

# A Learning Automata-based Algorithm for Area Coverage Problem in Directional Sensor Networks

**Zhimin Liu<sup>1</sup>, Zhangdong Ouyang<sup>2</sup>**

<sup>1</sup>School of Information Science and Engineering, Central South University, Changsha, 410205, China  
[e-mail: zhiminl@qq.com]

<sup>2</sup>School of Mathematics and Computational Science, Hunan First Normal University, Changsha, 410002, China  
[e-mail: oymath@163.com]

\*Corresponding author: Zhangdong Ouyang

*Received March 15, 2017; revised May 24, 2017; accepted June 27, 2017;  
published October 31, 2017*

---

## **Abstract**

Coverage problem is a research hot spot in directional sensor networks (DSNs). However, the major problem affecting the performance of the current coverage-enhancing strategies is that they just optimize the coverage of networks, but ignore the maximum number of sleep sensors to save more energy. Aiming to find an approximate optimal method that can cover maximum area with minimum number of active sensors, in this paper, a new scheduling algorithm based on learning automata is proposed to enhance area coverage, and shut off redundant sensors as many as possible. To evaluate the performance of the proposed algorithm, several experiments are conducted. Simulation results indicate that the proposed algorithm have effective performance in terms of coverage enhancement and sleeping sensors compared to the existing algorithms.

---

**Keywords:** Coverage enhancement, energy consumption, directional sensor networks, learning automata

## 1. Introduction

Recently, directional sensor networks consist of a large number of directional sensors have attracted a lot of attention due to their wide range of applications in different fields of target detecting, classification, and environmental detection. Compared with traditional wireless sensor networks, directional sensor networks can gather much richer information in the form of images or videos, and provide more detailed data about the environment [1]. The most distinguishing characteristic of directional sensor networks is that the sensing angle is limited due to technical constraints or cost considerations [2]. The sensing range of each directional sensor is a sector of disk, and each sensor can rotate sensing direction to monitor different regions.

One major problem in directional sensor networks is the coverage problem which reflects how well the target region is monitored. This problem handles the ability of the network to cover a certain area or certain events. Coverage problem can be categorized as three different types [3].

- Area coverage: covering the whole area of the network is the main objective of area coverage problem.
- Target coverage: Aiming to cover a set of stationary or moving points in monitoring region.
- Barrier coverage: minimizing the probability of undetected intruders through the barrier or sensor network.

In this paper, we focus on area coverage problem with aiming at enhancing coverage and closing redundant sensors as many as possible. The main method to solve this problem is to adjust sensing direction of sensors to enhance coverage, meanwhile, shut off redundant sensors. A learning automata-based algorithm is proposed which consists of two phrases: area coverage enhancement and sleeping redundant sensors. In the first phrase, a learning automata strategy based on a novel overlapping sensing ratio is adopted to enhance area coverage. Subsequently, a learning automata scheduling method based on sensor working state is presented to close redundant sensors so that energy consumption is reduced. Experimental results demonstrate that our proposed algorithm outperforms the existing methods in terms of coverage enhancement and shutting off redundant sensors.

The main results and contributions are summarized as follows:

- To the best of our knowledge, this work is first to tackle the area coverage with minimum number of sensors in directional sensor networks, based on learning automata.
- We use the notions of ineffective field of view and effective overlapping sensing area to derive an expression of overlapping sensing ratio for avoiding the sensing region of sensors falling outside monitoring region and overlapping with those of its neighbor sensors.
- We use an improved learning automata-based strategy to enhance area coverage in DSNs with better convergence speed. Subsequently, a learning automata scheduling method is present to shut off redundant sensors.

The remainder of this paper is organized as follows, In Section 2, related work is briefly summarized. In Sect 3. The problem statement is given. Section 4 briefly reviews the learning

automata theory. In Section 5 the proposed algorithm is presented. A series of simulation results are given in Section 6. Section 7 is the conclusion.

## 2. Related Work

Coverage enhancement, network connectivity, extending network lifetime are main issues have attracted considerable attention in directional sensor networks [4]. Coverage problem is the fundamental issue that reflects the quality of a monitoring region being detected, it can be classified as area coverage, target coverage and barrier coverage [3]. Several studies could be found that attempted to solve coverage problem in WSNs. Cardei et al [5] introduced the target coverage problem and modeled it as disjoint cover sets in which each cover set can monitor all targets. In [6], the authors presented two greedy algorithm for maximizing the number of cover sets to extend the network lifetime. Gupta et al [7] studied stochastic k-coverage and connectivity estimation model in WSNs with boundary deployment, and derived probabilistic expressions for k-coverage and connectivity using exact geometry. In [8-9], the authors proposed learning automata to solve the target coverage problem in WSNs. Mostafaei [10] modeled barrier coverage in WSNs as a stochastic coverage graph, and proposed a distributed learning automata-based algorithm to find near minimum required number of sensors to construct sensor barrier path. Aiming at target coverage problem in WSNs with adjustable sensing ranges, Mohamadi et al [11] presented a learning automata-based algorithm to select a number of sensors with minimum energy consumption to monitor all the targets. Esnaashari et al [12] taken advantage of global optimization capability of learning automata to solve the problem of dynamic target coverage in WSNs. A learning automata-based algorithm is proposed to select minimum number of sensors for covering a desired portion of the monitoring region of interest preserving the connectivity in WSNs [13].

