A New Learning Automata-based Algorithm to Solve the Target k-coverage Problem in Wireless Sensor Networks

Leila Ajam[1], Ali Nodehi[2*], Hosein Mohamadi[3]

Abstract– Recently, a number of algorithms have been proposed to solve the target coverage problem in wireless sensor networks (WSNs). Conventionally, it is assumed that only a single sensor is sufficient for covering a target; though, in real situations, more than one sensor may be required for this purpose. This problem is known as k-coverage problem that its NP-completeness has been already proved. To solve the problem, this paper proposes a learning-automata based algorithm equipped with a pruning rule. The aim of the proposed algorithm is to determine minimum number of sensors in such a way that each target can be monitored for at least k times. The proposed algorithm performance was evaluated through conducting a number of experiments. The experimental results were compared to those of a greedy-based algorithm. As shown by the final results, the learning-automata based algorithm was more successful than the greedy-based one regarding the construction of cover sets with minimum number of sensors.

Keywords: Wireless sensor networks, Cover set formation, Learning automata, k-coverage

1. Introduction

Wireless sensor networks (WSNs) have been highly attractive to researchers due to their wide range of applications such as military, national security, and environmental monitoring [2]. In such networks, one of the key issues is coverage problem. In a general definition, coverage refers to a measurement of the quality of services within a sensor network. It is mainly aimed to assure that each point in the field of interest be in the sensing range of at least one sensor node. Based on the targets that need to be monitored, coverage problems are classified into three types as follow [3]: 1) point coverage where a set of separate points is under a continuous coverage; 2) area coverage where all points in a bounded area are covered continuously; and 3) barrier coverage where particular path or boundary of an area is continuously covered. In a broader view, two types of coverage are involved in the coverage problem: simple coverage and k-coverage. In the former, each point of interest is monitored by one sensor node, which may lead to a low level of accuracy in the monitoring operation. Such inaccuracy has caused the researchers to pay more attention to the latter (k-coverage) in which each point of interest is monitored by at least k sensor nodes. This type of coverage can improve both accuracy and reliability of the monitoring operation.

The present paper addresses the target coverage problem specifically in cases where each target needs to be monitored by multiple sensors. This problem is generally recognized as k-coverage problem. The ultimate objective of this study is to maximize the network lifetime that refers to the amount of time during which the monitoring operation can be continued. This problem is significant because of two reasons: 1) the power resource of the sensors is limited and the batteries cannot be recharged or replaced, particularly in harsh environments [2]. Accordingly, power saving mechanisms that can optimize sensor energy consumption can be of a great advantage to our problem. One of the most popular power saving techniques is scheduling sensors to switch between active and sleep modes [1, 4]. In this technique, sensors are grouped into several cover sets each of which is able to monitor all targets. In this situation, the sensors that exist in the cover set are active, while the others are kept in sleep mode in order to save the energy as much as possible. Note that the scheduling technique actually takes the advantage of redundancy in sensor distribution. 2) the monitoring operation needs to be improved in terms of both accuracy and reliability. The networks that offer a higher expansion of coverage are generally more reliable since they further resist the sensor failures and errors that may occur during the monitoring operation.

Although literature is consisted of several solutions proposed to solve the k-coverage problem, modern heuristic methods such learning automata (LA), which have shown a high competency in solving NP-complete problems, have been ignored in this regard. This paper proposes an LA-based scheduling algorithm to find a near-optimal solution to the k-coverage problem. In this algorithm, network operation falls into a number of rounds each of which results in a cover set. To form a cover set, the algorithm identifies a potential cover set of the network. The

1 Department of Computer Engineering, Aliabad Katoul Branch, Islamic Azad University, Aliabad Katoul, Iran
2* Corresponding Author: Department of Computer Engineering, gorgan Branch, Islamic Azad University, gorgan, Iran. Email: ali.nodehi84@gmail.com
3 Department of Computer Engineering, Azadshar Branch, Islamic Azad University, Azadshar,Iran

Received: 2024.04.05; Accepted: 2024.07.06

constructed cover set receives a reward if its cardinality is less than that of the best cover set already found. As the algorithm is in progress, LA learn how to select the best sensors to form an optimal cover set among all existing cover sets. The cover set construction process goes on until all targets are provided with k-coverage. A number of experiments were carried out to evaluate the performance of the proposed algorithm regarding the size of the constructed cover sets. The results of the proposed algorithm were compared with those of a greedy-based one (originally proposed in [23]) that was modified for the purpose of the present research. As demonstrated by the final results, the algorithm proposed in this paper outperformed the greedy algorithm in terms of constructing cover sets with minimum number of sensors.

In general, this paper makes the following contributions: 1) proposing an LA-based algorithm as a solution to the k-coverage problem; 2) devising a pruning rule to enhance the proposed algorithm performance; 3) developing a greedy-based algorithm to solve the k-coverage problem; and 4) evaluating the performance of the algorithms through conducting a number of experiments.

The remainder of this article is organized as follows. Section 2 discusses the studies conducted to solve the target coverage problem. In Section 3, the problem of target k-coverage is presented. In Section 4, LA and variable action-set LA are introduced. In Section 5, a new scheduling algorithm is proposed for solving the problem. In Section 6, the performance of the proposed algorithm is evaluated through the simulation experiments. Finally, Section 7 concludes the paper.

2. Related work

Sensor networks are mainly used to collect data from harsh and remote environments. In such conditions, sensor nodes are commonly scattered in a random way, which may lead to a reduction in accuracy level of coverage. To compensate such lack of accuracy, we have to distribute more sensors than actually needed. This condition may cause WSNs to be susceptible to some errors since some targets are covered in a redundant way. On the other hand, sensors are limited in their power resource, which gives a high significance to the problem of maximizing the network lifetime. One of the efficient approaches to this problem is to schedule the sensors through dividing them into several cover sets and alternately activate one of them. In the following, we focus on solving the target coverage problem using the scheduling technique.

