms of its class.

Key words: Energy based clustering, cat swarm optimization, packet delivery ratio, energy outflow, wireless sensor networks, class

INTRODUCTION

Recent advances in method technologies and also, the shrinking sizes of radio communication devices and sensors allowed researchers to mix three operations (i.e., sensing, communication and computation) into small devices known as wireless sensor nodes. Once these devices are scattered through the atmosphere, they'll simply construct data-oriented networks called Wireless Sensor Networks (WSNs). Today, there are a huge variety of application eventualities involving WSNs in business, military, medical and science domains.

The time period, measurability, latent period and effective sampling frequency are among the foremost important parameters of WSNs and that they are closely associated with one crucial resource constraint that's terribly onerous to satisfy, the power consumption. The WSN nodes are designed to be battery operated, since, they'll be utilised in any quite surroundings together with thick biological science, volcanic mountains and ocean beds. Consequently, everything should be designed to be power-aware in these networks. Small-scale in operation systems, like TinyOS (Hill *et al*., 2000) ambient RT (Anonymous, 2013) and communication intensive applications considerably increase the energy consumption of the processor component of WSN nodes.

Today, new device platforms with 16 bit (Anonymous, 2013) and 32 bit processor architectures target additional and additional power greedy applications. By Conner *et al*. (2003), the researchers show that the processor itself dissipates 35th of the whole energy budget of the MICA2 platform whereas running surge, a TinyOS observation application. By Madden *et al*. (2005), the researcher claim similar energy values once running TinyDB question coverage light and accelerometer readings once each minute. By Polastre (2003), the researchers notice that the energy consumption of the processor/memory element for data compression is more than the energy consumption of data transmission. Similarly, nowadays most of the WSN applications avoid intensive computations and prefer to transfer data to server machines to extend the period of the device nodes.

On the contrary, our projected style encourages the WSN application developers to style less centralized applications by distributing the computation research and reducing the network traffic among the nodes.

Temporal and value locality: Device network applications have periodic behaviour. Especially, observance applications, like surge, might sense and research on constant knowledge and constant memory locations for

Corresponding Author: R. Sivaranjani, Department of Computer Technology, Hindusthan College of Arts and Science, Coimbatore, India

long durations. During this study, initial we have a tendency to show that, we tend to significantly scale back the energy dissipation of the WSN processors by caching unremarkably used knowledge during a small range of latches (MoteCache) (Kucuk and Basaran, 2006). Then, we tend to conjointly show that almost all of the shop instructions in WSN applications are silent (these directions write values that precisely match the values that are already hold on at the memory address that's being written) and propose to filter them by extending the MoteCache design.

Common data values: We discover that there are some common knowledge values flowing through the data path. Within the second a part of this study, we have a tendency to propose a way known as Content-Aware knowledge Management (CADMA) that exploits this behaviour and boosts our energy savings by reducing the read/write energy not solely in SRAM however, conjointly in register file and therefore the Mote-cache.

MATERIALS AND METHODS

Greedy algorithm based energy efficient routing algorithm: Since, the nodes of wireless sensor networks are within the condition of a highly-limited and un-replenished energy resource corresponding to battery power, computation and storage space, the energy potency is that the most significant key purpose of the network routing planning. In this study, a completely unique routing algorithmic program which mixes with hierarchic routing and geographical routing is planned. Supported the hierarchic specification, the method of forwarding packets between the supply nodes within the target region and also, the base station consists of two phases inter cluster routing and intra-cluster routing, a greedy algorithmic program is adopted within the method of the inter-cluster routing and an multi-hop routing formula supported the forwarding restriction angle is meant for the intra cluster routing. The analysis and simulation results show that our routing algorithmic program has higher performance in terms of energy consumption and delay, it's appropriate for the information transmission in a very high-density wireless sensing element network.

Routing algorithm design: In our algorithmic program, the sensing element nodes within the sensing element network organize themselves into local clusters with one node acting because the local base station or cluster head. The precise cluster algorithmic rule is analogous to LEACH, here, it isn't mentioned very well. The functions of cluster-head embrace not solely the overall functions, corresponding to information acquisition and forwarding, however, conjointly the local information fusion to "compress" the quantity of information being sent from the clusters to the bottom station, additional reducing energy dissipation and enhancing system time period. The method of forwarding packets between the supply nodes within the target region and also the base station consists of two phases.

