

Cluster-based Coverage Scheme for Wireless Sensor Networks using Learning Automata

Seyyed Keyvan Mousavi

Department of Computer Engineering, Urmia Branch, Islamic Azad University, Urmia, Iran
mosavikeyvan90@gmail.com

Ali Ghaffari*

Department of Computer Engineering, Tabriz Branch, Islamic Azad University, Tabriz, Iran
A.ghaffari@iaut.ac.ir

Received: 27/Dec/2020

Revised: 11/May/2021

Accepted: 25/May/2021

Abstract

Network coverage is one of the most important challenges in wireless sensor networks (WSNs). In a WSN, each sensor node has a sensing area coverage based on its sensing range. In most applications, sensor nodes are randomly deployed in the environment which causes the density of nodes become high in some areas and low in some other. In this case, some areas are not covered by none of sensor nodes which these areas are called coverage holes. Also, creating areas with high density leads to redundant overlapping and as a result the network lifetime decreases. In this paper, a cluster-based scheme for the coverage problem of WSNs using learning automata is proposed. In the proposed scheme, each node creates the action and probability vectors of learning automata for itself and its neighbors, then determines the status of itself and all its neighbors and finally sends them to the cluster head (CH). Afterward, each CH starts to reward or penalize the vectors and sends the results to the sender for updating purposes. Thereafter, among the sent vectors, the CH node selects the best action vector and broadcasts it in the form of a message inside the cluster. Finally, each member changes its status in accordance with the vector included in the received message from the corresponding CH and the active sensor nodes perform environment monitoring operations. The simulation results show that the proposed scheme improves the network coverage and the energy consumption.

Keywords: Wireless sensor networks; Clustering; Learning automata; Network coverage.

1- Introduction

Wireless sensor networks are consist of a large number of sensor nodes which are densely deployed inside a phenomenon or very close to it [1-6]. One of the basic challenges in wireless sensor networks is the coverage problem [7, 8]. Coverage favors the placement of sensors in the environment and determines how much the environment is monitored by the sensors. In WSNs, sensor nodes are usually distributed randomly in the environment which causes the density of nodes become high in some areas and low in some other [9, 10]. The redundancy of sensor nodes in an area, firstly leads to waste the energy of some sensor nodes which causes to reduce the network lifetime and secondly, leads to overlap that area with a high probability while some areas may remain out of coverage [11-14]. Hence, in these networks in order to reduce the amount of overlap of sensor nodes and optimal

coverage of the network and also reducing the energy consumption and prolonging the network lifetime, identifying the redundant nodes seems to be an essential problem [15-18].

Three main categories of coverage related problems can be identified into three classes: (i) target coverage, (ii) barrier coverage, and (iii) area coverage. In target coverage class, we try to monitor various targets with sensor nodes in target tracking applications. In barrier coverage, we try to minimize the probability of undetected penetration through a sensor node. On the other hand, area coverage can be divided into two groups: partially coverage and full coverage.

In partially coverage, a determined percent of monitored area can be covered with distributed sensor nodes. While in full coverage class we must continuously monitor whole interested area [19, 20].

This paper proposes a new method based on learning automata to improve the coverage and to prolong the lifetime of WSNs. The proposed scheme reduces the

* Corresponding Author

amount of energy consumption in the network and as a result increases the network lifetime by identifying and disabling the redundant nodes. The simulation result shows that proposed method improves the network performance metrics such as coverage and energy consumption.

Briefly, the main contribution of this paper are as follows:

- Discussing the various related works about WSNs coverage optimization strategies.
- Proposing a new coverage optimization mechanism in WSNs using learning automata.
- Evaluation and validation of the effectiveness of proposed mechanism in MATLAB simulator under varying sensor nodes.

The rest of the paper is organized as follows. Section 2 indicates related works. Section 3 describes the proposed scheme. Section 4 describes the performance evaluation of the proposed scheme and finally, Section 5 provides the conclusion and future works.

2- Related Works

This section provides an overall picture of the literature related to the coverage problems in WSNs.

In [21], a method based on target coverage in heterogeneous wireless sensor networks has been proposed. This method is one of the connected k-coverage methods. In addition, each node in the coverage set can communicate with the sink either directly or via a multi-hop connection through its neighbors. In this method, neighbor nodes can be used as a relay to establish connection between nodes in the network. The disadvantages of this method are high energy consumption and as a result low network lifetime.

In [22], a method called coverage and energy strategy has been proposed for WSNs. In this method, each sensor node has four modes including INITIAL, WORKING, SLEEPING and CHECKING. This algorithm includes two phases. In the first phase, all nodes are initially in WORKING mode. In WORKING mode, firstly each node exchanges its location information with its neighbors and then estimates the coverage share of itself with the neighbor nodes which are in WORKING mode. If the coverage share is independent or the neighbor node which is in WORKING mode is a redundant node, the node goes to SLEEPING mode otherwise it stays in WORKING mode. In the second phase, the nodes try to exchange their duties between SLEEPING and WORKING modes based on the coverage and connectivity conditions. In SLEEPING mode, the nodes will periodically wake up and switch to CHECKING mode. A node which is in WORKING mode goes to SLEEPING mode if its neighbor node in CHECKING mode, has a greater level of residual energy. The large

number of active nodes in this method leads to waste the energy and also redundancy in the coverage. In [23], an algorithm called Imprecise Detections Algorithm (IDA) has been proposed to improve coverage in wireless sensor networks. In this algorithm, firstly the environment is modeled as a grid and the sensor nodes are located in the grid points. Suppose the probability of detecting a target by a node has an exponential relation with the distance - between the target and the sensor node. This means a target located at distance d from a sensor node is detected by that node with the probability of $e^{-\alpha d}$. The algorithm presented in [24] by using of Voronoi diagram, tries to select the most suitable nodes for complete coverage of the environment. One of the factors that has been considered in this algorithm for selecting active nodes is the residual energy of nodes. This algorithm tries to increase the network lifetime by balancing the energy consumption of entire network. In this algorithm, all nodes are initially in the sleep mode. Then, one of the sensor nodes is selected as the starting node and creates a Voronoi cell for other sensors.

