Data Aggregation Tree Structure in Wireless Sensor Networks Using Cuckoo Optimization Algorithm

Elham Mohsenifard
Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
mohsenifards@gmail.com
Ali Ghaffari*
Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
a.ghaffari@iaut.ac.ir

Received: 28/May/2016 Revised: 25/Jul/2016 Accepted: 02/Aug/2016

Abstract

Wireless sensor networks (WSNs) consist of numerous tiny sensors which can be regarded as a robust tool for collecting and aggregating data in different data environments. The energy of these small sensors is supplied by a battery with limited power which cannot be recharged. Certain approaches are needed so that the power of the sensors can be efficiently and optimally utilized. One of the notable approaches for reducing energy consumption in WSNs is to decrease the number of packets to be transmitted in the network. Using data aggregation method, the mass of data which should be transmitted can be remarkably reduced. One of the related methods in this approach is the data aggregation tree. However, it should be noted that finding the optimization tree for data aggregation in networks with one working-station is an NP-Hard problem. In this paper, using cuckoo optimization algorithm (COA), a data aggregation tree was proposed which can optimize energy consumption in the network. The proposed method in this study was compared with genetic algorithm (GA), Power Efficient Data gathering and Aggregation Protocol- Power Aware (PEDAPPA) and energy efficient spanning tree (EESR). The results of simulations which were conducted in matlab indicated that the proposed method had better performance than GA, PEDAPPA and EESR algorithm in terms of energy consumption. Consequently, the proposed method was able to enhance network lifetime.

Keywords: Wireless Sensor Networks (WSNs); Data Aggregation Technique; Data Aggregation Tree; Cuckoo Optimization Algorithm (COA); Network Lifetime Enhancement.

1. Introduction

WSNs are resource constrained networks and include many limited-energy sensor nodes. In WSN, each sensor can sense specific data and transmit it to its neighbors [1]. In general, a central node, namely sink is the destination of all data packets. For transmitting data to long distances, a lot of energy should be consumed. Hence, in many cases, nodes communicate with the sink node through their neighbors. In this case, each node should know which neighbor is more appropriate for packet transmission. In WSNs, congestion control mechanisms are important techniques for decreasing timeliness and increasing packet delivery rate [2].

Communication protocols play a significant role in enhancing the efficiency and lifetime of WSNs [3,4]. On the other hand, energy is one of the most important parameter and the key objectives in data gathering in WSNs. Hence, designing efficient protocols with regard to energy consumption for WSNs is essential since it can not only reduce the total energy consumption in the network but also distribute energy steadily and uniformly to the network nodes. Recently, many algorithms have been proposed for data aggregation in WSNs which try to find routes towards the sink through which data can be aggregated.

Given a data aggregation tree, sensors receive messages from children periodically, merge them with its own packet, and send the new packet to its parent. The problem of finding an aggregation tree with the maximum lifetime has been proved to be NP-hard and can be generalized to finding a spanning tree with the minimum maximum vertex load, where the load of a vertex is a nondecreasing function of its degree in the tree [5].

Among the proposed protocols, data aggregation technique [6], [7] is an energy conservation scheme which tries to decrese the volume of data communicated by collecting local data at intermediate nodes and forwarding only the result of an aggregation operation, such as min and max, towards the sink node [8], [9], [10]. In data aggregation technique [11], [12], [13], [14], [15] relevant data packets were combined with one another in intermediate nodes and form a packet. Hence, the number of packets to be transmitted in the network decreases [16], [17]. Consequently, less energy will be consumed [16].

In this paper, an efficient and low-energy data aggregation method based on COA [18] is introduced which can reduce the total communication energy through creating a data aggregation tree. The problem addressed in this paper is an important special case of the general data aggregation problem in which all the sensor nodes in the network are source nodes. Hence, it can efficiently

* Corresponding Author

reduce energy consumption and enhance network lifetime. The objective in this paper is to simultaneously minimize total energy consumption during the entire tree structure.

The rest of the paper is organized as follows: in section 2, a summary of the related studies and works is briefly overviewed. Then, the proposed algorithm is described in section 3. Then, simulation results are reported in section 4. Finally, the conclusion of the study is given in section 5.

2. Related Works

Regarding related studies on data aggregation, In [19], PEDAP (Power Efficient Data Gathering and Aggregation Protocol) was proposed. PEDAP computes a routing tree with MST (Minimum Spanning Tree) algorithm by setting the energy cost delivering a data packet between two sensors as the weight of the link between them. However, routing trees constructed with MST are not optimal energy efficient routing trees and cannot obtain long network lifetime. Based on PEDAP, another algorithm, PEDAP-PA (Power Efficient Data Gathering and Aggregation Protocol-Power Aware) [19], was proposed two minimum spanning tree based data gathering and aggregation schemes to maximize the lifetime of the network, where one is the power aware version of the other. The non-power aware version (PEDAP) extends the lifetime of the last node by minimizing the total energy consumed from the system in each data gathering round, while the power aware version (PEDAPPA) balances the energy consumption among nodes. In PEDAP, edge cost is computed as the sum of transmission and receiving energy. In PEDAPPA, however, an asymmetric communication cost is considered by dividing PEDAP edge cost with transmitter residual energy. A nodewith higher edge cost is included later in the tree which results few incoming messages. Once the edge cost is established, routing information is computed using Prim's minimum spanning tree rooted at base station. The routing information is computed periodically after a fixed number of rounds. These algorithms assume all nodes perform in-network data aggregation and base station is aware of the location of the nodes.