The above algorithms do not receive appropriate performance in directional sensor networks because of limited sensing angle of sensors. Consider target coverage and connectivity problems in heterogenous directional sensor networks (HDSNs), Li et al [14] presented a learning automata-based method to choose cover-set that can cover all targets, simultaneously, preserve connectivity of HDSNs. Cai et al [15] introduced the multiple directional cover sets problem (MDCS), and proposed a feedback algorithm based on linear programming to organize the directions of sensors into a group of non-disjoint cover sets to prolong the network lifetime. Aiming at MDCS problem, a shortest path from target to sink (SPTS) – greedy algorithm is proposed to construct non-disjoint cover sets for extending the network lifetime [16]. Wang et al [17] proposed a genetic algorithm based on priority associated with each target for target coverage problem, a minimum subset of sensors that can monitor all targets was constructed. In [18], Aiming to reducing coverage redundancy as well as energy consumption in maximum target coverage with minimum number of sensors problem (MCMS), a distributed clustering algorithm was presented to gather sensors into clusters minimize the coverage redundancy and maximize the number of sleeping sensors. Recently, learning automata has attracted a wide interest in the field of target coverage problem in directional sensor networks. In [19], learning automata is used to find a near-optimal solution that can cover all targets with the minimum number of active sensor. Besides, four different learning automata-based algorithms are proposed to construct cover sets so that the network lifetime is extended. To solve priority-based target coverage problem (PTC), a learning automata-based algorithm is proposed to organize the sensors into several cover sets in such a way that each cover set can satisfy coverage requirements of all the targets

[20]. In [21], authors presented a new scheduling algorithm based on distributed learning automata for solving the connected target coverage problem in directional sensor networks.

Several studies have been found to solve area coverage problem in directional sensor networks. Liu et al [22] studied coverage prediction and number estimation model for directional sensor networks by consideration of boundary effect, a probability-based model is proposed to evaluate the area coverage and the number of sensors required to obtain a certain area coverage rate. In [23], authors studied the maximum directional area coverage (MDAC) problem, a distributed greedy algorithm is used to find approximate optimal solution. Tezcan et al [24] considered the presence of obstacles in the area, and presented a distributed algorithm based on sectional neighbor nodes and obstacles. Tao et al [25] first proposed the virtual potential field theory for solving area coverage problem of directional sensor network by introducing sensing centroid concept innovatively.

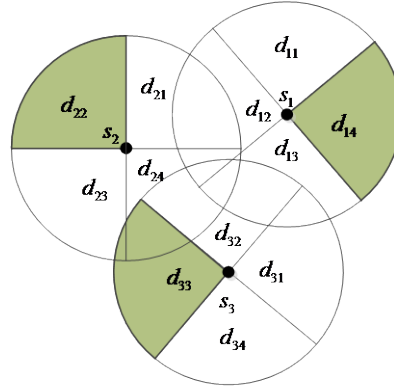
Zhao et al [26] proposed a virtual centripetal force-based coverage enhancing algorithm in wireless multimedia sensor networks by introducing the concept of centripetal force and sensing area grid and defining overlapping sensing ratio. In the algorithm of VCFCEA, new sensing blanks may generate when all redundant sensors are shut off in one-time to extend the network lifetime, and the convergence of the algorithm is slow. To solve these problems, Chen et al [27] proposed a traditional overlapping sensing ratio-based coverage-enhancing algorithm (OSRCEA). This algorithm includes two step: a priority-based strategy for enhancing coverage and shutting off redundant sensors, respectively. Sensing direction can be adjusted according to overlapping sensing ratio. After area coverage enhancement, the redundant sensors with highest priorities shut themselves off simultaneously so that new sensing blanks may be generated. In algorithms of VCFCEA and OSRCEA, the authors did not take boundary effect into consideration. When the sensing range of sensor falls outside monitoring region, the traditional overlapping sensing ratio ignores the boundary effect so that accuracy of adjusting sensing direction is not enough.

Considering the drawbacks of current algorithms (OSRCEA and VCFCEA) for area coverage in directional sensor networks. In this paper, we define the concepts of effective field of view (EFOV), ineffective field of view (IEFOV), effective overlapping sensing area (EOSA) and overlapping sensing ratio (OSR). Moreover, a new learning automata-based algorithm is designed to enhance area coverage and shut off redundant sensors, respectively. The proposed algorithm intend to construct a set of directional sensors that can cover maximum area with minimum number of active directional sensors.