In [25], a protocol called Coverage-Preserving Clustering Protocol (CPCP) has been presented for wireless sensor networks. To ensure balanced energy consumption among CHs in the network, most of existing clustering protocols favor uniformly distribution of clusters with stable average cluster sizes and obtaining the best cluster distribution number per a unit of time is the main goal of clustered wireless sensor networks. In this method, CHs are distributed uniformly entire the network by limiting the maximum cluster area. Thus, clusters in a sparse area are formed as well as clusters in a dense area which prevents the costly transfer by the costly nodes to the remote CHs. In [26], the authors proposed PCLA (Partial Coverage with Learning Automata), a novel algorithm that relies on learning automata to implement sleep scheduling approaches. It minimizes the number of sensors to activate for covering a desired portion of the region of interest preserving the connectivity among sensors. In [27], the authors proposed an integrated and energy-efficient protocol for Coverage, Connectivity, and Communication (C3) in WSNs. This protocol uses RSSI (received signal strength indicator) to divide the network into virtual rings, defines clusters with CH at alternating rings, and defines dings that are rings inside a cluster. This scheme uses triangular tessellation to identify redundant nodes, and sends gathered data to sink through CHs. In C3, the distance of CHs from the sink node is too long. This weakness makes the CHs apply more intermediate nodes to send their data directly to the sink node and consequently leads to more energy consumption in the network.

In [28], the authors proposed a novel algorithm called complex alliance strategy with multi-objective

optimization of coverage (CASMOC) which could improve node coverage effectively. CASMOC achieves the energy balance of whole network through giving the proportional relationship of the energy conversion function between the working node and its neighbors, and applies this relationship in scheduling low energy mobile nodes. In [29], the authors presented a distributed protocol, called single phase multiple initiator (SPMI). Based on determining a connected dominating set (CDS), SPMI connects cover set for assuring the coverage and connectivity in WSN. Without using sensors' location information, SPMI only requires a single phase to construct a CDS in distributed manner.

In [30], the authors proposed a cover set to find the minimum set of sensors that completely cover the sensing ranges within an interest area as a criterion by selecting an optimal number of active sensors considering residual energy and the cover set and to keep alive the important sensors for the sensing coverage task. Also, the authors proposed an area coverage-aware clustering protocol (ACACP) with optimum energy consumption considering the activation node, network clustering, and multi-hop communication to improve overall network lifetime while preserving coverage.

3- The Proposed Method

To address the mentioned problem of C3, in the proposed method the sink node plays its main role and acts as an interface between user and the network. Due to this reason, in the proposed method the probability of being CH for the nodes that are closer to the sink will be greater than C3 [27] method. This will lead to reduce the distance from the sink that makes the CHs to send their data directly to the sink apply fewer intermediate node and consequently less energy is consumed in the network. In C3 [27] method, the energy of sensor nodes is not considered to select the redundant nodes. Hence in the proposed method, the energy of sensor nodes is also considered to select the redundant nodes. Another weakness of C3 [27] method is that the selection process of redundant nodes is done greedily. Though this method leads to increase the amount of coverage at the beginning of network lifetime, but with the passage of time, the algorithm will converge to the local optimal. While in the proposed method, the action of selecting the redundant nodes is performed by the use of intelligent learning automata algorithm that causes the proposed method converge to the optimal solution as time goes by and doesn't trap in a local optimal. In the proposed method, each sensor node can be in two different modes, including: Active mode and sleep mode.

In the proposed method, it is assumed that all the sensors deployed in the environment have the same sensing area and initial energy and each sensor node is aware of its geographic location by using global positioning system (GPS) or other positioning devices of the network. Also, it is assumed that all sensor nodes are initially in the active mode. After deploying the sensors in the environment, the sensor nodes by broadcasting the Hello packets, identify all its neighbors and send their status information to these neighbor nodes. The proposed algorithm consists of three phases in each round. The first phase includes network clustering, formation of the action set vector and the action probability vector. In the second phase, the coverage set within each cluster is created by the CH node using learning automata. Finally, in the third phase, the coverage set begin to monitor the environment. The details of each of these phases are explained below.

3-1 First Phase: Network Clustering, Formation of Action Set Vector and Action Probability Vector

In this phase, the following three types of messages can be used:

- i) Status Message: by using this message, each sensor node sends the information about its status such as neighborhood degree, overlap degree and residual energy in order to become a CH to all its neighbors.
- ii) CH Message: The candidate node to become a CH, by using this message announces itself as a CH along with the latest information about its status to all its neighbor nodes.
- iii) Membership Message: Each sensor node by using this message, declares itself as a single-hop member of a certain CH.