In [20], EESR is proposed that uses energy efficient spanning tree based multi-hop to extend network lifetime. EESR generates a transmission schedule that contains a collection of routing trees. In EESR, the lowest energy node is selected to calculate its edge cost. A node calculates its outgoing edge cost by taking the lower energy level between sender and receiver. Next, the highest edge cost link is chosen to forward data towards base station. If the selected link creates a cycle, then the next highest edge cost link is chosen. This technique avoids a node to become overloaded with too many incoming messages. The total procedure is repeated for the next lowest energy node. As a result, higher energy nodes calculate their edge costs later and receive more incoming messages, which balances the overall load among nodes. The protocol also generates a smaller transmission schedule, which reduces receiving energy. The base station may broadcast the entire schedule or an individual tree to the network.

Wang et al [21] devised a data aggregation tree based on first-fit algorithm for reducing data transmission delay. Chaon et al [22] proposed SFEB (structure-free and energy-balanced) protocol which is an unstructured data aggregation protocol; it results in a balanced energy consumption among the nodes. Consequently, network lifetime is enhanced. Liu et al [23] proposed an approximation algorithm using directional antennas for aggregating data which led to the reduction of interface in transmitting and receiving data in network. Using ant colony algorithm, Liao et al [24] proposed a routing protocol which increased network lifetime. Islam et al., [25] produced a balanced data aggregation tree based on genetic algorithm which was energy-efficient. The authors used GA to create multi-hop spanning trees and it was not based on hierarchical clusters. The data aggregation trees created by the proposed GA technique were more energy efficient than some of the current data aggregation tree-based techniques.

Using lexicographic method, Lai and Ravindran [26] investigated accessing maximum network lifetime and fairness regarding simultaneous resource allocation in data aggregation tree. Kwond et al., [27] allocated a dynamic time for data aggregation in a node which was characterized with high energy efficiency and little delay overhead resistance. Al-Karaki et al [28] proposed a design which used optimization and heuristic algorithms simultaneously for locating minimum data aggregation points and routing data to base station so that network lifetime is maximized.

In [29], Firstly, the authors proposed an adaptive spanning tree algorithm (AST), which can adaptively build and adjust an aggregation spanning tree. Owing to the strategies of random waiting and alternative father nodes, AST can achieve a relatively balanced spanning tree and flexible tree adjustment. Then, the authors presented a redundant aggregation scheme (RAG). In RAG, interior nodes help to forward data for their sibling nodes and thus provide reliable data transmission for WSN. The simulation results demonstrate that AST can prolong the lifetime and RAG makes a better trade-off between storage and aggregation ratio, comparing to other aggregation schemes.

In [30], the authors proposed a novel Hierarchical Data Aggregation method using Compressive Sensing (HDACS), which combines a hierarchical network configuration with compressive sensing (CS). The authors' main idea is to set multiple compression thresholds adaptively based on cluster sizes at different levels of the data aggregation tree to optimize the amount of packet delivery rate (PDR). The advantages of the proposed model in terms of the total amount of PDR and data compression ratio are analytically verified.

In [31], the authors presented the problem of scheduling virtual data aggregation trees to maximize the network lifetime when a fixed number of data are allowed to be aggregated into one packet, termed the Maximum Lifetime Data Aggregation Tree Scheduling (MLDATS) problem. The authors showed that the MLDATS problem is to be NP-complete. In addition, the authors proposed a local-tree-reconstruction-based scheduling algorithm (LTRBSA) for the MLDATS problem.

In [32], the authors applied data aggregation on two nodes levels in WSNs. They worked on sending fewer data from aggregator to the sink, along with the equation that expresses all data. They applied Bayesian belief network algorithm to measure the accuracy of the proposed scheme.

In [33], the authors propose a new scheme in the network layer, called Weighted Compressive Data Aggregation (WCDA), which benefits from the advantage of the sparse random measurement matrix to reduce the energy consumption. The novelty of the WCDA algorithm lies in the power control ability in sensor nodes to form energy efficient routing trees with focus on the load-balancing issue. In the second part, they present another new data aggregation method namely Cluster-based Weighted Compressive Data Aggregation (CWCDA) to make a significant reduction in the energy consumption in our WSN model. The main idea behind this algorithm is to apply the WCDA algorithm to each cluster in order to reduce significantly the number of involved sensor nodes during each CS measurement. In this case, candidate nodes related to each collector node are selected among the nodes inside one cluster. This yields in the formation of collection trees with a smaller structure than that of the WCDA algorithm

In this paper, a new method is proposed for WSNs which selects data aggregating nodes via COA. Indeed, the purpose of the study is to aggregate data with the aid of COA. Further, as our proposed method uses data aggregation spanning trees, it is only compared with other data aggregation spanning tree approaches such as PEDAPPA [19] and EESR [20] and GA [25]. In this paper, the frequency of usage of data aggregation tree phase is fixed and it is not dynamically adjusted.