### 3. Problem statement

#### 3.1 Network model

In this paper, we focus on the following scenario. A large number of directional sensors are randomly deployed within a two-dimensional monitoring region. Each sensor has several selectable sensing directions, and only one sensing direction is in working state at each unit of time, which is called working direction. After random deployment of directional sensor networks, each sensor initially selects one of its directions as working direction, and the working direction determines the active sensing range. A region is said to be covered if any point in the region is covered. It should be noticed that some overlapping sensing regions covered simultaneously by neighboring sensors may be generated.



**Fig. 1.** An illustrative example of a directional sensor networks

In this paper, all directional sensors have the following characteristics:

- All sensors in the network are homogeneous, i.e. the same sensing radius, the same field-of-view (Fov) angle, etc.
- All sensors are located at fixed position and their sensing direction is adjustable.
- Sensor knows its own position and sensing direction, and as well as related information of its neighboring sensors.

The following notations are used throughout this paper:

- $N$  the number of directional sensors
- $M$  the number of sensing directions per sensor
- $s_i$  the  $i^{\text{th}}$  sensor,  $1 \leq i \leq N$
- $d_{ij}$  the  $j^{\text{th}}$  sensing direction of sensor  $s_i$ ,  $1 \leq i \leq N, 1 \leq j \leq M$
- $S = \{s_1, \dots, s_N\}$  the set of sensors.

**Fig. 1** shows an example of directional sensor networks. In this example,  $s_i$  ( $1 \leq i \leq 3$ ) represents a set of sensors. Each sensor has four sensing directions and  $d_{ij}$  ( $1 \leq i \leq 3, 1 \leq j \leq 4$ ) indicates the sensing directions of  $s_i$ . The current working directions of  $s_1$ ,  $s_2$  and  $s_3$  are denoted as  $d_{14}$ ,  $d_{22}$  and  $d_{33}$ , respectively.

**Problem:** How to adjust sensing directions of each sensor so that the area coverage is enhanced, at the same time, how to find a subset of sensors in DSNs that can cover points in monitoring region as many as possible.

### 3.2 Directional sensing model

Directional sensors are randomly scattered to monitor a bounded region. Each sensor  $s_i$  has a sensing radius  $r$ , a Fov angle  $2\alpha$ , and an orientation (direction) vector  $V_i$ , which together define the sensing sector (**Fig. 2**). We use  $s_i$  to denote the  $i^{\text{th}}$  sensor. Without ambiguity,  $s_i$  also denotes the sensor's position. A point  $P$  is covered by a sensor  $s_i$  if  $P$  is in the sensing sector of  $s_i$ , that is,  $|s_i P| < r$  and  $\partial(\overrightarrow{s_i P}, \overrightarrow{s_i V_i}) < \alpha$ , where  $s_i P$  is the vector from  $s_i$  to  $P$ .

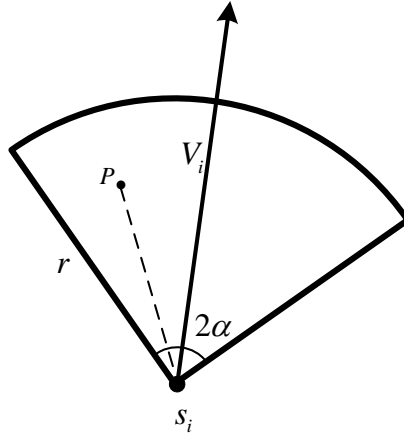


Fig. 2. Directional sensing model

### 3.3 Notations and definitions

In this section, a grid model in [27] is adopted to calculate overlapping sensing region accurately. Fig. 3 shows that the sensing radius  $r$  and the Fov angle  $2\alpha$  are equally divided into  $P$  and  $Q$ , respectively, so the sensing region is divided into  $P \times Q$ . The center point of each grid is regarded as the grid's position. The area of a grid located in  $p$ -row- $q$ -column is denoted as:

$$AG_{ipq} = (p^2 - (p-1)^2) \times \left(\frac{r}{P}\right)^2 \times \left(\frac{\alpha}{Q}\right) \tag{1}$$

It can be noted that the area of each grid is just related to the row variable  $p$ , while the other parameters are constant.