In this phase, it is tried to select the CHs among the nodes located in the densely areas of network and with the highest overlap degree. The degree of a node, is referred to the number of single-hop neighbors of that node in the network and the overlap degree is referred to the number of neighbors that overlap with this node at the certain points. To calculate the degree of a node, after network setup and deployment of the sensors in a given environment, the sink node sends a setup message to all nodes in the network. Each sensor node upon receiving this message, at a random time in the interval between zero and T_{max} , transmits its information to all its neighbors. Each neighbor node upon receiving this message and by checking the content of the message, can determine its degree. The reason of using a random time in the range between zero and T_{max} is to reduce collision while sending the HELLO packets. To determine the overlap degree of a node with its neighbors, firstly with considering the fixed points in the sensing area of the sensor nodes, the area coverage problem is converted to

the target coverage problem and then by using Eq. (1), the overlap degree of nodes is calculated:

$$OD_i = \frac{\sum CT_i}{\sum NT_i} \quad (1)$$

In Eq. (1), $\sum CT_i$ indicates the total number of times that the target points existing in the sensing area of node i are covered by its neighbors and $\sum NT_i$ indicates the sum of the target points existing in the sensing area of node i . In figures 1 and 2 the hypothetical sensor nodes A and B have the same number of neighbors, but their overlap degrees are different from each other. The neighborhood degrees of both nodes A and B are equal to 4, but since the target points of node A have been covered 6 times by its neighboring nodes and the target points of node B have been covered 4 times by its neighbors.

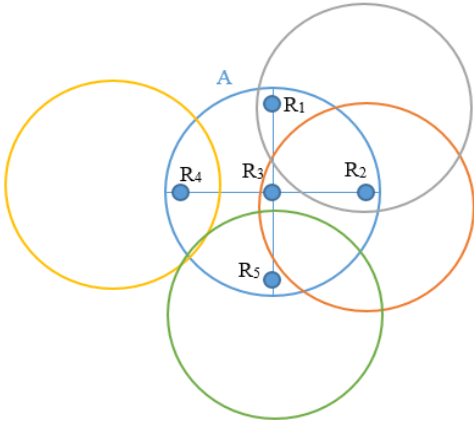


Fig. 1. The status of the target coverage of neighbors of the hypothetical node A

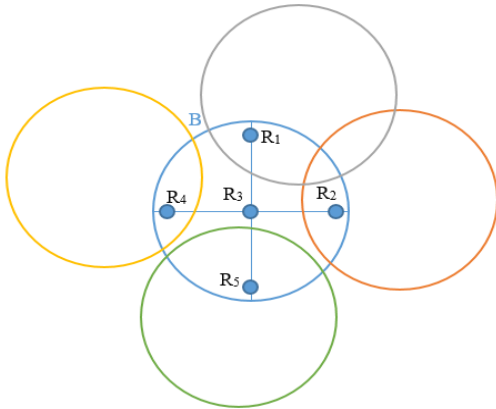


Fig. 2. The status of the target coverage of neighbors of the hypothetical node B

After determining the neighborhood degree and the overlap degree of sensor nodes, in the next step each node after sending its status to all its neighbors, waits for a certain time period T_{ni} until it receives a CH message from the nearest candidate CH. If within this time period it doesn't receive any message, announces itself as a CH

and broadcasts a CH message along with the information about its status throughout the network. The waiting time T_{ni} of each node is calculated using Eq. (2) [31]:

$$T_{ni} = \alpha \cdot e^{\frac{1}{d(n_i) + od(n_i)}} \quad (2)$$

In Eq. (3), $d(n_i)$ shows the neighborhood degree or the number of neighbors of node n_i and $od(n_i)$ indicates the overlap degree of node n_i . Also, the constant factor α should be set such that the condition $0 < T_{ni} \leq T_{max}$ is satisfied.

To minimize the number of clusters and select the best node in that area as CH, the residual energy of nodes is also considered. In this case Eq. (2) turns into Eq. (3) and the node having the highest residual energy is chosen as CH. As a result, instead of having multiple clusters in an area, we will have a single cluster.

$$T_{ni} = \alpha \cdot e^{\frac{1}{d(n_i) + od(n_i)} + \frac{E_{res}(n_i)}{E_{ini}}} \quad (3)$$

where in Eq. (4), E_{ini} and $E_{res}(n_i)$ indicate the initial and the residual energy of node n_i . In the next phase, each sensor node upon receiving the CH message from the nearest candidate node, declares itself as a member of that CH by sending a message containing its information. The CH upon receiving this message, stores its member node's information in its database and identifies this node as a single-hop member of itself.

After selecting the cluster heads and clustering the network, the CHs determine a certain time interval for each of its members to send their information so that each member can send its information to the corresponding cluster only within this time interval. This determined time interval to send the information by each cluster member is announced through sending a message by the cluster head node. With scheduling the member nodes by the cluster heads, it is prevented any collision of information while sending data to the cluster head. In order to balance the network load, at the beginning of each period, the clustering process is repeated again. After selection of the cluster heads and clustering the network, firstly inside the clusters each node creates the action vector and the probability vector of learning automata for itself and its neighbors. Then each node according to the probability vector of learning automata, determines the state of itself and its neighbors randomly and sends it as a message to its corresponding cluster head. The action vector of automata of a node includes the status of that node and its neighbors and is a vector as $VA = \{a, s\}$. In this vector, "a" indicates the active mode and "s" shows the sleep mode of a node. The probability vector of a node is equal to 0.5 initially and is a vector as (0.5, 0.5). In figure 4, the six nodes named A, B, C, D, E

and F are the members of a cluster and the node CH is the cluster head. Also, nodes B, C, D, E and F are the one-hop neighbors of node A.