There are a number of studies in literature attempting to propose a solution to the target coverage problem in WSNs by means of the scheduling technique (e.g., [5, 6, 7, 8]). Cardei et al., [5] were one of the first researchers who addressed this problem in WSNs and modeled it as disjoint cover sets. In their work, each covers set was able to cover all the targets. Additionally, they proved the NP-completeness of the target coverage problem. In [6], the scholars extended the previous study to non-disjoint cover sets in which each of the sensors could be a member of more than one cover set. In [7], two greedy algorithms were proposed to maximize the number of cover sets and, at the same time, manage critical targets. In [8], to prolong the network lifetime, the researchers made use of the optimization capability of memetic algorithms.

Literature also consists of some studies that have used LA to solve the target coverage problem (e.g., [10, 11, 12]). In [10], an effective scheduling method was proposed on the basis of LA. They provided each sensor node with a learning automaton to make it possible for the sensor to choose its own proper mode, i.e., either active or asleep, at any given time. In [11, 12], several LA-based scheduling algorithms were proposed to solve the target coverage problem in WSNs. These algorithms made use of LA in order to identify those sensors that need to be activated at each stage to cover all the targets. Furthermore, several pruning rules were designed to enhance the network lifetime. In [1], the authors attempted to solve the target coverage problem using sensors with adjustable sensing range. In their proposed algorithm, LA were used to select several sensor nodes with minimum energy consumption in a way to cover all the targets. LA have been used also in solving the target coverage problem in directional sensor networks and they have shown a high capability in solving such problems [9, 13, 14, 17, 18, 19].

In the studies discussed above, only simple target coverage has been addressed, whereas in real applications, targets may require to be monitored by more than one sensor. This problem is generally known as k-coverage problem. In [20], the authors adopted centralized and distributed approaches to solve the k-coverage problem in mission-oriented mobile WSNs. They addressed also the sensor placement problem using the Helly's Theorem and geometric analysis of the Reuleaux triangle. In [21], the researchers investigated how to choose the minimum number of connected sensor nodes in a way to provide the targets with k-coverage. They proposed a connected k-coverage working set construction algorithm on the basis of the Euclidean distance in order to k-cover the sensing region and, at the same time, minimize the number of working sensors. To solve both coverage and connectivity problems, the authors in [22] introduced a scheme based on genetic algorithm in order to provide all the targets with k-coverage and each sensor node with m-connectivity.

Although several algorithms have been proposed to solve the k-coverage problem in WSNs, LA have not received their deserved attention considering their high capacity in solving complex problems. In this paper, we propose a scheduling algorithm based on LA to solve the k-coverage problem. In this study, LA are used to select appropriate sensors to provide full k-coverage for all targets. In addition, a pruning rule is designed to improve the performance of LA through preventing the selection of redundant sensors. The efficiency of the proposed algorithm is examined by several experiments through which the effects of some parameters on the size of constructed cover sets are also investigated.

3. The Problem of Target k-coverage

In this paper, the following problem is addressed. Assume a set of targets spread randomly within a 2-D environment. Additionally, a number of sensors are distributed randomly in the vicinity of targets to monitor them. Completely different from the simple coverage in which each target is monitored by a single sensor node, in our problem, targets need to be monitored by multiple sensor nodes (it depends on the value of k). That is, we are to address the k-coverage problem referring to a condition where each target is monitored by at least k sensor nodes. Initially, all sensors have the same amount of battery power that is non rechargeable. Each sensor in this problem monitors simultaneously all the targets positioned within its sensing range. Table 1 presents the notations used in this paper.

Table 1: Notations

Notation	Meaning
N	The number of sensors
M	The number of targets
k	The level of required coverage
s	A sensore, for all i=1,2,…,N
t	A target, for all m=1,2,…,M
l_i	Lifetime of sensor s_i
S	Set of sensors, $s_1 . s_2 . \dots . s_N$
T	Set of targets, $t_1 . t_2 . \dots . t_M$
$T(s_i)$	Refers to all the targets covered by sensor s_i

Problem: How to unify a minimum number of sensor nodes in a set (cover set) in such a way that it can completely provide k-coverage for all targets and, at the same time, the network lifetime can be maximized. To further clarify the problem, here is provided an example. Consider a sensor network comprising 2 targets and 4 sensors whose task is to monitor the targets. As can be observed in Figure 1, target t1is monitored by three sensors ({s1, s2, s3}) and target t2is monitored by three sensors ({s1, s3, s4}). If a simple coverage is desirable, we can form a cover set through activating a sensor s1or s3. However, if we are to establish a k-coverage (for example, if k=2) the cover set formed in the simple coverage is not applicable to the current situation. Therefore, some new sensors need to be added to the cover set in order to fully provide the k-coverage for all targets. As a result, selection of appropriate sensors to form the cover set is of a high importance to the network lifetime extension.

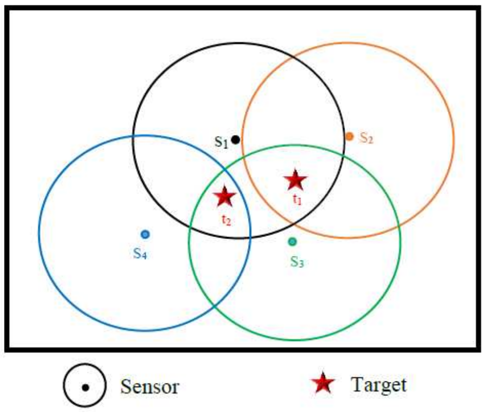

Fig. 1. Example network with four sensors and two targets.