3. The Proposed Method

In this section, we briefly describe our proposed scheme.

3.1 Preliminary Definitions

The primary model for WSNs which was presumed in this study is as follows:

- The network has *n* sensors which have been randomly and uniformly distributed in a pre-specified environment.
- Sensors and the main database have a fixed and certain location. They are not capable of moving.
- Nodes are distributed uniformly and randomly.
- The initial energy of the nodes is identical. It is equal to 1*J*.
- All the sensors can be detected and identified by means of their own unique IDs.

The following system model is used in this paper. The network consists of n wireless sensors, each containing a transceiver with a maximum transmission range. This work adopts the use of multi-hop transmis-

sions in its communication model. For this purpose, a tree is constructed. In-network aggregation at intermediate nodes usually results in reducing the size of data that is forwarded by an intermediate node to its parent [7]. An undirected graph, i.e. G (V, E), was used for featuring the sensor network with V nodes which was produced based on the distance between nodes and the radio range. Random graph was produced in the following way: V sensors were randomly placed in a pre-specified environment where they were connected to their neighbors with the Euclidean distance equal to or less than their radio range.

Definition 1: in a data aggregation cycle, the data received from the environment is gathered by the intermediate nodes and transmitted to the base station.

Definition 2: data aggregation tree is a spanning tree with the root of base station which includes routing data for all the network nodes.

Definition 3: network lifetime is defined as the number of cycles of algorithm execution where all the network nodes are active.

Definition 4: the load of a sensor refers to the required energy for receiving and transmitting gathered data to its own father.

3.2 Cuckoo Optimization Algorithm (COA)

One notable search technique in computer science is to find the optimal solution in searching issues. COA is considered to be a supplementary algorithm which has been inspired from the life of a species of a bird known as cuckoo. It was developed in combination with levy flight instead of simple isotopic walk.

Evolution commences with a completely randomized set of entities and is repeated in the next generations. In each generation, the most appropriate ones but not the best ones are selected. A solution for a specific problem is illustrated through a list of parameters which are referred to as habitats. At the outset, several features are randomly produced for creating the first generation. During each generation, all the features are evaluated and the fitness value is measured by the fitness function. The next step is to produce the second generation of the community. For each individual, new locations are selected for the egg laying and migration stages. These selections are made in a way that they have the highest values with regard to the fitness function. Consequently, the most appropriate habitat with the highest benefit should be selected. The next steps are related to finding new locations with the maximum optimization benefit.

Since metaheuristic algorithm has no information about the global optimization point and the degree of the optimality of the solutions, certain criteria are needed for stopping them. The algorithm proposed in this study was designed in a way that it stops after the production of a number of generations. That is to say, if the populations of cuckoos get close to one another, practically, there will be no homogeneity and all the cuckoos reach an optimal point. In this case, algorithm will stop.

3.3 The Proposed Algorithm

Since the majority of data aggregation trees are produced based on remaining energy or the distance between sensors, hence, both parameters, namely the remaining energy and the distance between sensors were taken into consideration. As mentioned before, the algorithm proposed in this study for producing data aggregation tree is COA.

A. Habitat structure

Habitats provide data aggregation trees. Each habitat has a specific length which is appropriate for a specific number of available nodes in the network. The index of the array indicates the ID (identification) and the content of the array indicates the identity of its father. Figure 1 shows a habitat where the network includes 100 nodes. ID of the first node is 1 and the ID of its father is 5. Hence, the last node's father is node 43. The node with the zero value was regarded as the workstation or the root of the tree.

1	2	3	4	…	98	99	100
5	6	21	0	…	2	54	43

Fig. 1. Habitat structure

The production of the initial population and the next generations from it with regard to the above-mentioned structure of the habitat will be based on the following conditions:

- Each habitat should have one zero value (indicating workstation).
- The content of the array should not be the same as its ID.
- The content of the array has a random value between zero and *N* (*N* refers to the number of nodes).

Since the features within the habitats are randomly produced in creating the initial population, a node will be selected as the father where there is no mane between them in the graph. Thus, the produced habitat is invalid and other random numbers are produced.

B. Population

Population refers to a set of habitats. According to another definition, population includes valid data aggregation trees. For the initial population, parent nodes are randomly selected by acknowledging their validities.

C. Generation

The new generation is selected by applying egg laying operation and migration. The best habitat is selected based on the fitness function so that it is transmitted to the next generation.