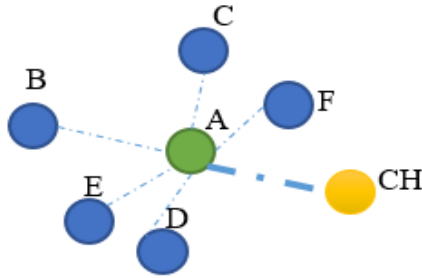


Fig. 3. The hypothetical node A and its single-hop neighbors inside the cluster

In figure 4, the node A firstly creates the action vector and the probability vector of itself and its neighbors as Table 1. Then as Table 1, according to the probability vector of itself and its neighbors, randomly selects action “a” or “s” for itself or its single-hop neighbors and determines the state of itself and its neighbors accordingly. Finally, as Table 1, sends the final status of itself and its neighbors to the cluster head as a message. The process of selecting the automata action by a node for itself and its neighbors is as follows: firstly, the node generates a random number between 0 and 1 for its intended node and then compares this number with the range of existing numbers in the probability vector related to each action of its intended node. Afterward, it considers the action that the generated number is within its probability range as the state of the node. For example, in figure 4, the node A to select the state of node B, firstly generates a random number as 0.6. Then, according to the probability vector of node B where the selection range of action “a” is (0, 0.5) and the selection range of action “s” is (0.5, 1), determines the placement range of the random number 0.6. Then, since the random number 0.6 is within range (0.5, 1), thus node A selects action “s” as the status of node B. The node A continues this process until it has not set the status of itself and all its neighbors.

Table 1. The action and probability vectors of the generated automata by node A for itself and its neighbors

Node	Action vector	Probability vector
A	{a, s}	{0.5, 0.5}
B	{a, s}	{0.5, 0.5}
C	{a, s}	{0.5, 0.5}
D	{a, s}	{0.5, 0.5}
E	{a, s}	{0.5, 0.5}
F	{a, s}	{0.5, 0.5}

Table 2. The Selected action by node A for itself and its neighbors

node	Generated number	Selected action
A	0.4	a
B	0.6	s
C	0.3	a
D	0.7	s
E	0.1	a
F	0.6	s

Table 3. The transmitted message from node A to the cluster head

node	Action vector	Probability vector	Selected action
A	{a, s}	{0.5, 0.5}	a
B	{a, s}	{0.5, 0.5}	s
C	{a, s}	{0.5, 0.5}	a
D	{a, s}	{0.5, 0.5}	s
E	{a, s}	{0.5, 0.5}	a
F	{a, s}	{0.5, 0.5}	s

3-2 Second Phase: Formation of Coverage Set

In this phase, first of all, the cluster head according to the information received from its cluster members, starts to reward or penalize the received messages by the use of the relations of learning automata [7] and then transmits the results to the sender of each received message. Afterward, among the transmitted vectors to its cluster members, it selects the best vector of actions and broadcasts the final status of nodes to its members in the form of a message for optimal covering the cluster. For example, suppose that the cluster head CH has received Table 3 as a message from node A. Among the one-hop members of the cluster, the nodes having the highest overlap degree are considered as the redundant nodes and should go into the sleep mode. To achieve this goal, the cluster head according to Table 3, firstly creates Table 4 based on the overlap degree and the residual energy of nodes and to calculates the value of *un* field of each node by using Eq. (4):

$$un_i = \beta \times \frac{od(n_i)}{Max(od)} + (1 - \beta) \times \frac{E_{res}(n_i)}{E_{res}(n_i)} \quad (4)$$

In Eq. (5), *Max(od)* indicates highest overlap degree among the nodes included in the forwarded message to the CH, *E_{res}(n_i)* indicates the minimum residual energy among the nodes, *E_{res}(n_i)* indicates the residual energy of node *n_i* and β is a constant number between 0 and 1. In this paper, β is set to 0.65.

Table 4. Details of the existing sensor nodes within the received message by the cluster head

Node name	Status	Residual energy	Overlapping degree	Amount of <i>un</i>
A	a	1.3 j	1.2	0.7010
B	s	1.5 j	1.3	0.7088
C	a	1.6 j	1.4	0.7008
D	s	1.5 j	0.8	0.9588
E	a	1.5 j	1.1	0.7816
F	s	1.7 j	1.2	0.7833

Afterward, the cluster head sorts Table 4 based on the un values of nodes in descending order from the highest value to the lowest value and by using Eqs. (5) to (8), learning automata [7] starts to reward or penalize the nodes and finally converts Table 4 into Table 5. Ideally, the nodes having the highest un values should go into the sleep mode, because they have the highest overlap degree and the lowest residual energy. Accordingly, if the sleep mode is selected for a node while the node with a higher un value is in the active mode, this action is penalized by the cluster head via Eqs. (7) and (8), otherwise it is rewarded using Eqs. (5) and (6).

$$P_i(n+1) = P_i(n) + a \times (1 - P_i(n)) \quad (5)$$

$$P_j(n+1) = (1 - a) \times P_j(n) \quad \forall_{j,j \neq i} \quad (6)$$

$$P_i(n+1) = (1 - b) \times P_i(n) \quad (7)$$

$$P_j(n+1) = \left(\frac{b}{r-1}\right) + (1 - b) \times P_j(n) \quad \forall_{j,j \neq i} \quad (8)$$

In relations (5) and (7), $P_i(n)$ indicates the probability of i^{th} action in the n^{th} step and $P_i(n+1)$ indicates the probability of i^{th} action in the $(n+1)^{\text{th}}$ step. In Eqs. (5) and (6), a shows the coefficient of reward and in relations (6) and (8), b shows the coefficient of penalty. In relations (7) and (8), $P_j(n)$ indicates the probability of j^{th} action in the n^{th} step and $P_j(n+1)$ indicates the probability of j^{th} action in the $(n+1)^{\text{th}}$ step. Furthermore in Eq. (8), r shows the number of automata actions.

