

Target coverage heuristic based on learning automata in wireless sensor networks

ISSN 2043-6386

Received on 28th July 2017

Revised 20th November 2017

Accepted on 4th January 2018

E-First on 13th February 2018

doi: 10.1049/iet-wss.2017.0090

www.ietdl.org

Manju¹ ✉, Satish Chand², Bijender Kumar¹¹Department of Computer Engineering, Netaji Subhash Institute of Technology, Sector-3, Dwarka, New Delhi 110 078, India²School of Computer and System Sciences, Jawaharlal Nehru University, New Delhi 110 067, India

✉ E-mail: manju.nunia@gmail.com

Abstract: In wireless sensors networks, the sensor nodes are densely deployed. Owing to this excessive deployment of sensor nodes, each target is covered by multiple sensors at a time. To prolong the network lifetime, the authors can schedule the sensor activity in such a way that only a subset of sensor nodes, called cover set, is sufficient enough to cover all the targets. In this study, they propose an energy-efficient scheduling algorithm based on learning automata for target coverage problem. The learning automata-based technique helps a sensor node to select its appropriate state (either *active* or *sleep*). To prove the effectiveness of their proposed scheduling method, they conduct a detailed set of simulations and compare the performance of their algorithm with the existing algorithms.

1 Introduction

The wireless sensor networks (WSNs) have been receiving great attention in recent years. The basic structure of a WSN consists of huge number of low-power, low-cost, battery operated, multi-functional sensing devices called sensors. These sensors are placed on different locations inside a certain terrain to monitor the given area or the targets. A sensor is equipped with a small processor, transceivers, memory resource, actuators and a limited battery device [1]. There are certain issues while monitoring a given physical phenomenon/activity using these sensors. Utilising energy efficiently is one of the critical issues in the battery-powered WSNs as the sensor nodes' batteries are not reachable and sometimes not even replaceable in case of inaccessible terrain. There have been discussed several methods, which use the approaches such as energy-aware routing, energy-efficient data aggregation and dissemination, transmission power control and nodes activity scheduling, for using the network energy efficiently, so that the network lifetime can be prolonged. The energy of a sensor node can be used efficiently by keeping it in two modes of action namely, *active* and *sleep* states. In *sleep* mode, the sensor does not deplete its energy much; it simply listens the environment/targets for activities. In *active* state, the sensor monitors the environment/targets for their activities. Therefore, by scheduling the activity of each sensor in *active* and *sleep* states, the network lifetime can be enhanced. There is an important issue related to the coverage in WSNs, which deals with the ability to cover a certain area or a given set of targets. Depending on the coverage interest [2], the coverage problem may be classified as the area coverage or the target coverage problem.

In this paper, we address the target coverage problem [3–9] in which the sensor nodes are randomly deployed in a given fixed area consisting of the targets that monitor all that targets for the maximum possible duration. Generally, the given area is densely deployed to fulfil the coverage requirement. Therefore, in order to prolong the network lifetime, instead of activating all the sensor nodes simultaneously, it is more appropriate to organise these sensor nodes into subsets in such a way that each subset is capable of covering all the targets. We term these subsets as the cover sets. By activating one cover set at a time in a subsequent manner, the deployed network will reduce its energy consumption, and as a result it will be functional for longer period. This happens so because the sensors which are not part of the current cover set are no longer required to be *active*. Therefore, they can go to the *sleep*

state to save the energy. There are several methods for scheduling the sensor nodes for providing various node scheduling criteria to efficiently use the limited battery of sensor nodes.

Recently, researchers [3, 8, 10] have used the learning automaton to solve the target coverage problem in order to enhance the network lifetime. In learning automata-based scheduling mechanism, each sensor is assigned with a learning automaton that helps it to select an appropriate state (*active* or *sleep*) in order to maximise their battery uses. Work in [8] addresses the target coverage problem using the learning automata in which a sensor covering the maximum number of targets has more probability to be selected as a member of the current cover set. Since the sensor's coverage is fixed in a network, this criterion keeps selecting the same sensor in subsequent cover sets till its battery is not exhausted and this result in coverage holes in the given network. The coverage holes basically consist of those targets which become uncovered due to the energy exhaustion of the sensors covering them in spite of the fact that there are many sensors with good amount of energy in the network. In this work, we propose a new energy-efficient scheduling mechanism based on learning automata. To avoid early coverage holes in a network, we propose to select the sensors to be a member of cover sets that have high residual energy instead of having large coverage.

The rest of this paper is organised as follows. In Section 2, we provide the detailed work done on the energy-efficient target coverage problem. Section 3 gives the problem statement and some definitions to understand the target coverage problem. Section 4 discusses the learning automata theory and its application to coverage problems. In Section 5, we provide our proposed scheduling mechanism based on learning automata. Section 6 presents the simulation results, and finally Section 7 concludes the work and gives future directions.