1. Egg laying

In COA, the new generation is selected after the egg laying operation. Then, in the next stage, cuckoos move towards the optimal response with a little change. For using the positive characteristics of habitats, the operators are used in this way: each cuckoo randomly lays some eggs in the nest of the host bird which is located in the ELR (egg laying range) distance from it. Then, the profit function is determined for all the cuckoos. The best cuckoos with appropriate positions are selected according to the maximum number of cuckoos and are transmitted to the next generation. The remaining cuckoos which have less profit are destroyed. ELR (maximum egg laying range) is specified via equation (1):

$$\mathrm{ELR} = \alpha \times \frac{\text{Number of currcnt cuckoo}'\text{seggs}}{\text{Total number of eggs}} \times (\mathrm{var_{hi}} - \mathrm{var_{low}}) \quad (1)$$

In this equation, $\mathrm{var_{hi}}$ refers to the maximum number of each cuckoo's eggs, $\mathrm{var_{low}}$ denotes the minimum number of each cuckoo's eggs and α is a variable which adjusts the maximum value of ELR.

2. Migration

If only the egg laying operator is used for benefiting from the good features of habitats, the problem of premature convergence might occur. Hence, for preventing this problem and using the entire space for search, we use the migration operator. That is, when cuckoo chicks grow up and become mature, they live in the environment and their own groups for a while. However, when the egg laying time is approaching, they migrate to better habitats where there is more chance of being alive. After cuckoo groups in different areas are established, the average profit is measured so that the relative optimization of each group is obtained. For selecting the target point inside the chosen cluster with more average profit, the profit of all the cuckoos inside this cluster is measured and the cuckoo with the highest profit is selected as the target point. For specifying the new position of cuckoos around this point, the current position of the cuckoos is multiplied by the movement coefficient of the cuckoos and the new position of the cuckoos inside the optimal cluster is determined. The new position of the cuckoos after the migration operation is defined through Equation (2):

$$\mathrm{X_{Next\ habitat}} = \mathrm{X_{Current\ habitat}} + \mathrm{F} \times (\mathrm{X_{Goal\ point}} - \mathrm{X_{current\ habitat}}) \quad (2)$$

In Equation 2, F is a parameter which leads to deviation.

3. Correction function

Since different problems such as father-child, child-father problems, habitat without workstation and circle production might occur in the mentioned habitat, a correction function will be used for detecting and solving these problems which will prevent the transmission of invalid habitats to the next generation.

Habitat without work station: in habitat, the workstation has the value of zero. However, it is likely that no habitat has the value of zero for its content since the numbers are selected randomly. The following figure illustrates this problem for a network with 5 nodes.

1	2	3	4	5	6
3	2	4	2	3	5

Fig. 2. Habitat-related problem without workstation

Circle problem: this condition occurs when the content of array is the same as its index. Figure 3 shows this problem which occurs in the habitat.

1	2	3	4	5	6
3	1	6	4	3	5

Fig. 3. Circle problem

Father-child and child father problems: these problems occur when child a belongs to father b and when child *b* belongs to father *a* which is depicted in the following figure.

1	2	3	4	5	6
3	4	3	2	3	5

Fig. 4. Father-child, child-father problem

In case the circle problem occurs, one unit is added to the content of the array index. For solving the problem of residence location without workstation, zero value is randomly given to one of the elements of the array.

For sorting out the father-child, child-father problems, we begin with the first index which has a value other than zero. That is, we search for the related index in the elements of the array; if the obtained value has an index equal to the previous value, the problem has occurred; otherwise, we continue the procedure so as to navigate all the elements of the array. In case this problem occurs, one unit is added to the content of the related element. Figure 5 illustrates a pseudocode of the correction function which is executed for each habitat and it checks that the problems mentioned in the previous parts do not occur. If one of these problems occurs, the correction function will detect and correct it.

Correction function pseudo code

1: **Procedure** Repair function (habitat)
2: **For** each *habitat_i* in habitat **do**
3: **While** habitat$_i$ creates a cycle **do**
4: Replace value of *habitat$_i$* with (value of *habitat$_{i+1}$*)
5: **While** habitat have not a sink **do**
6: Randomly insert zero In the *habitat$_i$*
7: **While** habitat have a problem Father-Child and Child-Father **do**
8: Replace value of *habitat$_i$* with (value of *habitat$_{i+1}$*)
9: **While** *habitat$_i$* is not in range of board **do**
10: Replace value of *habitat$_i$* with other node ID
11: **End**.

Fig. 5.Correction function pseudo code

4. Evaluation and fitness

The algorithm proposed in this study has two purposes. Firstly, the tree should be effective with regard to energy consumption so that nodes can have more communications for long-term frequencies. Secondly, the tree produced among the nodes should be balanced. The fitness of the habitats is specified through the following parameters:

Energy expenditure rate: a sensor has different expenditures in producing different trees. The tree produced by the COA algorithm needs effective energy. It is preferred that each node consumes little energy in each communication cycle. For each *i* node, the energy consumption rate is measured through equation (3). This rate increases for all the sensor nodes.

$$L = \frac{e_i}{E_{rx}} \quad (3)$$

In this equation, e_i stands for the current energy of each node, E_{rx} denotes the amount of energy consumption for receiving data packets from the child. E_{rx} is determined through equation (4).

$$E_{rx} = k \times E_{elec} \quad (4)$$

In this equation, E_{elec} denotes the required energy for maintaining movement and traffic within the sensors and *K* stands for the number of available bits in the packet.