Table 5. The sorted version of table 4 based on un values

Hostname	Status	Amount of un	Reward / penalty
D	s	0.9588	reward
F	s	0.7833	reward
E	a	0.7816	penalty
B	s	0.7088	reward
A	a	0.7010	reward
C	a	0.7008	reward

As an example, in Table 5, to assign a reward or penalty to the actions selected by node A, the cluster head node starts from the node with the highest value of un . If the selected node is in status “a” and the status of the next neighbor node that has a fewer un value is “s”, thus that action will be penalized. If the selected node is in status “s” and the status of the next neighbor node that has a fewer un value is “a”, hence that action will be rewarded. If the next node is in the same status as the selected node, thus it will be considered a reward for that action. In Table 5, the node D has the highest value of un and thus the allocation of reward or penalty to the actions selected by node A starts from this node. Since node D is in status “s” and its adjacent node in the table (i.e., node F) is also

in status “s”, then it is rewarded the actions selected by node D using Eqs. (6) and (7). Similarly, since node E is in status “a” and its adjacent node in the table (i.e., node B) is in status “s”, then it is penalized the actions selected by node D using Eqs. (8) and (9). The last node in the table that has generated less number by using relation (5), according to the tests carried out, it is better to be activated. In other words, if the last node in the table is in status “a”, then it is rewarded the selected action using Eqs. (6) and (7), otherwise this action is penalized by Eqs. (8) and (9). In Table 5, since node C is in status “a”, it is rewarded the selected action of node C by Eqs. (6) and (7). After the allocation of reward and penalty to the vectors within the received messages from the members, the cluster head firstly sends each message to the sender of that message so that the sender node update its vector accordingly. Then among these messages, it selects the message that leads to a high reward and broadcasts the final status of nodes to its members in the form of a message for optimal covering the cluster. Each member node after receiving this message from the cluster head, changes its status based on the vector included in the received message. With the passage of time and network lifetime, the parameters of nodes may be changed. For example, the node C which has the least amount of un in Table 6, may achieve a better un value than its neighbors in the next clustering round. Also, at the beginning of the formation of clusters, this action of automata to form the coverage set in the second phase can be repeated for n times.

Table 6. Final status of the nodes

Hostname	Status
D	s
F	s
E	a
B	s
A	a
C	a

3-3- Third Phase: Monitoring

After selecting the active nodes in the previous phase, in this phase the active nodes start to perform monitoring operations in their sensing areas until the next clustering round is began or the lifetime of cluster head node is terminated. Figure 5, shows the flowchart of the proposed method.

Algorithm 1 shows the pseudo-code of the proposed method.

Algorithm 1. Pseudo code of the proposed scheme

```

1. Start
2. Distribute sensor nodes randomly.
3. Each node waits for a certain time period  $T_n$  which is
   calculated via relation (3).
4. If node receives a message within  $T_n$  time period
5. Then
6.   Join a cluster.
7. Else
8.   Announce itself as a cluster head.
9.   Start the coverage algorithm.
10.  Do for each node
11.    If node has an automata action vector for its neighbors
12.    Then
13.      Randomly select an action for each neighbor.
14.    Else
15.      Create the action vector for each neighbor.
16.      Randomly select an action for each neighbor.
17.    End If
18.    Send these selected actions as a message to the CH.
19.  End Do
20.  Do these operations for each CH
21.    After receiving the message, CH does as follow:
22.    Sort the message by using relation (4).
23.    Assign reward or penalty by considering the values
       obtained from relation (4) and nodes status.
24.    Announce the status of the performed reward/penalty
25.    Broadcast the message having the maximum reward as
       the best message.
26.  End Do
27.  Do monitoring.
28.  If network lifetime has over
29.  Then
30.    End.
31.  Else
32.    If cluster lifetime has over or cluster head is dead
33.    Then
34.      Go step 4.
35.  Else
36.    Go step 25.
37.  End If
38.  End If
39.  End

```

3-4- Energy Consumption Model

If we assume that l is the number of transmitted bits, d is the distance between transmitter and receiver, E_{elec} is the energy required for transmission of a bit, ϵ_{amp} is the energy required to relay the transmitted bits and d_0 is the transmitting distance threshold for amplifier circuit, then the energy required for sending l bits of data is calculated according to Eq. (9):

$$E_{Tx}(K, d) = \begin{cases} k(E_{elec} + \epsilon_{fs}d^2) & d < d_0 \\ k(E_{elec} + \epsilon_{amp}d^4) & d \geq d_0 \end{cases} \quad (9)$$

In Eq. (9), the amount of d_0 is calculated via Eq. (10):

$$d_0 = \sqrt{\frac{\epsilon_{fs}}{\epsilon_{amp}}} \quad (10)$$

The energy required for receiving the packets of size l bits is also calculated by Eq. (11):

$$E_{Rx}(K, d) = k \cdot E_{elec} \quad (11)$$

4- Performance Evaluation

In this section, the performance of the proposed method is compared with that of C3 method in terms of the network coverage percentage and the average residual energy of sensor nodes in an environment of size 160×160 m² and with a uniform distribution of 100 to 300 sensor nodes. Both proposed and C3 [27] methods have been implemented via MATLAB simulation software. All simulations have been repeated for 30 independent runs. The simulation parameters are given in Table 7.

Table 7. Simulation parameters

Parameter	Value
Number of sensors	100 to 300
Sensing radius	10 m
Sending and receiving radius	20 m
Network area	160×160 m ²
Number of running	30
Position of the sink node	(80,80), (35,25), (130,65)
α, β	0.18, 0.55
E_{elec}	50 nj/bit
E_{DA}	5 nj/bit/packet
ϵ_{fs}	10 pj/bit/m ²
ϵ_{amp}	0.0013 Pj/bit/m ⁴
Initial energy	2 j
Packet size	4000 bits
Control packet size	200 bits

4-1 Comparison in Term of Network Coverage Percentage

In this section, the performance of the proposed method is compared with C3 [27] method in term of the network coverage percentage under the following three scenarios.