2 Earlier works

In WSNs, the coverage problem aims to provide full coverage over a given area or set of targets in a physical phenomenon. The area coverage problem [11] refers to completely cover the given area, whereas the target coverage problem [8] refers to provide coverage of full target set in a fixed terrain. Since the scheduling algorithms can be an effective way for solving the target coverage problem, we review some of the important studies that are related to our work. Mostafaei and Meybodi [3] have provided a learning automaton-based heuristic solution for the target coverage problem

that generates the disjoint cover sets to find a near optimal solution. Mostafaei *et al.* [5–7] proposed a learning automata-based heuristic for partial coverage, where the objective is to minimise the selected sensors and maintain connectivity too. Pujari *et al.* [12] also provide a heuristic which generates disjoint cover set for target coverage problem. Owing to the restricted participation of sensor nodes, the network lifetime gained by such disjoint techniques may not be optimal. Therefore, to further improve the network lifetime, Bajaj and Manju [4] and Manju and Pujari [9] improve the network lifetime by using non-disjoint cover set, where a sensor can be part of more than one cover set. Work in [9] improves the network lifetime by using energy-based heuristic, where non-disjoint cover sets are formed with high-energy sensors.

Many centralised [10, 13] and decentralised algorithms [11, 14, 15] are discussed in the literature for solving the target coverage problem. In a centralised method, a special node (called sink node) has the information of the complete network topology and all the decisions regarding every sensor's state (*active* or *sleep*) are decided by that node only. In case of a decentralised method, there is as such no special node to coordinate the whole functionality of the networks, instead each node has an almost equal role to play in the network. Every node decides its state by its own based on its surrounding nodes' state. In decentralised algorithms, a sensor chooses its state to be active only if its covered region is not falling within the sensing range of its neighbouring nodes.

Owing to the presence of faulty nodes in a dense sensor network, in some crucial applications, single coverage is not sufficient to fulfil the quality of service requirements. Therefore, to address this issue, the target Q-coverage problem is introduced in [16, 17], where targets are said to be covered if each target is covered by at least Q-number of sensors. Furthermore, work in [18, 19] discusses variant called priority-based target coverage, where each target may have different coverage requirement. To maximise battery uses, multiple sensing range [20] and random battery life [15] are also proven to be an efficient way to maximise network lifetime.

The heuristics discussed so far for target coverage problem are greedy in nature. They all try to reach a near optimal solution. To improve the performance of generic mechanism, learning automata can be a useful tool. In this paper, we propose a scheduling algorithm based on learning automata to solve the target coverage problem. In our proposal, we try to give preference to those sensors, which have the highest ratio of the remaining energy to the total network energy available at that time.

3 Problem statement

Here, we consider the target coverage problem, where n number of sensors are randomly deployed in order to monitor a set of m targets with a fixed location within a predefined area of size $m \times m$. Each sensor has its sensing range as R and energy level as E_{initial} . We assume homogenous WSNs, where all the sensors are assigned with same sensing range and uniform energy level. A target is covered if it lies within the sensing range of at least one sensor. A sensor can be in either *active* or *sleep* state. To save energy, the sensors go to *sleep* state while the *active* sensors monitor the targets. Now, we provide some required notations which are followed in the subsequent parts of this paper:

n = number of sensors,
 m = number of targets,
 $S = \{s_1, \dots, s_n\}$ be the set of all sensors,
 $T = \{t_1, \dots, t_m\}$ be the set of all targets,
 s_i = the i th sensors, $1 \leq i \leq n$,
 t_j = the j th target, $1 \leq j \leq m$,
 E_{rem} = residual energy of sensor s_i ,
 E_{total} = sum of all the sensor's remaining energy,
 E_{initial} : initial energy assigned to each $s_i \in S$, and
 R_i = sensing range of sensor s_i

Here, our main focus is on designing an energy-efficient node scheduling mechanism. We do not emphasise the selection of any protocol for data gathering and forwarding.

3.1 Definition 1: target coverage problem

Given a WSN consisting of n sensors and m targets all with the fixed known location, where a target is said to be covered if it is covered by at least one sensor in the given network. To maximise the network lifetime, schedule the sensors activity and generate maximum possible cover sets such that each cover set is capable of completely covering all the targets.

Let C_1, \dots, C_p be the set of all feasible cover sets for a given network, and X_1, \dots, X_p be their respective cover sets working times in the network. For each sensor s_i , $i = 1, \dots, n$ and for each cover set C_k , $k = 1, \dots, p$, we set a binary variable a_{ij} equal to 1, if the sensor s_i is a part of some cover set C_k ; otherwise, a_{ij} is 0. We may write the target coverage problem for maximising total network lifetime as an optimisation problem which is given as follows:

$$\text{maximise } \sum_{k=1}^p X_k \quad (1)$$

$$\text{s.t. } \sum_{k=1}^p a_{ik} X_k \leq E_{\text{initial}} \forall i = 1, \dots, n \quad (2)$$

$$X_k \geq 0 \forall k = 1, \dots, p \quad (3)$$

The total network lifetime can be calculated from (1). The constraints in (2) ensure that none of the sensors should be used beyond its initial energy assigned to it and constraint (3) ensures that each cover set should be meaningful.

3.2 Definition 2: critical target

A target $t_j \in T$ is *critical* if sum of the residual energies of all the sensors covering t_j is minimum among all the targets in set T .

3.3 Definition 3: critical sensor

A sensor $s_i \in S$ is *critical* if it covers at least one *critical target*.