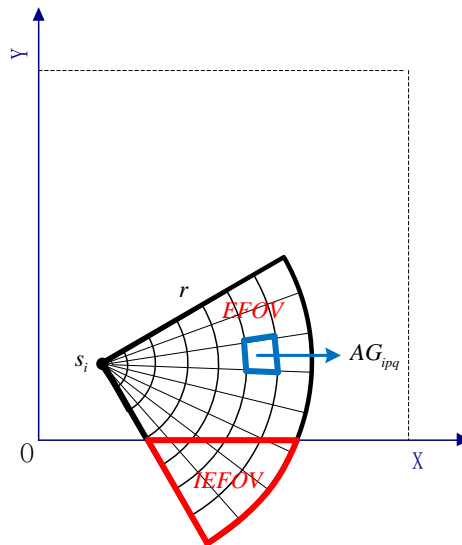


Fig. 3. Division of sensing area and computation of effective field of view

**Definition 1 Effective Field-of-view (EFOV):** The sensing region of a sensor interacts with the monitoring region is defined as effective field-of-view, as shown in Fig. 3. It is assumed that  $\Omega$  denotes the monitoring area. Given that  $G_{ipq}$  represents a covered mark of  $p$ -row- $q$ -column of sensor  $s_i$ , if a grid in  $p$ -row- $q$ -column is within  $\Omega$ ,  $G_{ipq} = 1$ ; otherwise  $G_{ipq} = 0$ . EFOV is mathematical expressed as follow.

$$EFOV = \sum_{p=1}^P \sum_{q=1}^Q G_{ipq} \times AG_{ipq} \quad (2)$$

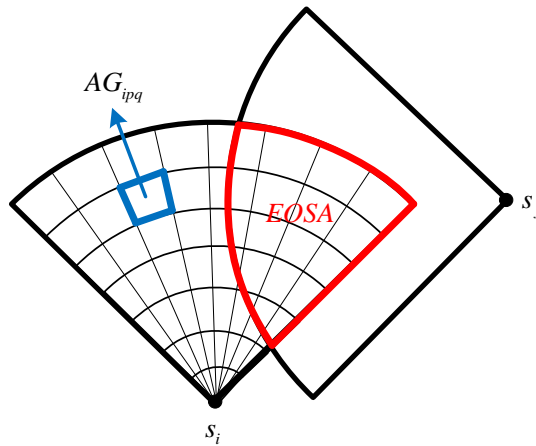
**Definition 2 Ineffective Field-of-View (IEFOV):** The sensing region of sensor falling outside monitoring area  $\Omega$  is called as IEFOV (Fig. 3). Given that  $FOV = \alpha r^2$  represents the sensing region of sensor  $s_i$ . According to the definition, IEFOV is denoted as  $IEFOV = FOV - EFOV$ .

**Definition 3 Effective Overlapping Sensing Area (EOSA):** The EFOV of  $s_i$  interacts with the EFOV of its all neighbor sensors is defined as effective overlapping sensing area (Fig. 4.).  $C_{ijpq}$  denotes a covered mark of  $p$ -row- $q$ -column grid in EFOV of sensor  $s_i$ , if this grid in  $p$ -row- $q$ -column is covered by EFOV of neighboring sensor  $s_j$ , then  $C_{ijpq} = 1$ ; otherwise  $C_{ijpq} = 0$ . Given that  $N(s_i)$  is a set of neighbor sensors of  $s_i$ . EOSA is expressed as follow.

$$EOSR = \bigcup_{j=1}^{|N(s_i)|} \left( \sum_{p=1}^P \sum_{q=1}^Q C_{ijpq} \cdot AG_{ipq} \right) \quad (3)$$

**Definition 4 Overlapping Sensing Ratio (OSR):** (6): It could be defined as the ratio between the sum of IEFOV & EOSA and FOV (Fig.5.), denoted as follows.

$$OSR = \frac{IEFOV + EOSA}{FOV} \quad (4)$$



**Fig. 4.** Computation of effective overlapping sensing area

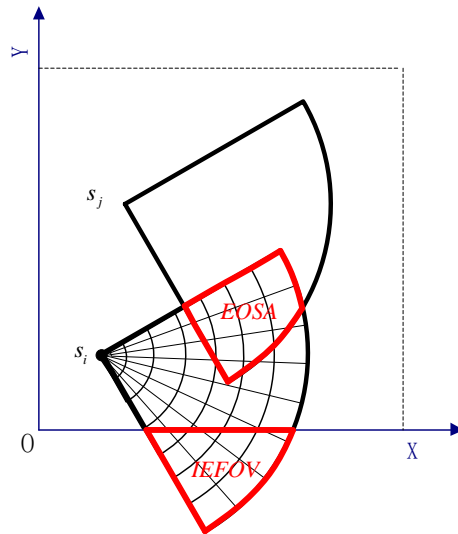


Fig. 5. Computation of overlapping sensing ratio

### 4. Learning automata

A learning automaton refers to an adaptive decision-making unit improving its own performance through learning how to choose the optimal action from a finite set of actions and performs it on a random environment [28]. To achieve this goal, it interacts continuously with a random operating environment. Learning automaton starts its operation by selecting one of its available actions based on its action probability vector. The chosen action is then served as the input to the random environment. Afterwards, using a reinforcement signal, the environment responds to the taken action. The action probability vector of learning automaton is updated according to the reinforcement feedback given by the environment. Generally, a learning automaton aims at finding the optimal action from the action-set in a way that the average reward received from the environment could be maximized [29-30], as shown in Fig. 6.