Inter-cluster routing: The cluster-head of the target region communicates with the bottom station directly or solely through alternative cluster-heads.

Intra-cluster routing: The supply nodes within the target cluster communicate with its cluster-head directly or solely through alternative supply nodes.

Genetic algorithm: Also, referred to as a worldwide heuristic algorithmic rule, a generic algorithmic rule estimates an optimum resolution through generating totally different individuals (Goldberg, 1989). Targeted fitness function is one among procedures of the algorithmic rule. Following study describes the basic components of a generic algorithmic rule. Figure 1-3 indicates the overall theme of genetic algorithmic program mechanism.

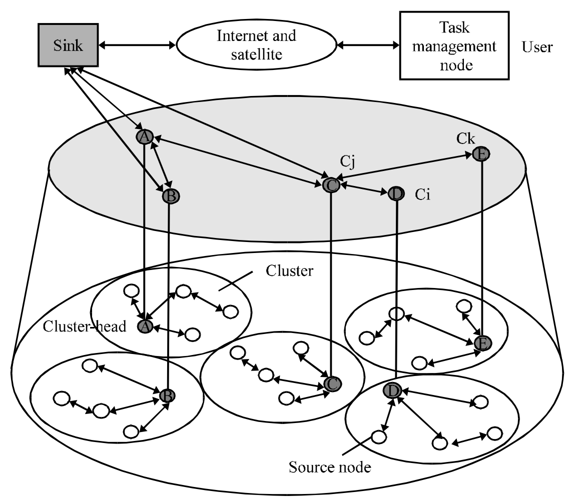

Fig. 1: Inter cluster routing

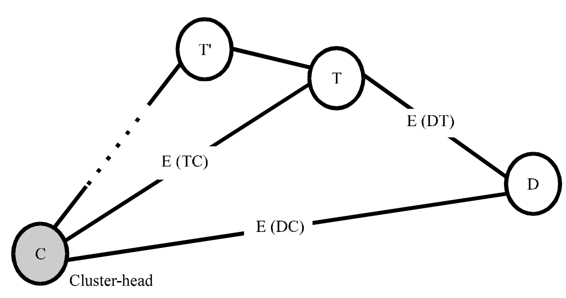

Fig. 2: Intra cluster data transmitting

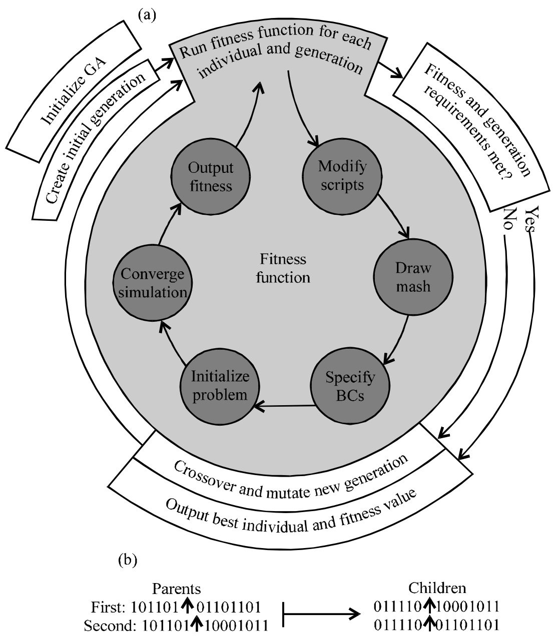

Fig. 3: a) General scheme of GA mechanism and b) Single point method at random point 6

Initialization: The genetic algorithm starts with an elementary population comprised of random chromosomes which includes genes with a sequence of 0 or 1 sec. Afterward, the algorithm leads individuals to achieve an optimum solution by the way of repetitive processes including crossover and selection operators.

There are two ways to develop a new population (Kreinovich *et al*., 1993), steady-state GA and generational GA. In the case of the former, one or two members in the population are replaced and at the same time, the generational GA replaces all the generated individuals of a generation.

Fitness: Under the genetic algorithm, the fitness function by definition is a process for scoring each chromosome based on their qualification. The assigned score is a trait for continuation of further reproduction. Dependence to problem by the fitness function is considerable, so that, in case of some problems, it is not possible to define the problem. Naturally, individuals are permitted to go to the new generation based on their fitness score. Therefore, the score dictates the fate of individuals.