3.4 Definition 4: working time of a cover set

The total activation time duration $W(C_k)$ for a cover set should not exceed the smallest available residual energy of any sensor which is part of that cover set. Thus

$$W(C_k) = \text{Min}_{s_i \in C_k} (E_{\text{rem}})$$

3.5 Definition 5: cover set energy

The total cover set energy of a cover set is the sum of the residual energies of those sensors belonging to that cover set. Thus

$$\text{cover set energy} = \sum_{s_i \in C_k} (E_{\text{rem}})$$

3.6 Definition 6: network lifetime

The total network lifetime can be defined as the sum of the working time of all the generated cover sets

$$\text{Network_Lifetime} = \sum_{k=1}^{|C|} (C_k)$$

To understand the basic functionality of learning automata, we briefly review the learning automata.

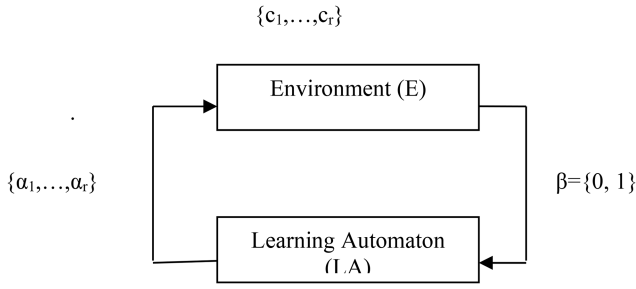


Fig. 1 Learning-environment

4 Learning automata

An ‘automaton’ is a self-operating mechanism to achieve certain goals by responding to a sequence of instructions (rules). These rules may be predefined or adapted to the working environment. The term ‘learning’ refers the act of gaining knowledge for modifying one’s behaviour toward certain action in a given environment. The automaton has a collection of actions and corresponding to each action, the environment can respond either in favour or against with certain probability [21]. Thus, an automaton acts as a decision maker which follows certain rules to reach the best output. To do so, the learning automata (LP) continues to keep interacting with the environment to make decision about best suitable action.

The LA environment, E , refers a medium in which LA functions are best described as a triplet $E = \{A, C, B\}$, where A , B and C are defined as follows [21]:

$A = \{a_1, a_2, \dots, a_r\}$ represents a finite set of inputs,

$B = \{b_1, b_2, \dots, b_r\}$ represents a finite set of outputs, and

$C = \{c_1, c_2, \dots, c_r\}$ represents a set of penalty probabilities such that the element $c_i \in C$ corresponds to the action a_i .

Each automaton is assigned a set of actions $\{a_1, \dots, a_r\}$ to choose from. The automaton chooses one of its action say a_i with the penalty probability c_i as initial input. Each LA is assigned with an action probability vector corresponds to its action set. The action probability prompts the LA with reward or penalty. The information provided about reward–penalty can help the LA to decide its future action in the given environment. Fig. 1 [3] shows the learning functionality of an LA.

The LA learns to choose the optimal action in the given environment E as shown in Fig. 1. In general, the LA aims to choose the optimal actions such that the average reward received so far in the environment can be maximised [2].

Generally, LA is categorised as: fixed structure LA or variable structure LA. In this paper, we consider only variable structure LA. In variable structure LA, the action probability vector is not fixed, instead, the action probabilities are updated after each instance. Any action selection from action pool is depending on the action probability vector which is updated based on the reward–penalty operation. Now, we define the variable structure LA as a quadruple $\{a, \beta, P, H\}$, where $a = \{a_1, \dots, a_n\}$, action set of automata, $\beta = \{\beta_1, \dots, \beta_n\}$, set of input, $P = \{P_1, \dots, P_n\}$, action probability set and H is the learning algorithm of the automata. The complete process performed by an automaton is as follows: the automaton randomly selects an action a_i based on the action probability p to perform in the given environment E . Once the reinforcement signal is received, the automaton updates its action probability vector as per the following (4) for favourable response and (5) in case of unfavourable response [8]

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1 - a)P_j(n) \quad \forall j, j \neq i \\ p_i(n+1) &= (1 - b)p_i(n) \end{aligned} \quad (4)$$

$$p_i(n+1) = \frac{b}{r-1} + (1-b)p_i(n) \quad \forall j, j \neq i \quad (5)$$

In both (4) and (5), a and b denotes the reward and penalty parameters, respectively. On the basis of the value taken for a and b , the learning algorithm can be grouped as follows: if $a = b$, then the recurrences (4) and (5) are called as linear reward–penalty ($L_R - p$) process, if $a \gg b$, then, these equations are called linear reward– ϵ penalty ($L_{R-\epsilon p}$) process, and if $b = 0$, then both (4) and (5) called as linear reward–inaction (L_{R-i}) methods [21].

The LA performs properly in a dynamic environment, where the network efficiency increases with the change in environment. The LA is also a suitable tool for many optimisation problems, where there is a high level of uncertainty. LA is also useful in solving many hard-to-solve problems [2]. Recently, many LA-based protocols have been addressed to improve the WSNs performance [3, 11, 22]. Mostafaei and Meybodi [3] discuss the target coverage problem and address a learning automata-based scheduling algorithm to extend the network lifetime. Another learning automaton-based method is discussed by Mostafaei *et al.* [11] to prolong the network lifetime for area coverage problem in WSNs.

As mentioned in above section, LA is a useful tool for solving many hard-to-solve optimisation problems, where the environment changes with time. As target coverage is also proven to be non-deterministic polynomial (NP)-complete [5]; therefore, the learning automaton-based approach may provide a better solution for the target coverage problem. In the next section, we propose a new energy-efficient learning automaton-based heuristic for target coverage problem in WSN.