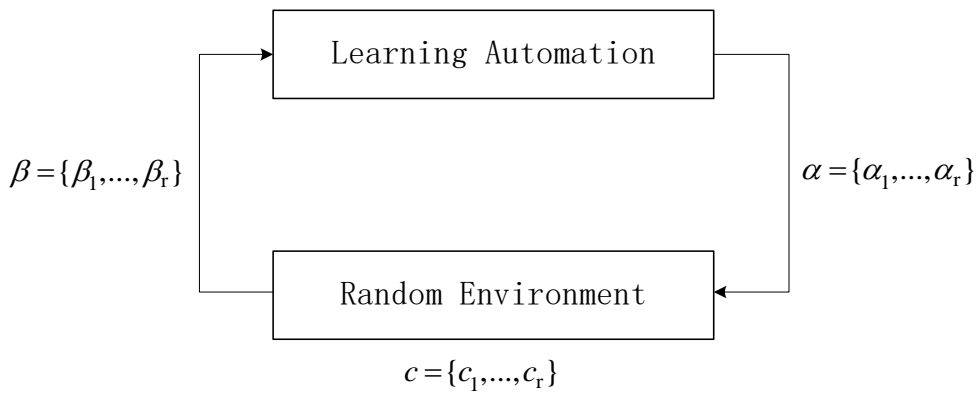


Fig. 6. Learning automata



#### 4.1 Random environment

The environment of LA is defined by a triple  $E = (\alpha, \beta, c)$  [28] where:

$\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  is the set of inputs

$\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$  is the set of outputs

$c = \{c_1, c_2, \dots, c_r\}$  is the set of penalty probabilities. Each element  $c_i$  is associated with an element of  $\alpha_i$ .

Depending on if  $c_i$  is constant or variable, the environment is categorized as stationary or non-stationary, respectively. Based on the nature of reinforcement signal  $\beta$ , the environment can be classified into P-model, Q-model, and S-model. P-model refers to an environment in which the reinforcement signal is able to take only two binary values 0 and 1. Q-model is an environment in which a finite number of the values in the interval  $[0, 1]$  can be taken by the reinforcement signal. Finally, in an S-model of the environment, the reinforcement signal is a continuous random variable that assumes values in the interval  $[a, b]$ .

#### 4.2 Learning automation

Learning automata can be categorized into two main groups [28]: fixed structure LA and variable structure LA. Variable structure LA are represented by a triple  $(\beta, \alpha, L)$ , where  $\beta = \{\beta_1, \beta_2, \dots, \beta_r\}$  signified the set of inputs,  $\alpha = \{\alpha_1, \alpha_2, \dots, \alpha_r\}$  represents the set of actions, and  $L$  is the learning algorithm that is a recursive relation used for updating the action probability vectors. The action, denoted as  $\alpha_i(k)$ , is selected by learning automata and  $p(k)$  represent the probability vector defined over the action-set at stage  $k$ . Let  $p_i(k)$  denote action probability related with action  $\alpha_i(k)$ . 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 stage  $k$ , if the chosen action  $\alpha_i(k)$  is rewarded by random environment, the action probability vector will be updated according to Eq.5. On the other hand, if the action is penalized by random environment, the action probability vector will be updated according to Eq.6.

$$p_j(k+1) = \begin{cases} p_j(k) + a[1 - p_j(k)] & j = i \\ (1-a)p_j(k) & \forall j \neq i \end{cases} \quad (5)$$

$$p_j(k+1) = \begin{cases} (1-b)p_j(k) & j = i \\ b/(r-1) + (1-b)p_j(k) & \forall j \neq i \end{cases} \quad (6)$$

In the above equations, If  $a = b$ , the recurrence equations are called the linear reward-penalty ( $L_{R-P}$ ) method; If  $a > b$ , those equations are called the linear reward- $\varepsilon$  penalty ( $L_{R-\varepsilon P}$ ) method; and if  $b = 0$ , they are called the linear reward-inaction ( $L_{R-I}$ ) method. In the last condition, once the chosen action is penalized by the environment, the action probability vectors remain unchanged.

## 5. Proposed algorithm

Coverage and energy consumption are major problems in directional sensor networks. Aiming at covering maximum area with minimum number of active sensors to enhance coverage and reduce energy consumption, a learning automata-based on overlapping sensing ratio is designed to enhance area coverage. Subsequently, a learning automata based on sensor working state is proposed to shut off redundant sensors for saving more energy.