In this paper, the following generic model is used to account for the energy consumed during communication:

$$E_{Tx}(i,j) = k[E_{elec} + \varepsilon_{amp} \times d^2] \quad (6)$$

$$E_{Total}(i,j) = k[2E_{elec} + \varepsilon_{amp} \times d^2] \quad (7)$$

In Equations (6) and (7), *d* is Euclidian distance between two neighbor nodes *i* and *j*, ε_{amp} is he amplification energy required to transmit a bit a unit distance and $E_{Total}(i,j)$ is the total energy consumption for transmission and receiving *k* bits of data from node *i* to node *j*. For sake of simplicity, we assume that the energy consumed in the sleep state is negligible.

Method of measuring e_i: e_i refers to the degree of current energy. The initial value of e_i is considered to be 1*J* but this value varies during the execution of the algorithm so that after the execution of each cycle, the new e_i is specified via Equation 8.

$$r_i = e_i - E_{rx} \quad (8)$$

In this equation, r_i stands for the remaining energy of the *i*th node after the execution of a cycle. That is, the remaining energy of each node at the end of a cycle is considered to be the current energy of it in the next cycle.

Standard deviation of the remaining energy: after each cycle, the amount of energy E_{rx} will be consumed by the i sensor (n_i). For maintaining the nodes in the working condition, it is essential that the energy expenditures be distributed in a balanced way based on the remaining energy of the sensors. Standard deviation of the sensors of the data aggregation tree is regarded as an appropriate criterion for expressing energy expenditure among nodes. The lower STD (r_i), the higher the network lifetime. The parameter of standard deviation is measured through equation (9):

$$\sigma = \sqrt{1/_N \sum_{i=1}^{N}(e_i - e^-)^2} \quad (9)$$

In equation (6), e^- is determined via equation (10):

$$e^- = {}^1/_N \sum_{i=1}^{N} e_i \tag{10}$$

The distance between the sesonrs: the distance between the sensors is one of the parameters which can be taken into consideration for producing data aggregation tree since the closer is the father node to the children, the lower will be the cost of transmitting and receiving data.

$$\text{d}\ (parent, child\ ) = \sqrt{(x_p - x_{ch})^2 + \ (y_p - y_{ch})^2} \tag{11}$$

In this Euclidean equation, $(\boldsymbol{x_p}, \boldsymbol{y_p})$ denote father's location coordinates and $(\boldsymbol{x_{ch}}, \boldsymbol{y_{ch}})$ stand for the child's location coordinates.

Energy on level rate: since data aggregation task in the tree is done by the father nodes, hence, the higher is the father's energy compared with child's energy, the better it will be. For examining this issue, *E* parameter is used which is determined through the following procedure:

1. *E*=0 (the value of this parameter is initially zero)
2. For the correctness of the Equation ${}^{e_p}/_{e_{ch}} > 1$, one unit is added to *E* in one element of the array (*E*=*E*+*1*).

Data aggregating nodes are selected based on the following parameters: nodes' energy level, energy expenditure rate, distance between father and children nodes, and standard deviation of the remaining energy of the nodes. Based on these parameters, a degree is determined for each node and the node with the highest degree of fitness will be selected as the data aggregator. This grading is conducted using the following Equation:

$$F_i = \alpha_1(E) + \alpha_2(\tfrac{1}{D}) + \ \alpha_3(L_i) + \alpha_4(\tfrac{1}{std}) \tag{12}$$

In this equation, $\boldsymbol{\alpha_1}, \boldsymbol{\alpha_{2,}}, \boldsymbol{\alpha_3}\ and\ \boldsymbol{\alpha_4}$ denote the weight of the parameters and the values are considered in a way that they fit the units of the parameters and would be true in Equation (13).

$$\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1 \tag{13}$$

Since of the parameters values of the equations were simulated for several times, the obtained values for the above-mentioned parameters are as follows:

$\alpha_1 = 0.25$,$\alpha_2 = 0.25$ m, α_3=0.25, $\alpha_4 = 0.25$ j

Each habitat is measured based on target function so that the habitat with the highest profit is transmitted to the next generation.

4. Performance Evaluation

This section shows the effect of the proposed algorithm on network lifetime and energy consumption. In the following analysis, the sensor nodes are considered to be uniformly and randomly placed in a two-dimensional area. All sensor nodes are homogeneous and have equal, omnidirectional transmission patterns of range and the sink energy supply is adequate. The efficiency of the proposed algorithm was investigated via different tests and simulations which were conducted in MATLAB R2009a. It should be noted that the performance and the efficiency of the proposed algorithm was compared with those of GA [25], PEDAPPA [19] and EESR [20]. In this paper and in simulation phase, we assume that all nodes perform in-network data aggregation and all the sensor nodes in the network are static and are loosely synchronized to enable TDMA (time division multiple access) scheduling. The topology graph constructed from the network is connected. Weights associated with all the edges are positive and hence, there are no negative weight cycles. A node can adjust its power level depending on its requirements. The results for lifetime are obtained for the 95% confidence interval.