Selection: During every successive generation, a new generation is developed through adopting members of the current generation to mate on the bases of their fitness. The individuals with higher fitness score have higher chance for being selected, the process which results in preferential adoption of the best solution. Majority of the functions include a stochastically designed element for adopting small number of less fit individuals for sake of keeping diversity in the population (Norouzi *et al*., 2011). Among the many selection methods, roulette-wheel is adopted to differentiate proper individuals with the probability of:

$$P_i = \frac{F_i}{\sum_{j=1}^{n} F_1}$$

where F_i and n are the fitness chromosome and the size of population, respectively. According to the roulette-wheel, each individual is assigned a value between 0 and 1.

Crossover: The crossover or reproduction process constitutes the major step toward production. In the reproduction process, crossover process adopts a couple of individuals as the parents through breeding selection process. The process continues to reach the desired size in the new population. Generally, several crossover operations take place, each of which with different aims. The easiest way is single point where a random point is adopted to divide the role of the patents. One example of mating by two chromosomes in single point way is pictured in Fig. 3a.

Figure 3b represents two children that are from a single set of parents. The bit sequence of the offspring duplicates one parent's bit sequence until the crossover point. Afterward, the bit sequence of the other parent is replicated as the second part of children.

Node sensor placement in wireless sensor network: The placement of sensor nodes on a monitored field may influence the general performance of the network. Taking into account the placement of nodes in the field, there are three main categories of placement of nodes in a network including the deterministic node placement (grid), the semi-deterministic node placement (e.g., biased random) and the nondeterministic (stochastic) node placement (e.g., simple diffusion and random). Long range transmission by sensor nodes is not energy efficient as it needs more energy than a linear function of transmission distance does. Clearly, node density is just one element in network topology as the placement of the node is another key factor. The placement of nodes influences the capacity of a network to correctly sense an event as well as the number of possible disjoint paths towards the sink (s).

Under the deterministic node placement, the nodes are placed on exact, preset points on a grid or in specific parts of the grid. Commonly, deterministic or controlled node placement dictates the type of nodes, the environment that nodes will be placed and the application.

Thus, in sensor indoor surveillance systems or building monitoring application nodes must be placed manually (Hussain *et al*., 2007).

Under semi-deterministic placement, on the other hand, individual nodes are positioned in a nondeterministic way on the grid (e.g., random) which covers the areas nodes must be spread. That is microscopic and macroscopic ways of placement of nodes are nondeterministic and deterministic, respectively.

To make sure that network runs with the highest feasible performance, the nodes are positioned on the campus network.

Along with balanced energy consumption of all nodes, a preferred node placement protocol is supposed to supply a better network through put through attenuating contention of channel and collision of packet under high load. An instance of a node placement scheme is pictured in Fig. 4 and 5.

The common benefits of correct sensing element propagation in WSNs are listed as (Sergiou and Vassiliou, 2012).

Scalability: A high variety of nodes may be deployed within the network, this can be appropriate once transmissions between the nodes don't seem to be unlimited.

Collision reduction: Since, the Cluster Head (CH) functions as an organiser, a restricted range of nodes gain access to the channel and cluster members and head communicate regionally.

Energy potency: High energy consumption may be a consequence of the periodic relocation. Still, duties of CH are also distributed among all alternative nodes through periodic relocation which ends up in lower energy consumption.

Low cost: The surplus prices may be avoided by deploying sensors at correct locations.

Routing backbone: The info collected by cluster members is aggregative in CH and sent to the sink. Thus, employing a very little route-thru traffic and routing backbone with enough potency one will build the network.