4. Learning automata and variable action-set learning automata

4.1. Learning Automata

Automaton is a machine that automatically follows a predetermined sequence of operations or responds to encoded instructions. Learning automata do not follow the predetermined rules, but rather adapt to changes that may occur in random environment, which is the consequence of the learning process. LA are capable of choosing optimal actions from among the set of allowable actions. More specifically, a learning automaton possesses a finite number of actions that can operate. To each of these actions, a probability is allocated. The environment produces a reinforcement signal when an action is applied to it. The automaton uses the reply produced by the environment in order to update its action probability vector. Through running this procedure, the automaton learns how to select optimally the actions from among its action-set [15]. Figure 2 shows the interaction between a learning automaton and WSN, which is considered as a random environment in this paper.

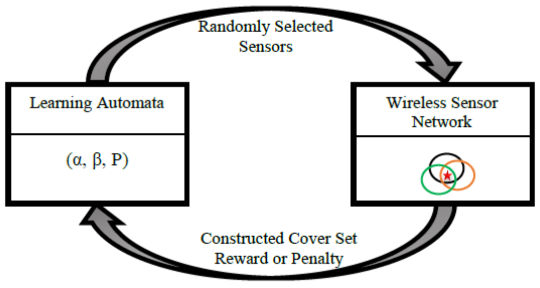
Randomly Selected Sensors
Learning Automata
(α, β, P)
Wireless Sensor Network
Constructed Cover Set Reward or Penalty

Fig. 2. Wireless sensor network and learning automata interaction.

The environment is shown by a triple E = (α,β,c) where:

α = { α1, α2, ..., αr} signifies the set of inputs (actions).

β = { β1, β2, ..., βr} is the set of outputs.

c = {c1,c2, ...,cr} is the set of penalty probabilities (the probability is measured by the reaction of the environment). Each element c is associated with an element of α (to assess the inputs that will be considered actions in the environment).

If the penalty probabilities are either constant or variable, the environment will be either stationary or non-stationary, respectively. Depending on the nature of the reinforcement signal b, the environment is categorized into either P model, Q model, or S model. In the P model environment, the reinforcement signal can take only two binary values of 0 and 1. In the Q model environment, the reinforcement signal is able to take a finite number of values in the interval [0,1], whereas in the S model environment, the reinforcement signal is actually a continuous random variable that assumes the values in the interval [a,b].

LA are categorized into two main groups [15]: fixed structure LA and variable structure LA. Here, the latter is explained in detail. Variable structure LA are represented by a triple (β, α, T), where β signifies the set of inputs, α represents the set of actions, and T is learning algorithm that is a recurrence relation used for modifying the action probability vector. Let αi(k)€α represent the action that is selected by learning automaton and p(k) denote the probability vector defined over the action-set at instant k. Let a represent the reward parameter that determines the amount of increase of the action probability values. And let b signify the penalty parameter determining the amount of decrease of the action probability values. Let r denote the number of actions that learning automaton can take. At each instant k, if the chosen action αi(k) is rewarded by the random environment, the action probability vector p(k) is updated according to Eq. (1). On the other hand, if that action is penalized, updating will be performed by Eq. (2). If a = b, the recurrence Eqs. (1) and (2) are called the linear reward-penalty (LR−P) method; if a»b, those equations are called the linear reward-e penalty(LR−εP) method; and if b = 0, they are called the linear reward-inaction (LR−I) method. In the last condition, once the chosen action is penalized by the environment, the action probability vectors remain unchanged.

$$p_j^{(k+1)} = \begin{cases} p_j^{(k)} + a[1 - p_j^{(k)}] & j = i \\ (1-a)p_j^{(k)} & \forall j \neq i \end{cases} \quad (1)$$

$$p_j^{(k+1)} = \begin{cases} (1-p)p_j^{(k)} & j = i \\ (\frac{b}{r-1}) + (1-b)p_j^{(k)} & \forall j \neq i \end{cases} \quad (2)$$

LA are great optimization tools that can be used in the following three conditions:

1) dynamic and complex environments that suffer from a high level of uncertainty;

2) situations in which there is a lack of information in regard to the environment;

and 3) cases where there is a need for solving large-scale NP-complete problems.

A typical example where all above-mentioned situations can be found is sensor network.

4.2. Variable Action Set Learning Automata

In a variable action-set learning automaton, the number of available actions at each instance is variable. In [16], it has been shown that once the reinforcement scheme is (LR−I), a variable action-set learning automaton is absolutely expedient and also e-optimal. Such an automaton consists of a finite set of n actions, α = { α1, α2, ..., αn}. A = {A1,A2, ...,Am} represents the set of action subsets and A(k) ⊆α stands for the subset of all actions that learning automaton can choose at each instant k. According to the probability distribution Ψ(k) = {Ψ1(k),Ψ2(k), ...,Ψm(k)}, an external agency selects randomly the particular action subsets. Where

Ψ(k) = prob[A(k) = Ai|Ai∈A, 1 ≤i≤ $2^n - 1$]

$\hat{p}$i(k) = prob[a(k) = ai|A(k), ai∈A(k)] represents the probability of choosing action αi, if the action subset A(k) has already been selected and αi∈A(k) too. The scaled probability $\hat{p}$i(k) is defined as

$$\frac{\mathrm{pi(k)}}{K(k)} \quad (3)$$

where K(k)=$\sum_{\alpha i \in Ak} pi(k)$ is the sum of the probabilities of the actions in subset A(k), and pi(k)=prob[α (k) = αi].