The comparison of these algorithms were done in different dimensions of the network with respect to network lifetime. The simulation parameters used in different tests and analyses are given in table 1.

Table 1. Parameters used in the simulation

Parameters	Values
Initial energy	1J
Number of sensors	100
Packet size	1000 Byte
Radio range	25 m
Cuckoos' movement coefficient	2
Maximum egg laying radius	15
E_{amp}	100 pJ/bit/m^2
E_{elec}	50 nJ/bit

In the conducted experiments, simulations were proceeded until the first node was destroyed. Indeed, the simulations were carried out in 10 rounds and each experiment was executed for three times and the average of the three executions were taken into consideration. The location of the base station in all the experiments was considered to be at the center of the sensing environment. Figures 6, 7 and 8 depict network lifetime for the network sizes of 50m*50m, 100m*100m and 200m*200m, respectively. The results of the simulations indicated that the proposed algorithm had better performance than GA [25], PEDAPPA [19] and EESR [20].

We have compared network lifetimes obtained in different networks with different number of sensors. As can be seen in the Figures, our proposed COA notably increases network lifetime.

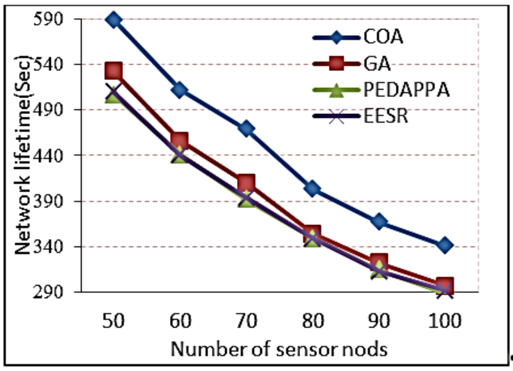

Fig. 6. Network lifetime in sensing environment with the size of 50m*50m

GA [25], PEDAPPA [19] or EESR [20] does not consider the maximal load of all sensors, while COA does. Therefore, network lifetimes obtained by GA [25], PEDAPPA [19] and EESR [20] are shorter than network lifetimes obtained by COA which reduces the maximal load of all sensors.

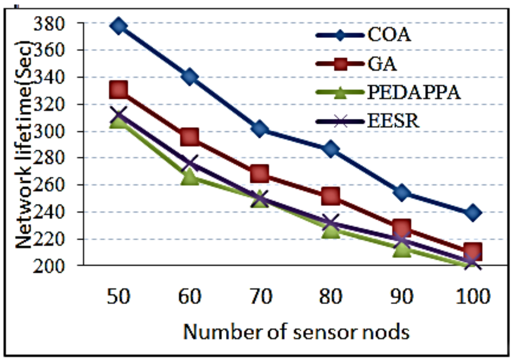

Fig. 7. Network lifetime in sensing environment with the size of 100m*100m

In PEDAP-PA [19], the sink needs to broadcast the topology of the network to all nodes in the network periodically; hence it consumes most energy. Due to fitness function of COA, it improves network lifetime in comparing with GA [25], PEDAPPA [19] and EESR [20].

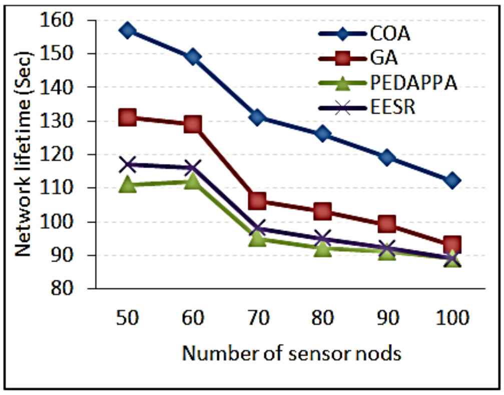

Fig. 8. Network lifetime in sensing environment with the size of 200m*200m

Figure 9 shows the remaining energy of networks after 100 rounds execution for CoA, PSO [34] (Particle Swarm Optimization) and ICA [35] (Imperial Competitive Algorithm). As Figure 9 depicts, the reaming energy for proposed scheme (COA) is better than PSO [34] and ICA [35]. Hence, the proposed scheme is energy efficient and improves network lifetime.

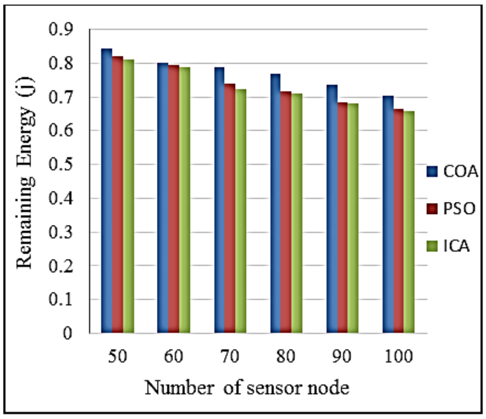

Fig. 9. Remaining energy of network after 100 rounds execution

Figure 10 shows the convergence diagram for proposed scheme.