5 Proposed algorithm

We propose an algorithm based on learning automata to maximise the network lifetime for target coverage problem in WSNs. Our proposed algorithm consists of four phases as follows: (a) network-setup phase, (b) learning phase, (c) monitoring phase, and finally (d) update phase. In the subsequent part of this section, we give a detailed description on each of the phases.

5.1 Network-setup phase

Our network consists of n sensors each having fixed sensing range R deployed randomly in a given area of size $M \times M$. There are m targets deployed in the close proximity of these n sensors. All the sensors and targets are fixed after deployment in the network. Thus, each sensor knows its targets covered in its sensing range.

In the network, each sensor s_i has its associated LA_i using which the sensor s_i determines the optimal action at any point of time in the given environment E . The learning automaton of every node can have two actions: *ACTIVE* or *INACTIVE*. We consider the variable structure learning automata, where the action probability of LA_i keeps changing in every iteration. Initially, both the actions have an equal probability of 0.5. Each LA_i updates its action probabilities vector using the linear reward–inaction (L_{R-i}) scheme. After completing this phase, we perform the learning phase as follows.

5.2 Learning phase

This phase is the backbone of our proposed algorithm. In this phase, each learning automata LA_i (it corresponds to node s_i) decides the possible action to be taken (*ACTIVE* or *INACTIVE*) in the current round. Initially, we assume that each LA is in an *INACTIVE* state. The action probabilities of all LA_i are updated in such a way that the sensors having higher probability given in (6) are more likely to be in *ACTIVE* state. We consider energy as a prime parameter to prolong the network lifetime because the sensors with high residual energy will be alive for a longer time. Therefore, we give preference to such high residual energy sensors, so that low-energy sensor can be also alive for a longer time which results in prolonged network lifetime. Therefore, automaton LA_i configures the choice probability for sensor s_j , denoted by p_{ij} , which may also be referred as energy ratio as follows:

$$P_j^i = \frac{E_{rem}(j)}{\sum_{k=1}^n E_{rem}(k)} \quad (6)$$

Here, E_{rem} denotes the residual energy of sensor s_j and $\sum_{k=1}^n E_{rem}(k)$ denotes the sum of the residual energy of all the sensors. Thus, it is evident from (6) that the sensors with more residual energy are likely to be in *ACTIVE* state.

5.3 Monitoring phase

In this phase, we form cover sets to monitor all the targets in the given network. Once learning phase is over, each node in the network selects its action with the higher probability. The action of LA_i which has higher action probability is to be chosen for sensor s_i during the current iteration. Once all the LA_i decides their optimal actions, then we keep selecting those sensors whose *ACTIVE* action probability given in (6) is more than *INACTIVE* action probability to form cover set. The network is going to exhaust when even a single target becomes uncovered. In an energy-constrained network, the critical target(s) is the one which becomes uncovered first. Therefore, to prolong the network lifetime, we try to keep alive those sensors (*critical* sensors) which cover critical target(s). To do so, we follow the *Rule 1* given below.

Rule 1: We first identify all the critical sensors present in the network. Then, we select only one critical sensor which (a) has *ACTIVE* action probability more than *INACTIVE* action probability, (b) whose energy ratio (6) is more and (c) which covers maximum uncovered critical targets in the networks.

We follow *Rule 1* to enhance the network lifetime significantly by keeping alive critical sensors for a longer period. Since the network is densely deployed, therefore, targets are covered by multiple sensors. To avoid redundant target coverage, we use the following *Rule 2* to further enhance the total network lifetime.

Rule 2: Once a cover set is formed in the monitoring phase, we remove redundant sensors by changing their action from *ACTIVE* to *INACTIVE*. After removing extra sensors from cover set generated so far, we find a minimal cover set.

After minimising the cover set, the working time is assigned to the generated cover set. Now, we either reward or penalise the actions chosen by all LA in above cover set. We follow two methods to decide whether to reward or penalise the action probability vector of these LA s which are *ACTIVE* in the current cover set as given below:

Method 1: The cardinality of the chosen cover set is compared with a dynamic threshold T_h after each iteration. Here, T_h is the cardinality of smallest cover set generated till the previous iteration. If the cardinality of the current cover set is smaller than T_h , then the action probability vector corresponding to those sensors which are part of the cover set are rewarded using (4); otherwise, penalised using (5).

Method 2: Here, the action probability vector of each LA_i which is *ACTIVE* in the current cover set is rewarded if the sum of total energy of cover set is smaller than the threshold energy E_{th} . Here, E_{th} is the dynamic threshold energy which is equal to the smallest cover set energy till the previous iteration. By using either method of rewarding/penalising the action probability, we complete the monitoring phase.

5.4 Update phase

We now update the residual energy of those sensors, which are part of cover set by reducing it with the amount equals to working time assigned to the cover set. After updating the energy of these sensors, we start the next iteration that consists of second, third and fourth phases to generate next cover set. This process is repeated till all the possible cover sets have been formed. We write this complete process in an algorithm form as given in (Fig. 2). To remove the coverage redundancy, we follow *procedure Trim* to minimise each generated cover set in (Fig. 2).