### 5.1 Learning automata-based algorithm for adjusting sensing direction (LAASD)

In this phase, the learning automata network is constructed by assigning a learning automation to each direction sensor. The constructed network can be modeled by a duple  $\langle A, \alpha \rangle$ , where  $A = (A_1, \dots, A_N)$  denotes a set of learning automata assigned to the sensors. By constantly interaction with random environment, learning automation is in charge of choosing near-optimal sensing direction among the directions of its corresponding sensors as current working direction. Whereas,  $\alpha = (\alpha_1, \dots, \alpha_N)$  denotes action sets that can be taken by  $A$ . Element  $\alpha_i = (\alpha_{i1}, \dots, \alpha_{iM})$  represents the action set of  $A_i$ . Where, action  $\alpha_{ij}$  corresponds to the selection of the  $j^{\text{th}}$  sensing direction of sensor  $s_i$ . Let  $p = (p_1, \dots, p_N)$  denotes the action probability vectors of the network of learning automata. Element  $p_i = (p_{i1}, \dots, p_{iM})$  represents the action probability vector of learning automation  $A_i$ , where  $p_{ij}$  is corresponding to the choice probability of action  $\alpha_{ij}$ , and  $\sum_j p_{ij} = 1$ . This phrase is an iterative algorithm consists

of several stages where  $k$  denotes the  $k^{\text{th}}$  stage. At the beginning of the algorithm ( $k = 0$ ), the action probability vector of learning automation  $A_i$  is defined as

$$p_{ij}(k) = \frac{1}{|\alpha_i(k)|} \quad \forall \alpha_{ij} \in \alpha_i, k = 0 \quad (8)$$

Where  $|\alpha_i(k)|$  represents the number of actions that learning automaton can take, also indicates the number of sensing directions of  $s_i$ .

---

**Phrase 1:** Learning automata-based algorithm for adjusting sensing direction (LAASD)

---

**Input :** Directional sensor networks

**Output:** The appropriate working directions

**Begin**

**Do**

Assign learning automation to sensor  $s_i$

Automation  $A_i$  randomly selects one of its actions(i.e, a working direction)

Compute its corresponding  $OSR$

**If** ( $OSR \leq OSR_{thr}$ ) **then**

Reward the selected action by automation  $A_i$

**Else**

Penalize the selected action by automation  $A_i$

**End if**

---

---

```

    k = k + 1
    While ( max{pij(k)} ≤ Pthr || k ≤ Kthr ).
end

```

---

The algorithm LAASD runs into a number of iterations at each of which learning automation corresponding to each sensor selects an action from action set based on its action probability vector and decides to update its action probability vector. Then, the action probability vector of learning automation is updated according to the formulas in [Eq.5](#) and [Eq.6](#)

In order to obtain better convergence speed, we modify the reward and penalty parameters. These parameters could be expressed as follows:

$$b = a = \frac{FOV - EOSR}{FOV + \varepsilon}, \varepsilon \in (0, 1] \quad (9)$$

The pseudo code of LAASD is shown as [Phrase 1](#), this phrase can be briefly described as follows. Each of learning automations randomly chooses a sensing direction of its corresponding sensor as working direction. At each iteration, if *OSR* exceed a specified threshold  $OSR_{thr}$ , the chosen action  $\alpha_{ij}$  could be penalized, and the action probability vector  $p_{ij}$  associated with  $\alpha_{ij}$  is updated according to [Eq.6](#). On the other hand, if the action is rewarded by random environment, the action probability vector will be updated according to [Eq.5](#). Each learning automation independently stops its action selection process in two conditions: the selection probability of one of its action exceeds the predefined  $P_{thr}$  or the number of iterations exceed the threshold  $K_{thr}$ .

## 5.2 Learning automata-based algorithm for shutting off redundant sensors (LAASRS)

After the enhancement of area coverage, a learning algorithm based on sensor working state is proposed to shut off redundant sensors so that the energy consumption is reduced in directional sensor networks.

In this phrase, the constructed network can be modeled by a duple  $\langle A, \alpha \rangle$ , where  $A = (A_1, \dots, A_N)$  represents the set of learning automation assigned to the sensors. These LA are in charge of selecting near-optimal working state among the working states of their own corresponding sensors as current working state. Whereas  $\alpha = (\alpha_1, \dots, \alpha_N)$  represents the action sets that can be selected by  $A$ . Where,  $\alpha_i = (active, sleep)$  defines the working state of sensor  $s_i$ . Let  $p = (p_1, \dots, p_N)$ , in which  $p_i = (p_{i,active}, p_{i,sleep})$ , denotes the action probability vectors of the network of learning automata. This phrase is also an iterative algorithm consists of several stages where  $k$  denotes the  $k^{th}$  stage. At the beginning ( $k = 0$ ), the action probability vector of learning automation  $A_i$  is defined as

$$p_{i,active}(k) = p_{i,sleep}(k) = \frac{1}{|\alpha_i(k)|} = 0.5 \quad k = 0$$

The pseudo code of LAASRS is shown as in [Phrase 2](#). In this phrase, a learning automation selects an action from action set based on its action probability vector, and decides to update its action probability vector. This phrase can be briefly described as follows.