- **First Scenario:** sink is located in coordinate (80, 80) of the environment. Based on figure 7, with the increasing the number of nodes, the network coverage percentage generally increases, but the rate of increase in the proposed method is greater than C3 [27] method.

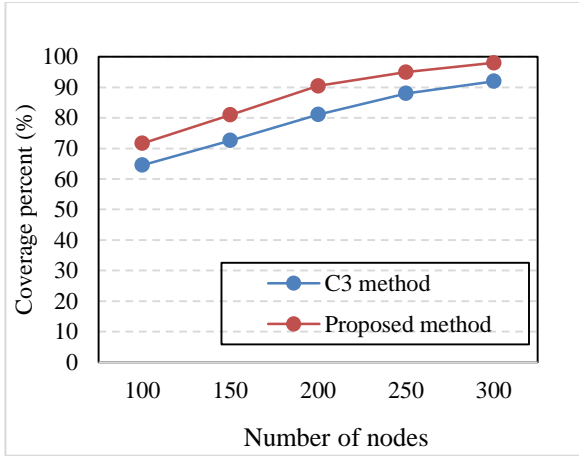


Fig. 5. Network coverage percentage in the first scenario

- **Second Scenario:** sink is located in coordinate (35, 25) of the environment. Based on figure 8, by increasing the number of nodes, the network coverage percentage also increases, but the rate of increase in the proposed method is more than C3 [27] method.

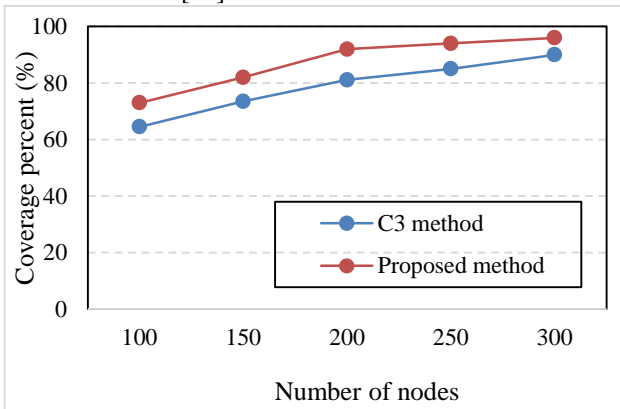


Fig. 6. Network coverage percentage in the second scenario

- **Third Scenario:** sink is located in coordinate (130, 65) of the environment. As shown in figure 9, with the increasing the number of nodes, the network coverage percentage increases, but the rate of increase in the proposed method is more than C3 [27] method. The simulation results of the three scenarios show that the proposed method performs better than C3 [27] method in term of the network coverage percentage.

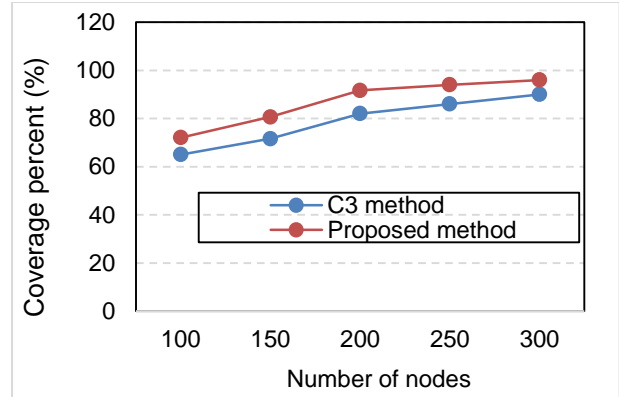


Fig. 7. Network coverage percentage in the third scenario

4-2 Comparing in Term of Average Residual Energy of Nodes

In this section, the efficiency of the proposed method is compared against C3 [27] method in term of the average residual energy of sensor nodes.

- **First Scenario:** sink is located in coordinate (80, 80) of the environment

According to figure 10, we can conclude that with the increasing the number of nodes, the average residual energy of network nodes generally increases, but the rate of increase in the proposed method is higher than C3 [27] method.

- **Second Scenario:** sink is located in coordinate (35, 25) of the environment. Based on figure 11, we can conclude that by increasing the number of nodes, the average residual energy of network nodes increases, but the rate of increase in the proposed method is more than C3 method.