Input: S:set of n sensors
T:set of m targets
W:working time of cover set
 E_i : initial energy of each $s_i \in S$
 LA_i : Each node s_i is assigned with learning automaton LA_i
 α, β : Learning parameter

Output: Network_lifetime

```

01. begin:
02. Let  $E_{rem}$  denotes the remaining energy of sensor s
03. Let  $E_{th}$  denoted dynamic energy threshold
    //Network-Setup phase
04. for each  $s_i \in S$ 
05.      $s_i$ .sense_environment()
06.      $s_i$ .find_sensor_coverage()
07. end for
08. while (network exhausted)
    //Learning phase
09.     for each  $s_i \in S$ 
10.         for each action of  $s_i$ 
11.             action_probability=0.5
12.         end for
13.     end for
14.     for each  $s_i \in S$ 
15.         calculate choice probability=  $\frac{E_{rem}(i)}{\sum_{k=1}^n E_{rem}(k)}$ 
16.     end for
    //Monitoring Phase
17.     for each  $s_i \in S$ 
18.         select_best_action()
19.     end for
20. repeat
21.     select  $s_i \in S$  whose  $LA_i = ACTIVE$  with highest choice probability
22. until Cover-Set formed (using Rule 1,2)
23. Minimize generated Cover_Set (using procedure Trim)
24. calculate Cover_Set energy ( $E_{cover\_set}$ )
25. for each  $s_i \in Cover\_Set$ 
26.     if  $E_{cover\_set} < E_{th}$ 
27.         reward chosen  $LA_i$ 
28.          $E_{th} = E_{cover\_set}$ 
29.     else
30.         penalize chosen  $LA_i$ 
31.     end if
32. end for
33. Network_Lifetime=Network_Lifetime+W
    //Update phase
34. for each  $s_i \in Cover\_Set$ 
35.      $E_i = E_i - W$ 
36. end for
37. end while
38. return Network_Lifetime

```

Fig. 2 Algorithm 1: Learning automata-based heuristic for target coverage problem

Line 2–3 initialises the remaining energy of all the sensors and the energy threshold, respectively. At Line 4, the first phase, i.e. network-setup phase starts. For each sensor s_i , Line 5–6 senses the given simulation environment and finds individual sensors' target

Input: S:set of all n Sensors
 C_p :Cover_set
Output: min_ C_p
1. *initialize* min_ $C_p = \emptyset$
2. *for* $i = \text{length}(C_p)$ down to 1 *do*
 // $s_i.C_p$ is i^{th} sensor in C_p
3. *if* $C - (s_i.C_p)$ still an Cover_set then
4. $C_p = C_p - s_i.C_p$
 // ignore $s_i.C_p$
5. *else*
6. min_ $C_p = \min_C_p \cup s_i.C_p$
7. *end if*
8. *end for*
9. *return* min_ C_p

Fig. 3 Trim: Minimise the Cover_set (C_p)

Table 1 Simulation parameters

Parameter	Value
network area ($M \times M$)	$(200 \times 200)M^2$
sensors	50–250
targets	20–100
sensing range	100M–500M
initial Energy of sensor	1 unit
cover set working time	0.5 time unit

coverage. At Line 8, the next phase, i.e. learning phase starts where each sensor is assigned with action probability (Line 9–13). Then, for each sensor, the choice probability is calculated (Line 14–16) based on (6). Then, monitoring phase starts (at Line 17), where for each sensor, the best action (action with higher probability) among Active or Inactive is chosen (Line 18–20). Once its done, then, the cover set is generated (Line 21–22) by selecting those sensors, whose state is ACTIVE and which has choice probability. The generated cover set may not be minimal; therefore, we minimise it by removing redundant sensors (Line 23). After that, the working time is assigned (Line 24) to the cover set formed in above step. Once the cover set is formed, then the action probabilities of those sensors which are part of the above-said cover set are either rewarded or penalised (Line 25–32). Finally, update phase will start where the remaining energies of those sensors which are part of the current cover set are updated (Line 34–36) (Fig. 3).

6 Simulation results

In this section, we evaluate the performance of our method using simulations in MATLAB (R2009) on a core i3 processor with 2.10 GHz processor and 4 GB random access memory system. For simulation purpose, we consider a fixed (two-dimensional) sensing area of size $200M \times 200M$ in which the sensor nodes are randomly deployed. A fixed number of targets are also deployed randomly within the area given. We consider homogenous sensor network, where sensors having an equal amount of energy and fixed sensing range. To study the effect of the node density on the performance of our algorithm, we vary sensors from 50 to 250 and targets from 20 to 100. We vary the sensing range from 100 to 500 m and take fixed W as 0.5 units. All simulations are run 50 times and then the average network lifetime is calculated. In all the algorithms, learning automata follow L_{R-1} scheme to update action probability vector (Table 1).