Novel cat swarm optimization energy efficient clustering algorithm: Optimization algorithms supporting the Swarm Intelligence (SI) were urbanized in order to simulate the intellectual performance of animals. In these types of modelling systems, a inhabitants of organisms similar to ants, birds, bees and fish interact with each other and with

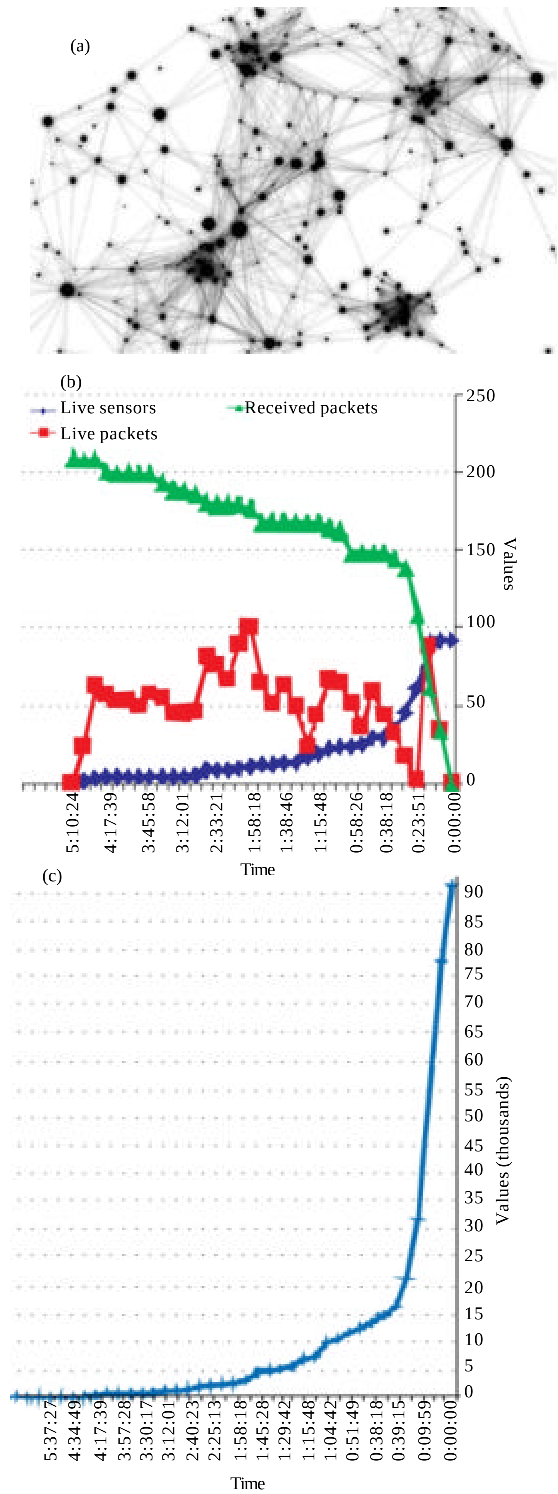

Fig. 4: a) Node placement scheme; b) Comparison between number of available sensors, live and received packets existing in the network and c) Comparison between mount of power and lifetime of network

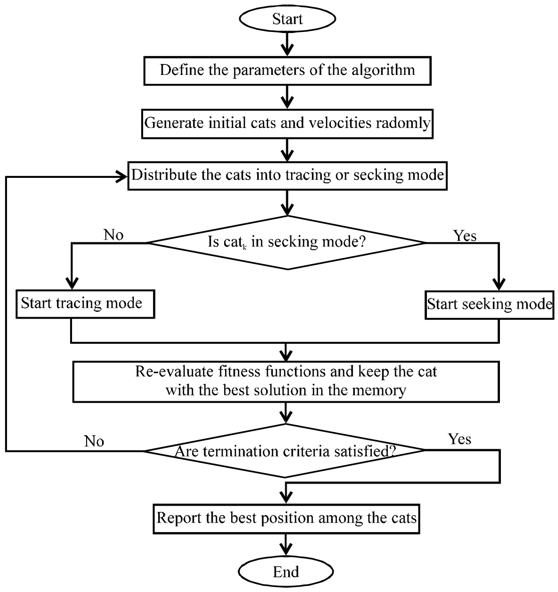

Fig. 5: Computational steps of CSO algorithm

their atmosphere in the course of sharing data, resulting in the use of their surroundings and resources. One amongst the newer SI-based optimisation rules is that the Cat Swarm Optimisation (CSO) algorithm that is predicated on the behaviour of cats. Developed by Chu and Tsai, the CSO rule and its varieties are enforced for various optimisation issues.