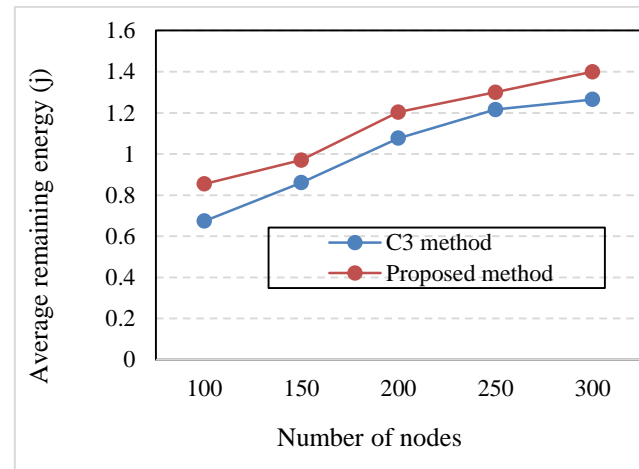


Fig. 8. Average residual energy of nodes in the first scenario

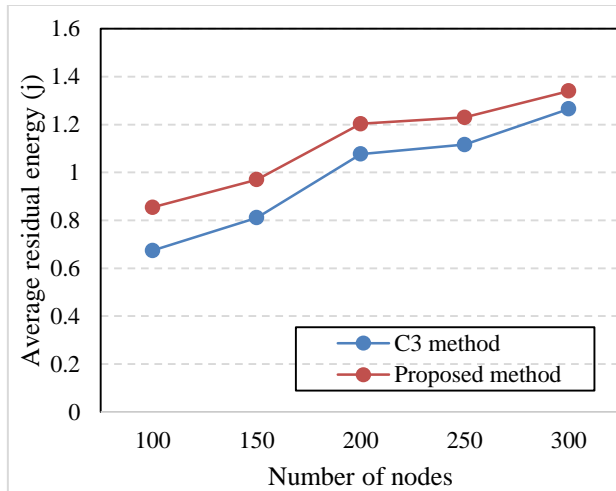


Fig. 9. Average residual energy of nodes in the second scenario

- **Third Scenario:** sink is located in coordinate (130, 65) of the environment. According to figure 12, we can conclude that with the increasing the number of nodes, the average residual energy of sensor nodes increases, but the rate of increase in the proposed method is greater than C3 [27] method. The simulation results of the three scenarios indicate that in the presence of different number of sensor nodes, the average residual energy of network nodes in the proposed method is higher than C3 method and as a result, the network lifetime of the proposed method is also higher than C3 method.

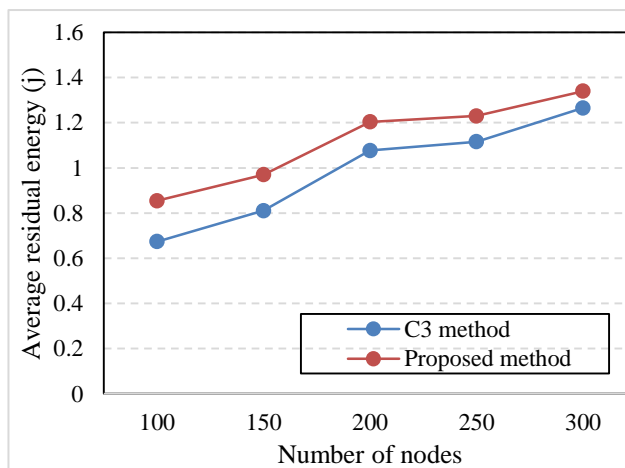


Fig. 10. Average residual energy of nodes

5- Conclusions

In this paper, a new method to improve the network coverage percentage and prolong the lifetime of wireless sensor networks was proposed. In the proposed method, after clustering the network, firstly inside the clusters