6.1 Impact of sensors and targets on network lifetime

Here, we want to show how the network lifetime changes when there is an increase in deployed sensor nodes and targets alternatively. To perform this task, we take fixed number of targets as 25, the number of sensors from 50 to 250, each of fixed sensing range R as 100 m. There is a significant increase in the network life by increasing the number of sensors as shown in Fig. 4a. This

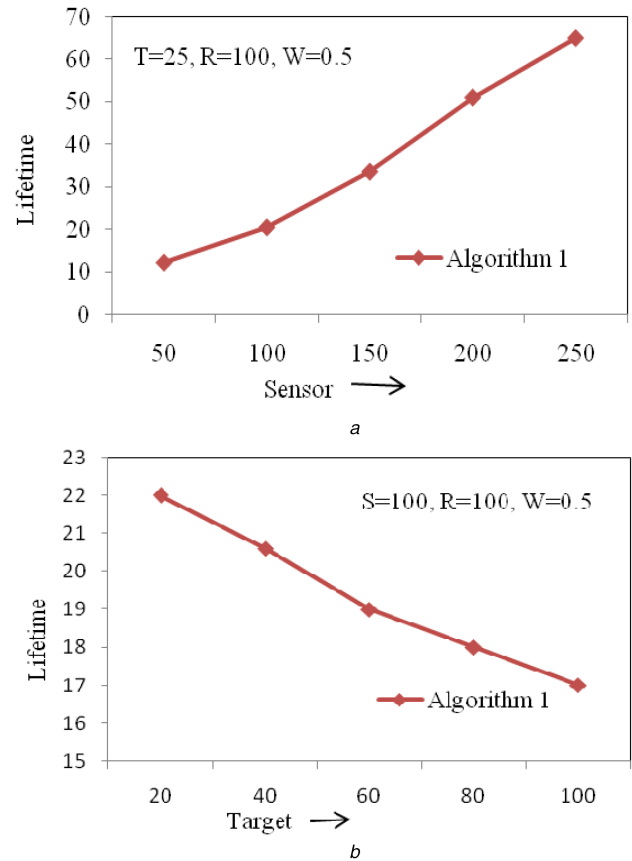


Fig. 4 Significant increase in the network life by increasing the number of sensors

(a) Network lifetime achieved by the proposed method (Fig. 2) with varying sensors, (b) Network lifetime achieved by the proposed method (Fig. 2) with varying targets

happens due to the fact that the network size is fixed and increasing the number of sensors, the individual target coverage increases because now a target is covered by more number of sensors, thus, resulting in improved network lifetime.

To understand the effect of the number of targets on the network lifetime, we consider the fixed number of sensors as 100 each having fixed sensing range as 100 m and the targets are varied from 20 to 100. The network lifetime decreases by increasing the number of targets as evident from Fig. 4b. It is due to the fact that increasing the number of targets, more sensors will be required to cover these targets and hence there will be more sensors in the ACTIVE state, resulting in the decreased network lifetime.

6.2 Impact of sensing range on network lifetime

Here, we study the effect of sensing range on the network lifetime. We take a fixed number of sensors as 150, a fixed number of targets as 25 and increase the sensing range from 100 to 500 m. Increasing the sensing range of the sensors increases the network lifetime as shown in Fig. 5. This is due to the fact that increasing the sensing range of a sensor; the sensor can cover more targets. Thus, less number of sensors is required to cover the targets, resulting in less energy consumption which in turn increases the network lifetime.

6.3 Number of alive nodes

Furthermore, we also simulated to see the statistics of a number of alive nodes in the networks. To do that, we take fixed 250 sensors and 50 targets and sensing area of $200M \times 200M$. The following Fig. 6 shows the results for above-said scenario.

6.4 Impact of working time on network lifetime

Now, we show the effect of working time of a cover set on the network lifetime. The choice/selection of working time for a cover

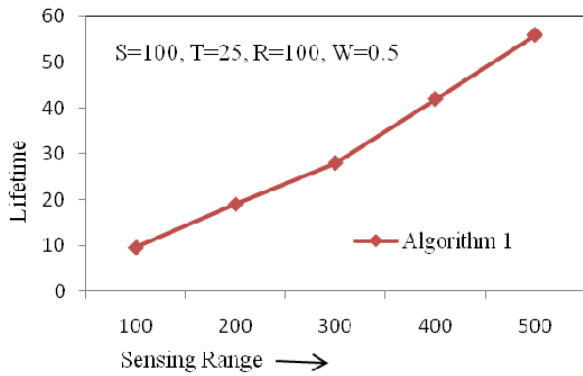


Fig. 5 Network lifetime achieved by the proposed method (Fig. 2) with varying sensing range

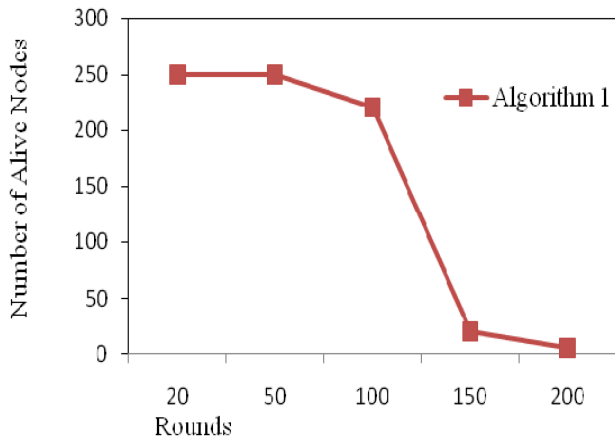


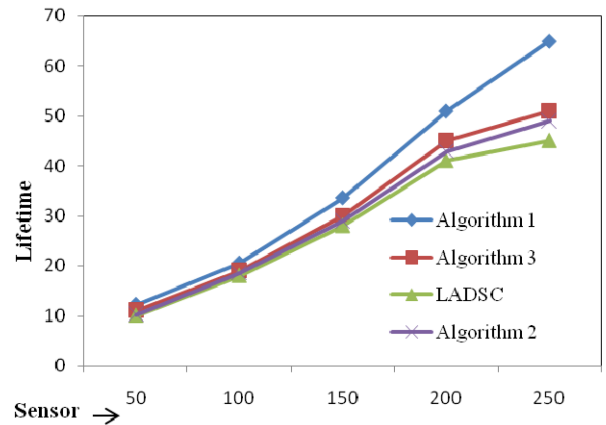
Fig. 6 Number of alive nodes over the time