(1) If the selected action corresponding to  $s_i$  is *active* : If  $OSR_i$  exceeds a certain threshold  $OSR_{thr}$ , the chosen action *active* is penalized by random environment, the related action probability vector will be updated according to [Eq.6](#). At the same time, the working state of sensor is changed to *sleep* ; otherwise, the action *active* will be awarded by the random environment according to [Eq.5](#).

---

**Phrase 2:** Learning automata-based algorithm for shutting off redundant sensors (LAASRS)

---

**Input :** Wireless Multimedia Sensor Networks after coverage enhancement

**Output:** The active sensors

**Begin**

**Do**

Assign learning automation to sensor  $s_i$

Automation  $A_i$  randomly selects one of its actions (*active*, *sleep*)

Compute its corresponding  $OSR_i$

**If** (*active*) **then**

**If**( $OSR \geq OSR_{thr}$ ) **then** Penalize the selected action *active*, set  $s_i$  be *sleep*

**Else** Reward the selected action *active* using automation  $A_i$

**End if**

**Else**

**If**( $OSR \geq OSR_{thr}$ ) **then** Reward the selected action *sleep* using automation  $A_i$

**Else** Penalize the selected action *sleep*, set  $s_i$  be *active*

**End if**

**End if**

$k = k + 1$

**While** ( $\max\{p_{ij}(k)\} \leq P_{thr} \parallel k \leq K_{thr}$ ).

Select optimal sensors as active according to their corresponding action probability vectors

**end**

---

(2) If the selected action corresponding to  $s_i$  is *sleep* : If  $OSR_i$  exceeds a certain threshold  $OSR_{thr}$ , the chosen action *sleep* is rewarded by random environment, the action probability vector will be updated according to [Eq.5](#); otherwise, the action *sleep* will be penalized by the random environment according to [Eq.6](#). At the same time, the working state of sensor is changed to *active*.

## 6. Simulation results

Authors [27] have demonstrated that the algorithm OSRCEA gained better performance than VCFCEA in terms of coverage enhancement and energy consumption. In order to simplify the experiment, in this section, the extensive comparison between the proposed algorithm and OSRCEA is presented. It is assumed that all sensors of directional sensor networks are homogeneous. The main parameters involved in the experiment are listed in [Table 1](#).

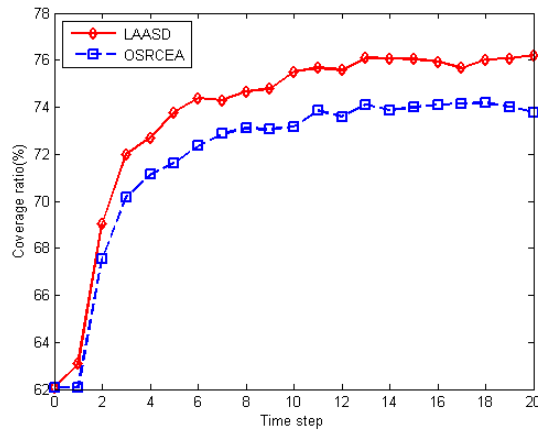
[Fig. 7](#) shows the convergence of LAASD and OSRCEA. The parameters of this experiment are as follows: the number of sensors is 100, sensing radius is 60m and the Fov angle is  $90^\circ$ . In order to obtain more stable data, this experiment is repeated 30 times and the mean value is

obtained. It can be seen that LAASD is convergent and has a faster convergence rate than OSRCEA. With the increase of adjusting times, the coverage ratio is increasing continuously. After 10 adjustments, LAASD has reached a stable state, and the coverage ratio is increased to 75.5%. The algorithm OSRCEA was not stable until the 11th adjustment, and the coverage ratio is only increased to 73.8%.

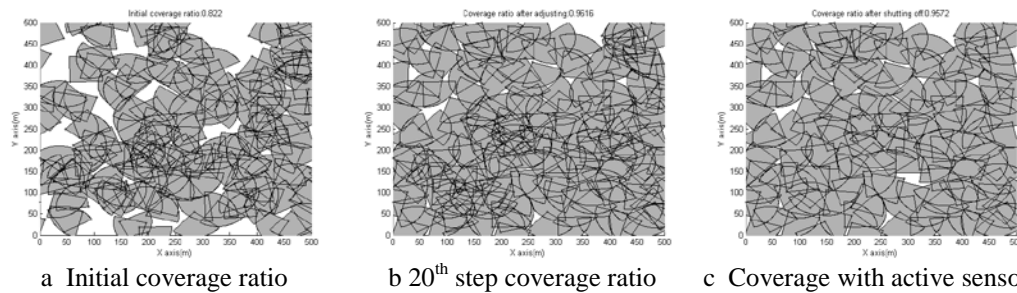
**Table 1.** Experimental parameters

| Parameter                              | Variation   |
|--|---|
| Length of monitoring area $L$          | $500m*500m$   |
| Number of sensors $N$                  | 50, 100, 150, 200, 250, 300   |
| Sensing radius $r$                     | 20m, 30m, 40m, 50m, 60m, 70m  |
| Fov angle $2\alpha$                    | $15^\circ, 30^\circ, 45^\circ, 60^\circ, 75^\circ, 90^\circ, 105^\circ$ |
| Number of directions of per sensor $M$ | 3   |