In a variable action-set learning automaton, the process of selecting an action and updating its probability can be explained as follows. Let A(k) denote the action subset chosen at instant k. Before selecting an action, using Eq. (3), the probabilities of all the actions in the selected subset are scaled. Then, according to the scaled action probability vector$\hat{p}$(k), the automaton randomly selects one of its own possible actions. Based on the responses from the environment, the scaled action probability vector of the learning automaton is updated. Note that in this step, only the probability of the available actions is updated. Finally, using pi(k+1) =$\hat{p}$i(k +1).K(k), probability vector of the actions contained in the chosen subset is re-scaled, for all αi∈A(k). A proof of the absolute expediency and ε-optimality of the above-described method can be found in [16].

5. Proposed Algorithm

In this section, a learning automata-based algorithm is proposed to solve the k-coverage problem in WSNs. This algorithm is equipped with a network of LA that collaborate with each other in a way to form efficient cover sets that are capable of extending the network lifetime. The algorithm operates in a few rounds (the number of rounds is dependent on the strategy adopted for selection of the sensors); in each round, one cover set is formed in a way to provide k-coverage for all the targets. Each round falls into two phases: 1) initialization where three main operations of the proposed algorithm are adjusted, and 2) cover set formation where cover sets are constructed with appropriate sensor nodes. The following subsections explain the details of the proposed algorithm.

5.1. Initialization

The initialization phase involves three operations: 1) constructing an LA network, 2) defining the action-set, and 3) configuring the action probability vector of LA. The LA network is formed by making available more than one learning automaton (maximally k automaton) for each of the targets. The reason of choosing maximally k automata is that in k-coverage, only one selection done by each automaton may fail to realize the final goal; thus, it is necessary to provide k automata for each target. LA is capable of identifying the most suitable sensors, i.e., the ones that provide k-coverage. The resultant network of LA can be modeled using a duple <A,α>, where A = {Ai,j|∀ti∈T,1 ≤j ≤k} signifies the set of LA corresponding to the targets within the network, and α= { αi,j|∀Ai,j∈A} represents the set of action-sets of LA where αi,j= {$\alpha_{i,j}^{1}$, j, $\alpha_{i,j}^{2}$, ..., $\alpha_{i,j}^{rj}$ } defines the setof actions that learning automaton Ai,j is able to select (for each αi,j∈α), and ri denotes the cardinality of action-set αi,j. In general, there are two types of LA: variable action set and fixed action set. As we are to avoid the selection of redundant sensors, the LA with variable action set are used and, additionally, a strong pruning rule is introduced to prune the action-set of LA. Each learning automaton in this algorithm can be in active or passive mode (remember that initially all LA are set to passive mode).

After the construction of an LA network, the action-set of LA need to be formed. In this algorithm, each learning automaton forms its action-set by assigning an action to each sensor that can monitor the target corresponding to the leaning automaton. That is, once an automaton chooses an action, it actually chooses a sensor to monitor its corresponding target as well as those positioned within the sensor's sensing range. Let target t_i be covered by αi= {$\alpha_{i,j}^{r}$} for each sensor covering target t_i. Action $\alpha_{i,j}^{r}$} is correspondent to the selection of sensor si(as an active sensor); this selection is done by learning automaton A_{ij}. When the actions-set of LA is formed, the algorithm needs to configure the action probability vectors of LA. letp = {pi,j|∀αi,j∈α} signify the set of action probability vectors and pi, j = {$p_{i,j}^{r}$|∀$\alpha_{i,j}^{r}$ ∈αi,j} stand for the action probability vector of learning automaton A_{ij}, where $p_{i,j}^{r}$is corresponding to the choice probability of action $\alpha_{i,j}^{r}$. To improve the convergence speed of the proposed algorithm, the action probability vector of automaton Ai need to be configured in such a way that the sensors with higher covering power have higher chance to be selected. For this purpose, the algorithm at first configures the action probability vector of the learning automaton Ai as follows:

$$p_{i,j}^{r}(\mathrm{q})=\frac{cp(si)}{\sum cp(si)} \tag{4}$$

where CP(si) signifies the covering power of sensor , s_i and CP(si) represents the total covering power of all sensors monitoring target t_i. To evaluate the covering power of sensor s_i, the following equation is used.

$$\mathrm{CP(si)}= |\mathrm{T(si)}\cap \mathrm{Tcur}| \tag{5}$$

where |T(si) ∩ Tcur| stand for the total number of uncovered targets monitored by sensor si. In this equation, a higher score is offered to the sensors that have higher covering power, while a lower one is offered to the sensors with lower covering power.

To elaborate the problem, an example network is depicted in Figure 1, which comprises four sensors and two targets. In this network, the coverage level is set to 2; therefore, maximally 2 automata can be assigned to each target(see Figure 3). The reason of selecting only 2 automaton is that output of each automaton is selection of only one sensor. If the first-level automaton of all targets is activated and k-coverage is not provided for some of the targets, then automaton of the next level (second automaton) of those targets will be activated and a sensor is selected to monitor the targets whose k-coverage has not been fully provided. As mentioned Before, each learning automaton forms its action-set by assigning an action to each sensor that monitors the target corresponding to the automaton. Remember that initially the action-set of an automaton corresponding to a particular target is identical to that of the other automata corresponding to the same target. For example, learning automaton A1,1 assigned to target t1holds three different actions since the target is monitored by three sensors ({s1, s2, s3}). The covering power of the sensors are as follow: CP(s1) = 2, CP(s2) = 1, and CP(s3) = 2. Here, the total covering power CP(si) is equal to 5. As a result, initially the action probability vector of automaton A1,1is defined as p1,1(1) = {$p_{1,1}^{s1}$= 0.4, $p_{1,2}^{s2}$= 0.2, $p_{1,3}^{s3}$= 0.4}. As can be seen, those sensors that have higher covering power are more likely to be chosen.