Process of cat swarm optimization algorithm: Despite defrayment most of their time in resting, cats has high alertness and curiosity concerning their surroundings and moving objects in their atmosphere. This behaviour helps cats find preys and searching them down. In CSO, a population of cats are created and indiscriminately distributed within the m-dimensional solution area with every cat representing a solution. This population is split into two subgroups. The cats within the 1st subgroup are resting and keeping an eye fixed on their surroundings (i.e., seeking mode) whereas the cats within the second subgroup begin on the move and chasing their preys (i.e., tracing mode). The mixture of those two modes helps CSO to manoeuvre toward the world resolution within the m-dimensional resolution area. Since, the cats pay insufficient time within the tracing mode, the quantity of the cats within the tracing subgroup ought to be tiny. This variety is outlined by mistreatment the Mixture quantitative Relation (MR) that encompasses a tiny worth. When sorting the cats into these 2 modes, new positions and fitness functions are going to be offered, from that the cat with the most effective resolution are going to be saved within the memory. These steps are repeated till the stopping criteria are satisfied.

Table 1: Characteristics of cso algorithm

General algorithm	CSO algorithm
Decision variable	Cat's position in each dimension
Solution	Cat's position
Old solution	Old position of cat
New solution	New position of cat
Best solution	Any cat with the best fitness
Fitness function	Distance between cat and prey
Initial solution	Random positions of cats
Selection	-
Process of generating new solution	Seeking and tracing a prey

Computational procedure: The computational procedures of CSO may be represented as follows:

Step 1: Create the preliminary inhabitants of cats and scatter them into the m-dimensional resolution area (Xi, d) and arbitrarily allocate each cat a rate to be different from the most rate value (ti, d).

Step 2: In keeping with the value of mr, assign every cat a flag to kind them into the seeking or tracing mode method.

Step 3: Valuate the fitness worth of every cat and save the cat with the most effective fitness perform. The position of the most effective cat (Xbest) represents the most effective resolution so far.

Step 4: Apply the cats into the seeking or tracing mode method, supporting their flags as represented.

Step 5: If the termination criteria are satisfied, terminate the method. Otherwise repeat steps 2 through 5 (Table 1).

Seeking mode (resting): During this mode the cat is resting while keeping an eye on its environment. In case of sensing a prey or danger, the cat decides its next move. If the cat decides to move, it does that slowly and cautiously. Just like while resting in the seeking mode the cat observes into the m-dimensional solution space in order to decide its next move. In this situation, the cat is aware of its own situation, its environment and the choices it can make for its movement. These are represented in the CSO algorithm by using four parameters, Seeking Memory Pool (SMP), seeking range of the Selected Dimension (SRD), Counts of Dimension to Change (CDC) and Self-Position Consideration (SPC) (Chu and Tsai, 2007). SMP is the number of the SRD is the maximum difference between the new and old values in the dimension selected for mutation. CDC tells how many dimensions will be mutated. All these parameters define the seeking process of the algorithm. SPC is the Boolean variable which indicates the current position of the cat as a candidate position for movement. SPC cannot affect the value of SMP. Following by Chu and Tsai, the process of the seeking mode is described as.

Step 1: Make SMP copies of each cati. If the value of SPC is true, SMP-1 copies are made and the current position of the cat remains as one of the copies.

Step 2: For each copy, according to CDC calculate a new position by using the following equation:

$$X_{cn} = (1+SRDXRD)\ XC$$

In which X_c current position, X_{cn} new position and R a random number which varies between 0 and 1.

Step 3: Compute the Fitness values (FS) for new positions. If all FS values are exactly equal, set the selecting probability to 1 for all candidate points. Otherwise calculate the selecting probability of each candidate point by using Eq. 2:

$$P_i = \frac{|FS_i-FS_b|}{|FS_{max}-FS_{min}|} \text{ where } 0<i<j$$

Where:

P_i = Probability of current candidate cat_i
Fs_i = Fitness value of the cat_i
Fs_{max} = Maximum value of fitness function
Fs_{min} = Minimum value of fitness function
Fs_b = FS_{max} for minimization problems
Fs_b = FS_{min} for maximization problems

Step 4: Using the roulette wheel, randomly pick the point to move to from the candidate points and replace the position of cat_i.