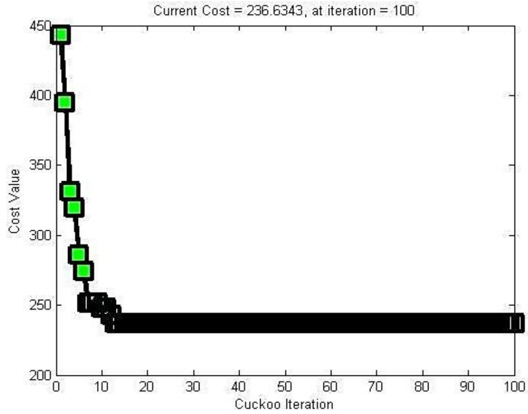

Fig. 10. Convergence diagram for proposed scheme (COA)

5. Conclusions

In this paper, a new method based cuckoo algorithm was proposed for aggregating data in WSNs. In this method, data aggregation tree was produced based on the following parameters: energy expenditure rate, standard deviation of the sensors' remaining energy, energy on level rate and the distance between the sensors. The proposed scheme for data aggregation trees extend the network lifetime as compared to EESR, PEDAPPA and GA. It was found that the proposed algorithm can reduce the energy consumption of the network significantly; hence, network lifetime is consequently enhanced. To sum it up, it can be concluded that the proposed method has desirable performance and efficiency. As a future work, we would like to investigate adaptive tree structure frequency techniques and we would like to use the proposed scheme in WSNs based on software defined networks (SDNs) [36] technology.

References

[1] J. Yick, B. Mukherjee, and D. Ghosal. "Wireless sensor network survey." Computer networks, vol. 52, pp. 2292-2330, 2008.

[2] A. Ghaffari. "Congestion control mechanisms in wireless sensor networks: A survey. " Journal of Network and Computer Applications, vol. 52, pp. 101-115, 2015.

[3] A. Ghaffari. "An energy efficient routing protocol for wireless sensor networks using A-star algorithm." Journal of applied research and technology, vol. 12, pp. 815-822, 2014.

[4] Z. Mottaghinia and A. Ghaffari. "A Unicast Tree-Based Data Gathering Protocol for Delay Tolerant Mobile Sensor Networks." Information Systems & Telecommunication, pp. 59, 2016.

[5] X. Zhu, G. Chen, S. Tang, X. Wu, and B. Chen. "Fast Approximation Algorithm for Maximum Lifetime Aggregation Trees in Wireless Sensor Networks." INFORMS Journal on Computing, vol. 28, pp. 417-431, 2016.

[6] A. Ghaffari, L. Darougaran, and A. Shiri. "Comparing data aggregation methods in wireless sensor networks." presented at the Third national symposium on computer engineering and information technology, 2010.

[7] S. Upadhyayula and S. K. Gupta. " Spanning tree based algorithms for low latency and energy efficient data aggregation enhanced convergecast (dac) in wireless sensor networks. Ad Hoc Networks, vol. 5, pp. 626-648, 2007.

[8] A. jhumka, M. Bradbury, and S. Saginbeko. "Efficient fault-tolerant collision-free data aggregation scheduling for wireless sensor networks." Parallel Distributed Computing, vol. 74, pp. 1789-1801, 2014.

[9] R. R. Rout and S. Ghosh. "Adaptive data aggregation and energy efficiency using network coding in a clustered wireless sensor network: An analytical approach." Computer Communications pp. 65-75, 2014.

[10] SH. Niu, C. Wang, Z. Yu, and S. Cao. "Lossy data aggregation integrity scheme in wireless sensor networks." Computers and Electrical Engineering, vol. 39, pp. 1726-1735, 2013.

[11] S. park. " performance Analysis of Data Aggregation Schemes For wirless sensor network, 2006

[12] S. Sicari, L. Grieco, G. Boggia, and A. Coen-Porisini. "DyDAP: A dynamic data aggregation scheme for privacy aware wireless sensor networks." The Journal of Systems and Software, vol. 85, pp. 152-166, 2012.

[13] Q. Liua, Y.Changa, and X. Jiab. "A hybrid metod of CSM/CA and TDMA for real-time data aggregation in wireless sensor networks." Computer Communications, vol. 39, pp. 269-278, 2013.

[14] W. Wu, J. Cao, H. Wu, and J.Li. "Robust and dynamic data aggregation in wireless sensor networks: A cross-layer approach." Computer Networks, vol. 57, pp. 3929-3940, 2013.

[15] H. Yousefi, M. Yeganeh, N.Alinaghipour, and A. Movaghar. "Structure-free real-time data aggregation in wireless sensor networks." Computer Communications, vol. 35, pp. 1132-1140, 2012.

[16] H.Li, CH.Wua, Q. Huab, and F. Lau. "Latency-minimizing data aggregation in wireless sensor networks under physical interference mode." Ad Hoc Networks, vol. 12, pp. 52-68, 2014.

[17] S. Ozdemir and Y.Xiao. "Integrity protecting hierarchical concealed data aggregation for wireless sensor networks." Computer Networks, vol. 55, pp. 1735-1746, 2011.