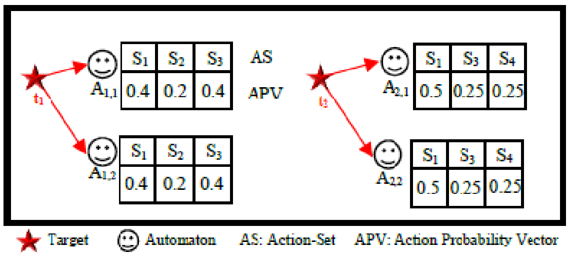

Fig. 3. Three operations of initialization phase

Bear in mind that the action-set and probability vector of LA might change over the time. In two situations, this variation takes place: 1) when energy of a sensor is exhausted, and 2) once the action-set of a learning automaton needs to be pruned. For instance, when sensor (s_i) becomes disabled at stage q+1, the action-set of learning automaton A_i is updated by removal of the action that is corresponding to the sensor. Afterwards, the choice probability of the eliminated action (α_i^j) is set to zero, whereas that of the other actions ($\alpha_i^{j'}$) is updated as follows.

$$p_i^{j'}(q+1)=p_i^{j'}(q).[1+\frac{p_i^j(q)}{1-p_i^j(q)}] \quad j\neq j' \qquad (6)$$

Up to this stage, three operations have been accomplished, namely LA network formation, action-set creation, and action probability vector configuration. The following subsection explain the phase of sensor selection.

5.2. Sensor Node Selection

In the sensor selection phase, the proposed algorithm chooses a subset of appropriate sensors to create a cover set. This algorithm is composed of a number of stages at each of which a subset of sensors is selected. These sensors are selected by LA through a process whose output is a cover set. During this process, the actions that have potential to cause redundancy are disabled. After that, a random environment is taken into account to measure the optimality of the resultant cover set. Here, WSN plays the role of a random environment for LA, which computes the cardinality (i.e., the number of activated sensors) of the created cover set as a response to optimality, according to which the selected actions are either rewarded or penalized. The iterative process of constructing cover set and updating the action probability vectors continues until a cover set with the minimum cardinality is formed. Algorithm 1 presents the pseudo code of the proposed algorithm. The qth stage of the algorithm can be explained as follows.

The T_{cur} set keeps the list of the targets not provided with k-coverage (all of the targets are initially assumed in this list). Let C_{cur} set hold the list of the sensors already involved in the cover set, whereas D_{cur} set keeps the list of dominated automata. After the process of cover set formation gets started, a target is randomly selected (i.e., t_i) and its first-level automaton is activated (i.e., Ai,1). Then, this automaton makes use of the pruning rule to prune its action-set and selects one of its actions that is actually a sensor monitoring the target. Next, the sensor corresponding to the chosen action is added to cover set C_{cur}. After that, the targets whose k-coverage is already provided are removed from the list T_{cur}. Afterward, the activated automaton (i.e., Ai,1) as well as the passive automata that is corresponding to the targets whose k-coverage is already provided are added to the dominated automata list (i.e., D_{cur}) in order to avoid activating these automata and choosing redundant sensors in the following stages. In other words, if a sensor monitoring multiple targets is selected, the automata correspondent to the targets provided with k-coverage must be added to the list of dominated automata. This way, the algorithm is able to choose only the automata whose actions cover new targets (i.e., the targets whose k-coverage has not been provided). The process through which a passive automaton is activated and an action is selected continues until all of the targets are provided with full k-coverage. Note that if the first-level automaton of all the targets is activated where the k-coverage of some targets is not satisfied, then the automata of the next level of those targets are activated, which perform the sensor selection process. At the time of selecting a sensor, the pruning operation is done in order to avoid selecting redundant sensors and already-activated ones. The LA activation process continues until a cover set is constructed.

Pruning rule: In this algorithm, each activated learning automaton prunes its action-set by disabling the actions corresponding to those sensors that cover the targets already provided with the k-coverage. In other words, this rule avoids the selection of redundant and already-selected sensors. This strategy reduces the number of actions, which causes the convergence speed to decrease. This finally reduces the running time of the algorithm.

Algorithm 1 : cover set formation

```
01.Input: wireless sensor network
02.Output: A cover set with appropriate sensors
03.assummption:
04.Assign one/more than (depending on k) automaton to each target
05.Let $\alpha_{i,j}$denote the action-set of automaton $A_{i,j}$
06.begin
07.Let $T_q$ denote the dynamic threshold at stage q
08.Let q denote the stage number initially set to zero
09. repeat
10.$T_{cur}\leftarrow T$
11.$Ccur\leftarrow \phi$
12.$Dcur\leftarrow \phi$
13. while $Tcur \not\equiv \phi$ do
14.    Find a passive automaton and activate it (call it $A_{i,j}$)
15.    Automaton $A_{i,j}$prunes its action-set and chooses one of its
actions (say ($s_i$))
16.    Add $s_i$corresponding to the selected action to $C_{cur}$
17.    Update the list of uncovered targets (i.e., $T_{cur}$)
18.    Add $A_{i,j}$and the automata related to the targets provided with
$k$-coverage by $s_i$to $D_{cur}$
19. end while
20. Compute the cardinality of constructed cover set($|C_{cur}|$)
21. if $|C_{cur}| \leq Tq$then
22.    Reward the chosen actions of the activated automata
23.    $Tq\leftarrow |Ccur|$
24. else
25.    Penalize the chosen actions of the activated automata
26. end if
27. re-enable all disabled actions
28. $q=q+1$
29. until (Stopping conditions = True)
30.end algorithm
```