Tracing mode (movement): The tracing mode simulates the cat chasing a prey. After finding a prey while resting (seeking mode), the cat decides its movement speed and direction based on (Kennedy and Eberhart, 1995) Bahrami *et al*. the prey's position and speed. In CSO, the velocity of cat k in dimension d is given by:

$$V_{k,\ d} = v_{k,\ d}+r_1\ Xc_1\ (X_{best,\ d}-X_{k,\ d})$$

Where:

$V_{k,d}$ = Velocity of cat k in dimension d
$X_{best,d}$ = Position of the cat with the best solution
$X_{k,d}$ = Position of the cat_k
c_1 = A constant
r_1 = A random value in the range of [0,1]

Using this velocity, the cat moves in the Mdimensional decision space and reports every new position it takes. If the velocity of the cat is greater than the maximum velocity, its velocity is set to the maximum velocity. The new position of each cat is calculated by:

$$X_{k,\ d,\ new} = X_{k,\ d,\ old}+v_{k,\ d}$$

Where:

$X_{k,\ d,\ new}$ = New position of cat k in dimension d
$X_{k,d,old}$ = Current position of cat k in dimension d

RESULTS AND DISCUSSION

Figure 6 and 7 show the routing model of wireless sensor network. The figure indicates sinkhole nodes,

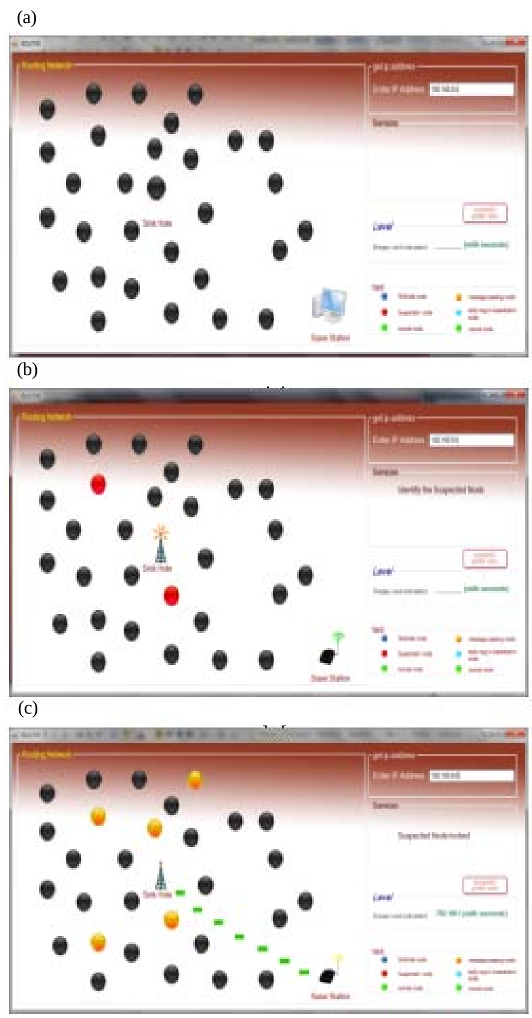

Fig. 6: Routing model for wireless sensor network: a) Routing model with initial nodes creation; b) Routing model with message passing nodes from source and c) Routing model with acknowledgement received node to source node

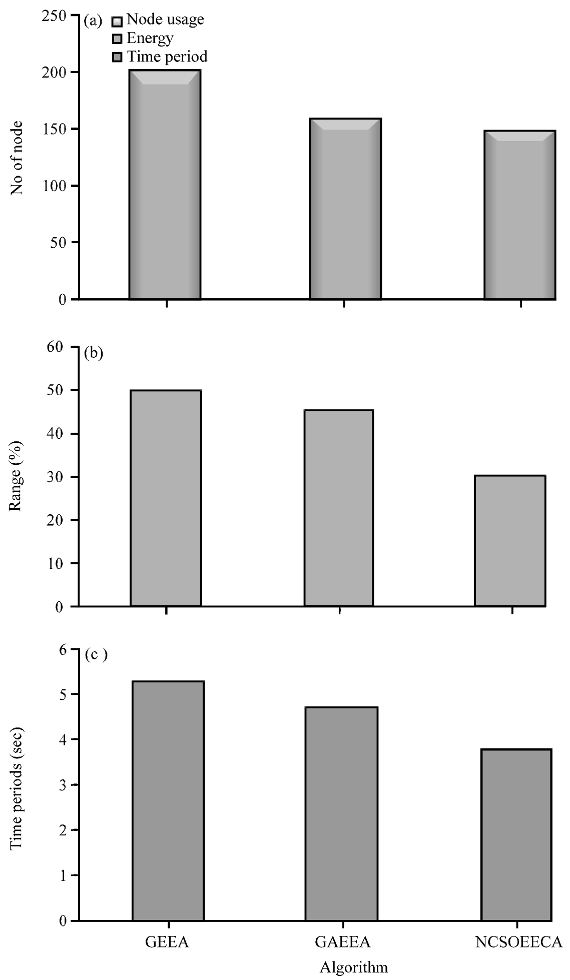

Fig. 7: a) Comparison on node usage; b) Comparison on energy usage and c) Comparison on Time period