[18] R. Rajabioun. "Cuckoo Optimization Algorithm." Applied soft computing, vol. 11, pp. 5508- 5518, 2011.

[19] H. Ö. Tan and I. Körpeoğ lu. "Power efficient data gathering and aggregation in wireless sensor networks." ACM Sigmod Record, vol. 32, pp. 66-71, 2003.

[20] S. Hussain and O. Islam. "An energy efficient spanning tree based multi-hop routing in wireless sensor networks." in 2007 IEEE Wireless Communications and Networking Conference, 2007, pp. 4383-4388.

[21] P.Wang, Y. He, and L.Huang. "Near optimal scheduling of data aggregation in wireless sensor networks." Ad Hoc Networks, vol. 11, pp. 1287- 1296, 2013.

[22] CH.Chaon and T. Hsiao. "Design of structure-free and energy-balanced data aggregation in wireless sensor networks." vol. 37, pp. 229-239, 2014.

[23] H.Liu, Z.Liu, D.Li, X.Lub, and H.Due. "Approximation algorithms for minimum latency data aggregation in wireless sensor networks with directional antenna." Theoretical Computer Science, vol. 497, pp. 139-153, 2013.

[24] W. -H. Liao, Y. Kao, and C.-M. Fan. "An ant colony algorithm for data aggregation in wireless sensor networks." in Sensor Technologies and Applications, 2007. SensorComm 2007. International Conference on, 2007, pp. 101-106.

[25] O. Islam, S. Hussain, and H. Zhang. "Genetic Algorithm for Data Aggregation Trees in Wireless Sensor Networks." 2007.

[26] S. Lai and B.Ravindran. "Achieving Max–Min lifetime and fairness with rate allocation for data aggregation in sensor networks." Ad Hoc Networks, vol. 9, pp. 821-834, 2011.

[27] S. Kwond, J. H. Ko, and J.Kim. "Dynamic timeout for data aggregation in wireless sensor networks." Computer Network, vol. 55, pp. 650-664, 2011.

[28] J.AL-Karaki, R.Mustafa, and A.Kamal. "Data aggregation and routing in Wireless Sensor Networks: Optimal and heuristic algorithms." Computer Network, vol. 53, pp. 954-960, 2009.

[29] Y. Zhang, J. Pu, X. Liu, and Z. Chen. "An adaptive spanning tree-based data collection scheme in wireless sensor networks." International Journal of Distributed Sensor Networks, vol. 2015, p. 2, 2015.

[30] X. Xu, R. Ansari, A. Khokhar, and A. V. Vasilakos. "Hierarchical data aggregation using compressive sensing (HDACS) in WSNs." ACM Transactions on Sensor Networks (TOSN), vol. 11, p. 45, 2015.

[31] N.-T. Nguyen, B.-H. Liu, V.-T. Pham, and Y.-S. Luo. "On maximizing the lifetime for data aggregation in wireless sensor networks using virtual data aggregation trees." Computer Networks, vol. 105, pp. 99-110, 2016.

[32] I. Atoui, A. Ahmad, M. Medlej, A. Makhoul, S. Tawbe, and A. Hijazi. "Tree-based data aggregation approach in wireless sensor network using fitting functions." in 2016 Sixth International Conference on Digital Information Processing and Communications (ICDIPC), 2016, pp. 146-150.

[33] S. Abbasi-Daresari and J. Abouei. "Toward cluster-based weighted compressive data aggregation in wireless sensor networks." Ad Hoc Networks, vol. 36, pp. 368-385, 2016.

[34] J. Kennedy and R. Eberhart. "Particle swarm optimization." in Neural Networks, 1995. Proceedings., IEEE International Conference on, 1995, pp. 1942-1948 vol.4.

[35] E. Atashpaz-Gargari and C. Lucas. "Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition." in 2007 IEEE Congress on Evolutionary Computation, 2007, pp. 4661-4667.

[36] R. Masoudi and A. Ghaffari. "Software defined networks: A survey." Journal of Network and Computer Applications, vol. 67, pp. 1-25, 2016.

Elham Mohsenifard is a M.Sc graduate of Computer Engineering at Tabriz branch, Islamic Azad University, Tabriz, Iran. She received her B.Sc degree from Islamic Azad University (IAU). Her research interests include Wireless Sensor Networks (WSNs) and Mobile Ad Hoc Networks (MANETs).

Ali Ghaffari received his B.Sc, M.Sc and Ph.D. degrees in Computer Engineering from the University of Tehran and IAUT (Islamic Azad University), TEHRAN, IRAN in 1994, 2002 and 2010 respectively. As an assistant professor of computer engineering at Islamic Azad University, Tabriz branch, IRAN, his research interests are mainly in the field of wired and wireless networks, Wireless Sensor Networks (WSNs), Mobile Ad Hoc Networks(MANETs), Vehicular Ad Hoc Networks(VANETs), networks security and Quality of Service (QoS). He has published more than 60 international conference and reviewed journal papers.