Once a cover set is constructed, its suitability value needs to be computed. For this purpose, the cardinality of the cover set is determined and compared to predefined dynamic threshold Tq. Accordingly, the selection action of the automata, which is activated to form the cover set, is either rewarded or penalized. In cases where the cover set cardinality is a value less than the threshold, the selected actions are rewarded by the environment; otherwise, they are penalized. Initially, the threshold is set to a large value, then at each stage, it is set to the cardinality value of the latest rewarded cover set. After updating the action probability vectors of the activated automata, the q-th iteration ends; then, the disabled actions of the activated LA are enabled again. As the algorithm continues its operation, LA learn how to choose a subset of sensors in order to create a cover set with minimum value of cardinality. The algorithm has two termination criteria: either the probabilities of all selected sensors are higher than the particular threshold, or the number of created cover sets exceeds a predefined threshold. The final output of this operation would be a cover set with the minimum cardinality. which is activated to form the cover set, is either rewarded or penalized. In cases where the cover set cardinality is a value less than the threshold, the selected actions are rewarded by the environment; otherwise, they are penalized. Initially, the threshold is set to a large value, then at each stage, it is set to the cardinality value of the latest rewarded cover set. After updating the action probability vectors of the activated automata, the q-th iteration ends; then, the disabled actions of the activated LA are enabled again. As the algorithm continues its operation, LA learn how to choose a subset of sensors in order to create a cover set with minimum value of cardinality. The algorithm has two termination criteria: either the probabilities of all selected sensors are higher than the particular threshold, or the number of created cover sets exceeds a predefined threshold. The final output of this operation would be a cover set with the minimum cardinality.

When an optimal cover set is constructed, a working time (i.e., a fixed unit of time) is assigned to it. The value of this working time is added to the total network lifetime. Next, the residual energy of the sensors existing in the cover set is updated, and the sensors that have no residual energy are eliminated from the set of available sensors. The process of constructing cover set continues until the k-coverage of all the targets within the environment is completely provided.

6. Simulation Results

This section reports a number of simulations carried out to assess the performance of the proposed algorithm. The MATLAB software was used to perform the simulations. Several experiments were conducted to test the effects of different parameters on the size of constructed cover set. To model a WSN, the sensors and targets were randomly distributed within a field of 500(m)*500(m). Note that in this scenario, the targets that were not monitored by any sensor were ignored and those sensors that were not able to monitor even one target were removed from the sensors list. If a sensor run out of energy, it was recognized as a dead sensor. Each simulation was performed for 10 times; afterward, the average size of constructed cover set was computed for that simulation. By default, each sensor had only one unit of energy. Furthermore, the number of sensors and targets was fixed at 120 and 8, respectively. The number of sensors needed to cover each target (represented by k) was set to 2, and the sensing range was fixed at 100(m).

In algorithms designed based on LA, a key factor that plays a significant role in optimality of solutions is learning rate. Therefore, the learning rate value need to be determined with a high accuracy in such a way that the algorithm can obtain acceptable results during a suitable running time. The learning rate for the experiments carried out in the present paper was fixed at 0.1. The action probability vector of the learning automaton was updated by means of the reinforcement schemeL_{R-I}. Each round of the proposed LA-based algorithm was programmed to end in two conditions: either the number of created cover sets reached more than 100, or the probability value of the cover set exceeded 0.9.

To evaluate the performance of the proposed algorithm, the results obtained from the LA-based algorithm were compared to those of a greedy-based algorithm adapted from [23] for the purpose of our study. The greedy algorithm had been originally proposed to solve the simple target coverage problem with adjustable sensors. We modified this algorithm in a way to be applicable to the k-coverage problem in which conventional sensors were used. The cover set formation process in the greedy-based algorithm is done as follows. At each round, the greedy-based algorithm forms maximally one cover set. To this purpose, the algorithm initially starts with an empty cover set and all targets are assumed uncovered. Then, from among all available sensors, the algorithm chooses a sensor that have the highest contribution to the process of cover set formation and adds it to the cover set. Accordingly, the targets whose k-coverage has been provided are removed from the list of uncovered targets. Such process goes on until all k-coverage is provided for all the targets. Once a cover set is constructed, it is evacuated from redundant sensors (if any). Then, a working time is assigned to the cover set, during which the sensors in the cover set are active. Next, the energy of the sensors existing within the covers set is updated and the sensors without energy are eliminated from the cover set. The cover set construction process goes on until all the targets in the field of interest are fully provided with k-coverage.

Experiment 1. This experiment examined the influence of the k value on the size of constructed cover set. For this purpose, k was fixed ranging from 1 to 5with incremental step 1. Figure 4 shows the number of sensors activated to form a cover set. As can be seen, when the k value increases, more sensors are required to provide k-coverage for all of the targets. In addition, Figure 5 displays the amount of energy consumed to construct a cover set. Obviously, an increase in the k value causes an increase in the amount of energy consumption. Based on the results obtained from the conducted experiments, the LA-based algorithm outperformed the greedy-based one regarding the parameters discussed above.