set plays an important role as the network lifetime is a summation of all the cover sets working time. To show the effect of working time of a cover set on the network lifetime, we consider fixed set of sensors as 100 each having fixed sensing range R as 100 m. We have evaluated the results for a different number of targets (i.e. 20, 30, 40, ..., 90). We have varied the working time (W) of a cover set W from 0.01 to 1 as shown in Table 2. As depicted in Table 2, we have obtained different network lifetime at a different value of W . This happens because W is the granularity parameter and the sensors' energy is divided according to the working time which results in different network lifetime each time.

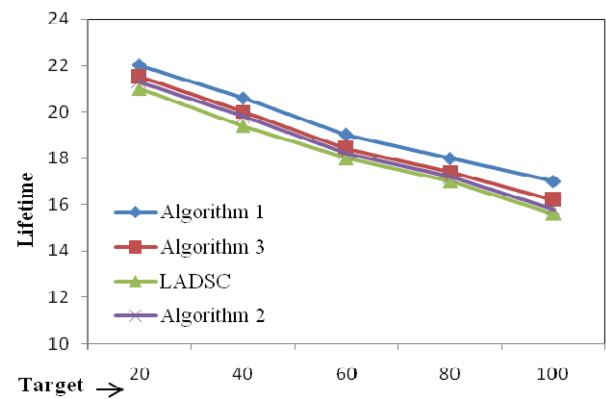
6.5 Fig. 2 versus pervious works

Here, we compare the network lifetime obtained using our algorithm [see Algorithm 1 (Fig. 2)] with learning automata based disjoint set cover (LADSC) [3], the algorithm in [8] and Algorithm 2 [23]. In [8], there have been given three algorithms for solving target coverage problem and the third algorithm provides the best performance; therefore, we have considered the third algorithm (see Algorithm 3) in [8].

LADSC [3]: To find network lifetime, while generating a cover set, LADSC randomly chooses actions (Active or Asleep) for all sensors. Then, LADSC keeps selecting sensors till cover is formed.



a



b

Fig. 7 Performance of all algorithms

(a) Network lifetime achieved by the proposed method (Fig. 2), LADSC, Algorithm 2 and Algorithm 3, (b) Network lifetime achieved by the proposed method (Fig. 2), LADSC, Algorithm 2 and Algorithm 3

The action chosen by sensors is penalised if targets covered a sensor are already covered by its neighbouring sensors those are also active; otherwise, selected action for the sensor is rewarded.

Algorithm 3 [8]: To generate a cover set, Algorithm 3 assigns action probabilities to all sensors based on their coverage. The sensor which is covering the maximum number of targets has more probability to be selected as a member of the current cover set. Algorithm 3 generates cover set by selecting these highest probability sensors. Chosen action is rewarded if cover set cardinality is less than the threshold T_h [8]; otherwise penalised.

Algorithm 2 [23]: Greedy approach is used by Algorithm 2 to generate cover set by selecting those sensors which are covering more uncovered targets. Once the cover set is formed, the action probability vector is updated based on several pruning rules [23] to maximise the network lifetime.

Fig. 7a shows the performance of all algorithms with respect to a number of sensors and Fig. 7b shows the performance with respect to numbers of targets. As evident from Figs. 7a and b, our method outperforms the LADSC, Algorithm 2 and Algorithm 3.

Table 2 Network lifetime achieved by proposed method (Fig. 2) with varying cover set working time

Targets	$W=1$	$W=0.5$	$W=0.25$	$W=0.025$	$W=0.01$
20	2.27	2.31	2.32	2.34	2.34
30	2.04	2.07	2.09	2.10	2.10
40	1.99	2.02	2.02	2.03	2.03
50	1.98	2.00	2.00	2.01	2.01
60	1.97	1.99	2.00	2.00	2.00
70	1.97	1.98	1.99	1.99	2.00
80	1.98	2.00	2.01	2.02	2.02
90	1.92	1.95	1.96	1.98	1.98

7 Conclusion

In this paper, we have discussed an energy-efficient algorithm for target coverage problem based on learning automata as it has been proved to be a useful tool for solving optimisation problems. We have evaluated the performance of our algorithm by considering a varying number of sensors, targets, sensing range and working time. In all scenarios, our method has provided better network lifetime as compared with the existing methods. We have considered only the regular target coverage problem in which learning automaton-based method has been used. Our method can be applied to solving many variants of the same problem such as target Q-coverage, target connected coverage and α -coverage problems, where LA-based approach can be used to maximise the network lifetime.

8 Acknowledgment

This work was supported by UPE-II, Jawaharlal Nehru University, India.