Table 2: Energy used for nodes

Algorithm	Energy (%)	Node usage (500)
GEEA	50	200
GAEEA	45	160
NCSOEECA	30	145

Table 3: Time period

Algorithm	Time period	Node usage
GEEA	5.3	200
GAEEA	4.7	160
NCSOEECA	3.8	145

message passing nodes, normal nodes, suspected nodes, reply message to base station nodes and energy level.

Table 2 and 3 explains the energy usage and node usage of NCSOEECA algorithm by comparing with GEEA, GAEEA algorithms. The time period of the proposed algorithm is compared with other algorithm and is explained in Table 3. Figure 5b explains the comparison of energy usage and time period of the proposed technique with various algorithms.

CONCLUSION

The performance of the proposed system is evaluated on the basis of energy usage, nodal usage and time period consumption. The result of the proposed algorithm is evaluated by comparing the result with other existing algorithms such as GEEA and GAEEA. From the result, it is clear that the proposed algorithm gives a better result in node usage and consumes less energy and time period when compared to other mentioned existing algorithms.

REFERENCES

Anonymous, 2013. Tmote sky. https: // wireless sensor networks. weebly.com/blog/tmote-sky

Conner, W.S., J. Chhabra, M. Yarvis and L. Krishnamurthy, 2003. Experimental evaluation of synchronization and topology control for in-building sensor network applications. Proceedings of the 2nd ACM International Conference on Wireless Sensor Networks and Applications, September 19, 2003, ACM, San Diego, California, USA., ISBN:1-58113-764-8, pp: 38-49.

Goldberg, D.E., 1989. Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Boston, Massachusetts, USA., ISBN:9780201015768, Pages: 412.

Hill, J., R. Szewczyk, A. Woo, S. Hollar, D. Culler and K. Pister, 2000. System architecture directions for networked sensors. Proceedings of the 9th International Conference on Architectural Support for Programming Languages and Operating Systems, November 12-15, 2000, Cambridge, MA., USA., pp: 93-104.

Hussain, S., A.W, Matin and O. Islam, 2007. Genetic algorithm for hierarchical wireless sensor networks. J. Networks, 2: 87-97.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proc. IEEE Int. Conf. Neural Networks, 4: 1942-1948.

Kreinovich, V., C. Quintana and O. Fuentes, 1993. Genetic algorithms: What fitness scaling is optimal?. Cybern. Syst., 24: 9-26.

Kucuk, G. and C. Basaran, 2006. Reducing energy dissipation of wireless sensor processors using silent-store-filtering motecache. Proceedings of the 16th International Workshop on Power and Timing Modeling, Optimization and Simulation (PATMOS'06), September 13-15, 2006, Springer, Montpellier, France, ISBN:978-3-540-39094-7,-pp: 256.

Madden, S.R., M.J. Franklin, J.M. Hellerstein and W. Hong, 2005. TinyDB: An acquisitional query processing system for sensor networks. ACM. Trans. Database Syst. TODS., 30: 122173.

Norouzi, A., F.S. Babamir and A.H. Zaim, 2011. A Novel Energy Efficient Routing Protocol in Wireless Sensor Networks. Wireless Sens. Netw., 3: 341-350.

Polastre, J.R., 2003. Design and implementation of wireless sensor networks for habitat monitoring. Master's Thesis, Department of Electrical Engineering and Computer Sciences, University of California at Berkeley.

Sergiou, C. and V. Vassiliou, 2012. Efficient Node Placement for Congestion Control in Wireless Sensor Networks. In: Wireless Sensor Networks-Technology and Applications, Sergiou, C. and V. Vassiliou (Eds.). InTech, Rijeka, Croatia, pp: 1-21.