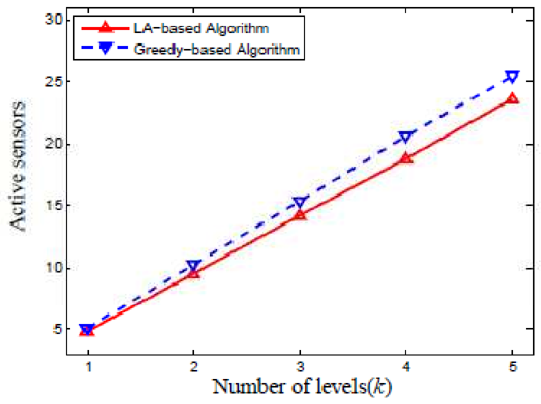

Fig.4. Effect of the coverage level on the number of active sensors

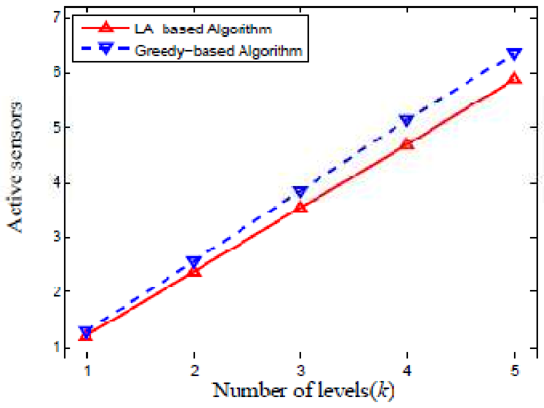

Fig. 5. Effect of the coverage level on the the energy consumption

Experiment 2. This experiment investigated the relationship between the size of constructed cover set and the number of sensors. In this test, the number of sensors was set to a value between 60 and 100 with incremental step 10. The obtained results presented in Figure 6 reveal that an increase in the number of sensors caused the size of cover set to reduce linearly. The reason was that some sensors were capable of covering more targets in a unit of time. In addition, the results demonstrated that the proposed algorithm outperformed the greedy-based algorithm in terms of forming cover sets with minimum number of active sensors. Bear in mind that this can result in improving the network lifetime. Remember that an increase in the number of sensors resulted in an increase in the network lifetime. The reason is that increasing the number of sensors cause each target to be monitored by more sensors, which finally led to formation of more cover sets.

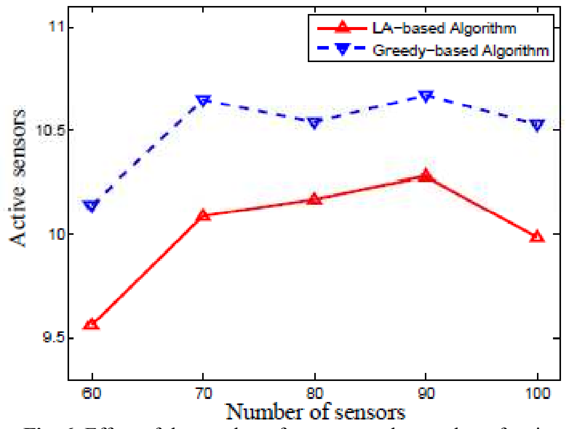

Fig. 6. Effect of the number of sensors on the number of active sensors

Experiment 3. This experiment examined the relationship between the number of targets and the size of constructed cover set. To this end, the number of targets was set to a number in the range of 6-10 with incremental step 1. As shown by the results presented in Figure 7, with an increase in the number of targets, the size of cover set linearly increased, too. This is natural since once the number of targets increases, more active sensors are required to provide the k-coverage for all the targets. As revealed by the experimental results, the proposed algorithm was more successful than the greedy-based one in all conditions regarding the construction of cover sets with minimum number of active sensors.

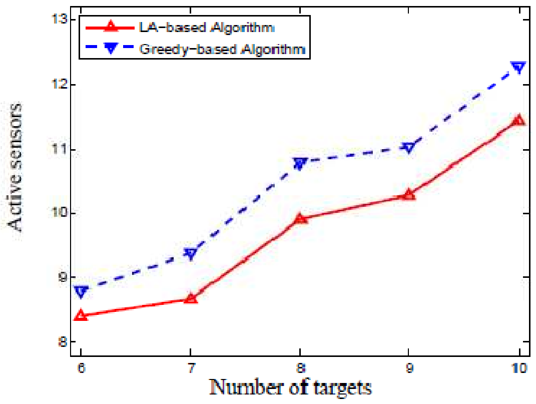

Fig.7. Effect of the number of targets on the number of active sensors

Experiment 4. This experiment investigated the influence of the sensing range on the size of constructed cover sets. To do this, the sensing range was set ranging from 80(m) to 120(m) with incremental step 10(m). As shown by Figure 8, increasing the sensing range led to formation of cover sets with lower number of active sensors. The reason was that when the sensing range broadened, the sensor nodes got able to monitor more targets; thus, fewer sensors were needed to provide all the targets with k-coverage.

Experiment 5. This experiment examined the coverage redundancy of the proposed algorithm in comparison with the optimal value that is always set to 1. As demonstrated by the results in Figure 9, the coverage redundancy of the proposed algorithm is very close to the optimal value. The reason is that the LA-based algorithm makes use of a pruning rule that prevents the selection of redundant sensors as far as possible.