9 References

- [1] Jin, R., Che, Z., Wang, Z., *et al.*: 'Battery optimal scheduling based on energy balance in wireless sensor networks', *IET Wirel. Sens. Syst.*, 2015, **5**, (6), pp. 277–282
- [2] Esnaashari, M., Meybodi, M.R.: 'Learning automata based scheduling solution to the dynamic point coverage problem in wireless sensor networks', *Comput. Netw.*, 2010, **54**, (14), pp. 2410–2438
- [3] Mostafaei, H., Meybodi, M.R.: 'Maximizing lifetime of target coverage in wireless sensor networks using learning automata', *Wirel. Pers. Commun.*, 2013, **71**, (2), pp. 1461–1477
- [4] Bajaj, D., Manju, : 'Maximum coverage heuristic (MCH) for target coverage problem in WSN'. Int. Advance Computing Conf., 2014, pp. 300–305
- [5] Mostafaei, H., Montierib, A., Persicoc, V., *et al.*: 'A sleep scheduling approach based on learning automata for WSN partial coverage', *J. Netw. Comput. Appl.*, 2017, **80**, pp. 67–78
- [6] Mostafaei, H., Montieri, A., Persico, V., *et al.*: 'An efficient partial coverage algorithm for wireless sensor networks'. IEEE Symp. Computers and Communication (ISCC), Italy, 2016, pp. 1–6
- [7] Mostafaei, H., Chowdhury, M.U., Islam, R., *et al.*: 'Connected P-percent coverage in wireless sensor networks based on degree constraint dominating set approach'. Proc. 18th ACM Int. Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems, Mexico, November 2015, pp. 157–160
- [8] Mohamadi, H., Ismail, A.S., Salleh, S.: 'Solving target coverage problem using cover sets in wireless sensor networks based on learning automata', *Wirel. Pers. Commun.*, 2014, **75**, pp. 447–463
- [9] Manju, ., Pujari, A.K.: 'High-energy-first (HEF) heuristic for energy efficient target coverage problem', *Int. J. Ad hoc Sens. Ubiquitous Comput.*, 2011, **2**, (1), pp. 45–58
- [10] Salleh, S., Marouf, S.: 'A learning automata-based solution to the target coverage problem in wireless sensor networks'. ACM Int. Conf. Proceeding Series, 2013, pp. 185–190
- [11] Mostafaei, H., Meybodi, M., Esnaashari, M.: 'A learning automata based area coverage algorithm for wireless sensor networks', *J. Electron. Sci. Technol.*, 2010, **8**, (3), pp. 200–205
- [12] Pujari, A.K., Mini, S., Padhi, T., *et al.*: 'Polyhedral approach for lifetime maximization of target coverage problem'. Int. Conf. Distrib. Comput. Netw., 2015, **14**, 1, 14:8
- [13] Singh, S., Chand, S., Kumar, R., *et al.*: 'NEECP: a novel energy efficient clustering protocol for prolonging lifetime of WSNs', *IET Wirel. Sens. Syst.*, 2016, **6**, (5), pp. 151–157
- [14] Mostafaei, H., Meybodi, M., Esnaashari, M.: 'EEMLA: energy efficient monitoring of wireless sensor network with learning automata'. Int. Conf. Signal Acquisition and Processing, 2010, pp. 107–111
- [15] Singh, S., Chand, S., Kumar, B.: 'Heterogeneous HEED protocol for wireless sensor networks', in 'Wireless personal communications' (Springer Science and Business Media, 2014), **77**, (3), pp. 2117–2139
- [16] Mini, S., Udgata, S.K., Sabat, S.L.: 'Sensor deployment and scheduling for target coverage problem in wireless sensor networks', *IEEE Sens. J.*, 2014, **14**, (3), pp. 636–644
- [17] Chaudhary, M., Pujari, A.K.: 'Q-coverage problem in wireless sensor networks'. Int. Conf. Distrib. Comput. Netw., 2009, pp. 325–330
- [18] Salleh, S., Marouf, S., Mohammadi, H.: 'A new learning automata-based algorithm to the priority-based target coverage problem in directional sensor networks', in (Eds.): 'Lecture notes of the institute for computer sciences, social-informatics and telecommunications engineering', 2015, **141**, pp. 219–229
- [19] Mohamadi, H., Salleh, S., Ismail, A.S.: 'A learning automata-based solution to the priority-based target coverage problem in directional sensor networks', *Wirel. Pers. Commun.*, 2014, **79**, pp. 2323–2338
- [20] Mohamadi, H., Salleh, S., Razali, M.N., *et al.*: 'A new learning automata-based approach for maximizing network lifetime in wireless sensor networks with adjustable sensing ranges', *Neurocomputing*, 2015, **153**, pp. 11–19
- [21] Narendra, K.S., Thathachar, M.A.L.: 'Learning automata: an introduction' (Prentice-Hall, Englewood Cliffs, 1989)
- [22] Mostafaei, H., Esnaashari, M., Meybodi, M.R.: 'A coverage monitoring algorithm based on learning automata for wireless sensor networks', *App. Math. Inf. Sci.*, 2015, **9**, (3), pp. 1317–1325
- [23] Mohamadi, H., Ismail, A., Salleh, S., *et al.*: 'Learning automata-based algorithms for finding cover sets in wireless sensor networks', *J. Supercomput.* (in press), 2013, **66**, (3), pp. 1533–1552