each node creates the action and probability vectors of learning automata for itself and its neighbors, then determines the status of itself and all its neighbors and finally sends them in the form of a message to the cluster head. Afterward, each cluster head starts to reward or penalize the vectors within the received messages and sends the results to the sender of each message for updating purposes. Thereafter, among the sent vectors, the cluster head node selects the best action vector and broadcasts it in the form of a message inside the cluster. Finally, each member changes its status in accordance with the vector included in the received message from the corresponding cluster head and the active sensor nodes perform environment monitoring operations. The environment monitoring operations are performed by the active nodes until the beginning of next clustering round or termination of the cluster head lifetime. The simulation results showed that the proposed method in terms of the network coverage percentage and the average residual energy of nodes performs better than C3 method. As a future work, we will perform the proposed scheme in other areas with holes or obstacles.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, vol. 38, no. 4, pp. 393-422, 2002.
- [2] D. KeyKhosravi, A. Ghaffari, A. Hosseinalipour, and B. A. Khasragi, "New Clustering Protocol to Decrease Probability Failure Nodes and Increasing the Lifetime in WSNs," *Int. J. Adv. Comp. Techn.*, vol. 2, no. 2, pp. 117-121, 2010.
- [3] A. Ghaffari, "Designing a wireless sensor network for ocean status notification system," *Indian Journal of Science and Technology*, vol. 7, no. 6, pp. 809-814, 2014.
- [4] A. Ghaffari, "Congestion control mechanisms in wireless sensor networks: A survey," *Journal of network and computer applications*, vol. 52, pp. 101-115, 2015.
- [5] A. Seyfollahi and A. Ghaffari, "A lightweight load balancing and route minimizing solution for routing protocol for low-power and lossy networks," *Computer Networks*, vol. 179, p. 107368, 2020.
- [6] A. Seyfollahi and A. Ghaffari, "Reliable data dissemination for the Internet of Things using Harris hawks optimization," *Peer-to-Peer Networking and Applications*, vol. 13, no. 6, pp. 1886-1902, 2020.
- [7] J. A. Torkestani, "An adaptive energy-efficient area coverage algorithm for wireless sensor networks," *Ad hoc networks*, vol. 11, no. 6, pp. 1655-1666, 2013.
- [8] S. M. Mohamed, H. S. Hamza, and I. A. Saroit, "Coverage in mobile wireless sensor networks (M-WSN): A survey," *Computer Communications*, vol. 110, pp. 133-150, 2017.
- [9] A. Ghaffari and A. Rahmani, "Fault tolerant model for data dissemination in wireless sensor networks," in *Information Technology, 2008. ITSIM 2008. International Symposium on, 2008*, vol. 4, pp. 1-8: IEEE.
- [10] H. D. Nikokheslat and A. Ghaffari, "Protocol for Controlling Congestion in Wireless Sensor Networks," *Wireless Personal Communications*, vol. 95, no. 3, pp. 3233-3251, 2017.
- [11] P. Natarajan and L. Parthiban, "k-coverage m-connected node placement using shuffled frog leaping: Nelder-Mead algorithm in WSN," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-16, 2020.
- [12] C. Shivalingegowda and P. Jayasree, "Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-14, 2020.
- [13] M. El-Hosseini, H. ZainEldin, H. Arafat, and M. Badawy, "A fire detection model based on power-aware scheduling for IoT-sensors in smart cities with partial coverage," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-20, 2020.
- [14] R. Yarinezhad and S. N. Hashemi, "A sensor deployment approach for target coverage problem in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, pp. 1-16, 2020.
- [15] A. Pakmehr and A. Ghaffari, "Coverage Improving with Energy Efficient in Wireless Sensor Networks," *Information Systems & Telecommunication*, p. 57.
- [16] M. A. Benatia, M. h. Sahnoun, D. Baudry, A. Louis, A. El-Hami, and B. Mazari, "Multi-Objective WSN Deployment Using Genetic Algorithms Under Cost, Coverage, and Connectivity Constraints," *Wireless Personal Communications*, vol. 94, no. 4, pp. 2739-2768, 2017.
- [17] A. Xenakis, F. Foukalas, G. Stamoulis, and I. Katsavounidis, "Topology control with coverage and lifetime optimization of wireless sensor networks with unequal energy distribution," *Computers & Electrical Engineering*, 2017.
- [18] J. Roselin, P. Latha, and S. Benitta, "Maximizing the wireless sensor networks lifetime through energy efficient connected coverage," *Ad Hoc Networks*, vol. 62, pp. 1-10, 2017.
- [19] D. Thomas, R. Shankaran, M. Orgun, and S. Mukhopadhyay, "A secure barrier coverage scheduling framework for WSN-based IoT applications," *IEEE Transactions on Green Communications and Networking*, 2021.
- [20] C. Shivalingegowda and P. Jayasree, "Hybrid gravitational search algorithm based model for optimizing coverage and connectivity in wireless sensor networks," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 2, pp. 2835-2848, 2021.
- [21] J. Yu, Y. Chen, L. Ma, B. Huang, and X. Cheng, "On connected target k-coverage in heterogeneous wireless sensor networks," *Sensors*, vol. 16, no. 1, p. 104, 2016.
- [22] N.-T. Le and Y. M. Jang, "Energy-efficient coverage guarantees scheduling and routing strategy for wireless sensor networks," *International Journal of Distributed Sensor Networks*, vol. 11, no. 8, p. 612383, 2015.
- [23] S. S. Dhillon, K. Chakrabarty, and S. S. Iyengar, "Sensor placement for grid coverage under imprecise detections," in *Information Fusion, 2002. Proceedings of the Fifth International Conference on, 2002*, vol. 2, pp. 1581-1587: IEEE.
- [24] K.-P. Shih, Y.-D. Chen, C.-W. Chiang, and B.-J. Liu, "A distributed active sensor selection scheme for wireless sensor networks," in *Computers and Communications, 2006. ISCC'06. Proceedings. 11th IEEE Symposium on, 2006*, pp. 923-928: IEEE.
- [25] S. Soro and W. B. Heinzelman, "Cluster head election techniques for coverage preservation in wireless sensor networks," *Ad Hoc Networks*, vol. 7, no. 5, pp. 955-972, 2009.
- [26] H. Mostafaei, A. Montieri, V. Persico, and A. Pescapé, "A sleep scheduling approach based on learning automata for WSN partial coverage," *Journal of Network and Computer Applications*, vol. 80, pp. 67-78, 2017.
- [27] M. Akhlaq, T. R. Sheltami, and E. M. Shakshuki, "C3: an energy-efficient protocol for coverage, connectivity and communication in WSNs," *Personal and Ubiquitous Computing*, vol. 18, no. 5, pp. 1117-1133, 2014.
- [28] Z. Sun, Y. Zhang, Y. Nie, W. Wei, J. Lloret, and H. Song, "CASMOc: a novel complex alliance strategy with multi-objective optimization of coverage in wireless sensor networks," *wireless Networks*, vol. 23, no. 4, pp. 1201-1222, 2017.
- [29] A. Khelil and R. Beghdad, "SPMI: Single Phase Multiple Initiator Protocol for Coverage in Wireless Sensor Networks," *Wireless Personal Communications*, vol. 96, no. 2, pp. 3159-3178, 2017.
- [30] T. G. Nguyen, C. So-In, N. G. Nguyen, and S. Phoemphon, "A novel energy-efficient clustering protocol with area

coverage awareness for wireless sensor networks," *Peer-to-Peer Networking and Applications*, vol. 10, no. 3, pp. 519-536, 2017.

- [31] Y. Hu, Y. Niu, J. Lam, and Z. Shu, "An energy-efficient adaptive overlapping clustering method for dynamic continuous monitoring in WSNs," *IEEE Sensors Journal*, vol. 17, no. 3, pp. 834-847, 2016.

Seyyed Keyvan Mousavi received the B.Sc. degree in Software Engineering from the Payame Noor University, Miandoab, Iran, in 2012, the M.Sc. degree in Software Engineering from the Islamic Azad University, Miandoab branch, in 2017, and Ph.D. in software engineering from Urmia branch in 2020. His research interests include Internet of things, Information security, and Cryptography. Currently his research work is based on Security and Privacy in IoT networks.

Ali Ghaffari received his BSc, MSc and Ph.D. degrees in computer engineering from the University of Tehran and IAU (Islamic Azad University), TEHRAN, IRAN in 1994, 2002 and 2011 respectively. He has served as a reviewer for *Applied Soft Computing*, *Ad Hoc networks*, *Future Generation Computer System (FGCS)*, Elsevier, *International Communication System (IJCS)*, *Journal of Ambient Intelligent and Humanized Computing (AIHC)*, *Wireless Networks*, *Wireless Personal Communication*. As an associate professor of computer engineering at Islamic Azad University, Tabriz branch, IRAN, his research interests are mainly in the field of software defined network (SDN), *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 100 international conference and reviewed journal papers.