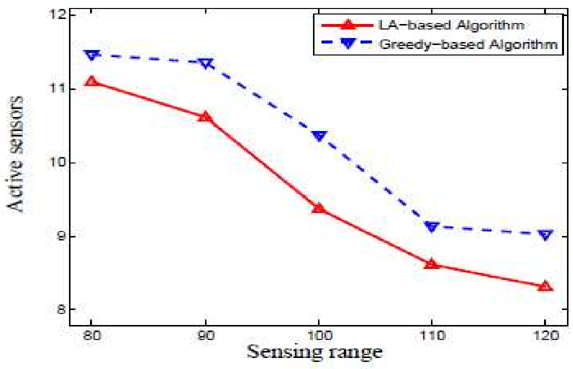

Fig. 8. Effect of sensing ranges on the number of active sensors

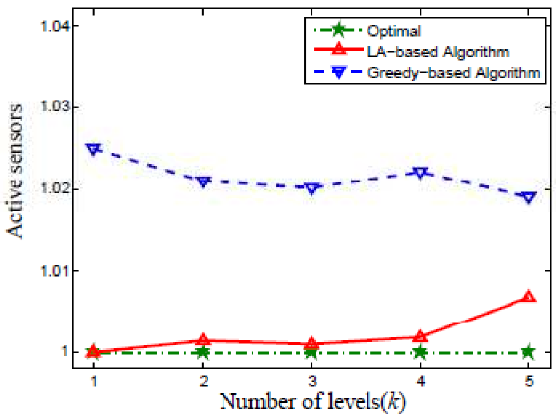

Fig. 9. Effect of the coverage level on the coverage redundancy

7. Conclusion

This study addressed the target k-coverage problem in WSNs. In this problem, to enhance both the accuracy and reliability of the monitoring operation, more than one sensor (remember that the exact number depends on k) is needed to monitor each target. As a solution to the problem, an LA-based scheduling algorithm was introduced in which LA were devised to select appropriate sensors capable of not only providing full k-coverage, but also maximizing the network lifetime through constructing cover sets with minimum number of sensors. In addition, LA was equipped with a pruning rule to improve the efficiency of the proposed algorithm. Several experiments were also conducted to evaluate the performance of the proposed algorithm, and the results were compared to those of a greedy-based one particularly developed for the purpose of the present paper. As demonstrated by the obtained results, both algorithms were successful in solving the k-coverage problem; though, the proposed LA-based algorithm showed higher efficiency in terms of forming cover sets containing the least number of sensors.

References

[1] H. Mohamadi, S. Salleh, M.N. Razali, S. Marouf, A new learning automatabasedapproach for maximizing network lifetime in wireless sensor networkswith adjustable sensing ranges, Neurocomputing, 153 (2015) 11-19.

[2] J. Yick, B. Mukherjee, D. Ghosal, Wireless sensor network survey, Computer Networks, 52 (2008) 2292-2330.

[3] C. Zhu, C. Zheng, L. Shu, G. Han, A survey on coverage and connectivityissues in wireless sensor networks, Journal of Network and ComputerApplications, 35 (2012) 619-632.

[4] H. Mohamadi, S. Salleh, M.N. Razali, Heuristic methods to maximize networklifetime in directional sensor networks with adjustable sensing ranges, Journal of Network and Computer Applications, 46 (2014) 26-35.

[5] M. Cardei, M.T. Thai, L. Yingshu, W. Weili, Energy-efficient target coveragein wireless sensor networks, In Proceedings of 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM),pp. 1976-1984 vol. 1973.

[6] M. Cardei, D.Z. Du, Improving wireless sensor network lifetime through power aware organization, Wireless Network, 11 (2005) 333-340.

[7] D. Zorbas, D. Glynos, P. Kotzanikolaou, C. Douligeris, Solving coverageproblems in wireless sensor networks using cover sets, Ad Hoc Networks, 8(2010) 400-415.

[8] C.K. Ting, C.C. Liao, A memetic algorithm for extending wireless sensor network lifetime, Information Sciences, 180 (2010) 4818-4833.

[9] H. Mohamadi, S. Salleh, A.S. Ismail, S. Marouf, Scheduling algorithms forextending directional sensor network lifetime, Wireless Networks, 21 (2015)611-623.

[10] H. Mostafaei, M. Meybodi, Maximizing lifetime of target coverage in wireless sensor networks using learning automata, Wireless PersCommun, 71(2013) 1461-1477.

[11] H. Mohamadi, A. Ismail, S. Salleh, A. Nodhei, Learning automata-based algorithmsfor finding cover sets in wireless sensor networks, J Supercomput,66 (2013) 1533-1552.

[12] H. Mohamadi, A. Ismail, S. Salleh, Solving target coverage problem usingcover sets in wireless sensor networks based on learning automata, Wireless Personal Communications, 75 (2014) 447-463.

[13] M.N. Razali, S. Salleh, H. Mohamadi, Solving priority-based target coverage problem in directional sensor networks with adjustable sensing ranges, Wireless Personal Communications, 95 (2017) 847-872.

[14] H. Mohamadi, S. Salleh, A.S. Ismail, A learning automata-based solutionto the priority-based target coverage problem in directional sensor networks,Wireless Personal Communications, 79 (2014) 2323-2338.

[15] K. Najim, A.S. Poznyak, Learning automata: theory and applications. NewYork: Printice-Hall, 1994.

[16] M.A.L. Thathachar, B.R. Harita, Learning automata with changing number of actions, IEEE Trans. Syst. Man Cybern., 17 (1987) 1095-1100.

[17] H. Mohamadi, A.S. Ismail, S. Salleh, A learning

automata-based algorithm for solving coverage problem in directional sensor networks, Computing, 95(2013) 1-24.

[18] H. Mohamadi, A. Ismail, S. Salleh, A. Nodehi, Learning automata-based algorithms for solving the target coverage problem in directional sensor networks, Wireless Personal Communications, 73 (2013) 1309-1330.

[19] H. Mohamadi, A.S. Ismail, S. Salleh, Utilizing distributed learning automatato solve the connected target coverage problem in directional sensor networks, Sensors and Actuators A: Physical, 198 (2013) 21-30.

[20] H.M. Ammari, On the problem of k-coverage in mission-oriented mobile wireless sensor networks, Computer Networks, 56 (2012) 1935-1950.

[21] J. Yu, X. Deng, D. Yu, G. Wang, X. Gu, CWSC: Connected k-coverage working sets construction algorithm in wireless sensor networks, AEU – International Journal of Electronics and Communications, 67 (2013) 937-946.

[22] S.K. Gupta, P. Kuila, P.K. Jana, Genetic algorithm approach for k-coverage and m-connected node placement in target based wireless sensor networks, Computers & Electrical Engineering, 56 (2016) 544-556.

[23] R. Cerulli, R. De Donato, A. Raiconi, Exact and heuristic methods to maximize network lifetime in wireless sensor networks with adjustable sensing ranges, European Journal of Operational Research, 220 (2012) 58-66.