There is an example shown in **Fig. 8**, The parameters of the example are as follows: the number of sensors is 200, the sensing radius is 60m and the Fov angle is  $90^\circ$ . **Fig. 8a** indicates the initial coverage ratio of the example is 82.2%. **Fig. 8b** shows the coverage ratio is enhanced to 96.16% after 20 adjustments of sensing direction, but many redundant sensors still exist in the network. In order to save energy of sensors, LAASRS is adopted to shut off redundant sensors, as shown in **Fig. 8c**. After 9 iterations, 47 redundant sensors are closed. It can be seen that the coverage loss is very small and the coverage ratio is still maintained at a high level of 95.72%.



**Fig. 7.** Comparison of convergence rate



a Initial coverage ratio

b 20<sup>th</sup> step coverage ratio

c Coverage with active sensors

**Fig. 8.** Example of coverage enhancement and redundant sensors-shutting off  
(a. Initial coverage ratio is 82.2% b. 20<sup>th</sup> step coverage ratio is 96.16% c. Number of active sensor is 153, coverage ratio is 95.72%)

### 6.1 Impact of the number of sensors

In this experiment, the number of sensors is ranged between 50 and 300 with incremental step 50, the sensing radius is 60m and the Fov angle is 90°. As shown in Fig. 9, whether the network is dense deployment or sparse deployment, LAASD obtains better coverage increment ratio than OSRCEA. This is because with the increase of number of sensors, a large number of sensors could fall in boundary region of monitoring region, algorithm LAASD is used to adjust working direction to avoid the IEFOV and EOSA, but algorithm OSRCEA ignore reducing IEFOV of sensors by adjustment of its working direction. Specially, when the number sensors is 100, the increment ratio of LAASD is over 30%, whereas, the increment ratio of OSRCEA is 25%.

Fig. 10 indicates the effect of different sensor scale on the number of closed sensors. When the number of sensor is greater than 100, LAASRS has better performance in shutting off redundant sensors than OSRCEA. Because redundant sensors with highest priorities shut themselves off in OSRCEA simultaneously, new sensing blanks may be generated, and redundant sensors with lower priorities may be selected to cover new sensing blanks.

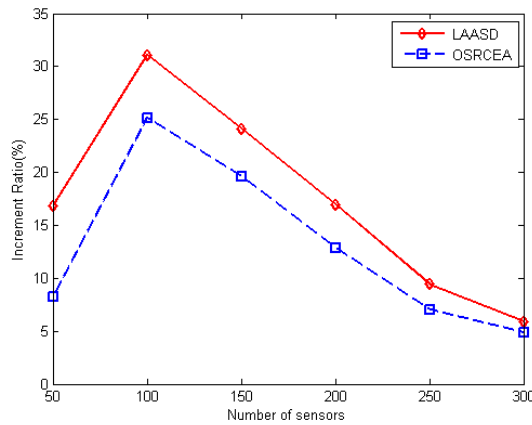


Fig. 9. Influence of sensor scale on the increment ratio

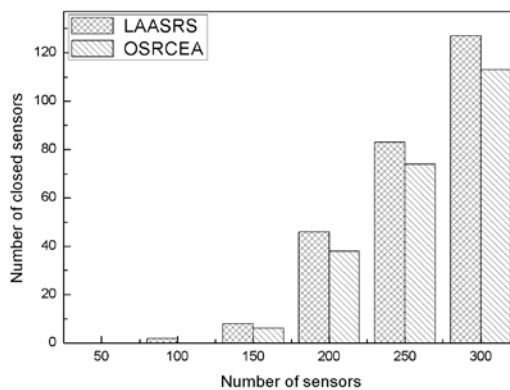


Fig. 10. Influence of sensor scale on the number of closed sensors

## 6.2 Impact of the sensing radius

In this experiment, the sensing radius is ranged between 20m and 70m with incremental step 10, the number of sensors is 200, and the Fov angle is  $90^\circ$ . Fig. 11 indicates LAASD gets higher increase of coverage ratio than OSRCEA. This is because with the increase of sensing radius, the sensing region of sensor could fall outside monitoring region with a high probability. LAASD can decrease IEFOV of sensor by adjustment of its working direction. Specially, when the sensing radius is in the range of 30m-60m, it can be noted that the increment ratio of LAASD is over 17%. To gain the same increment ratio of coverage in OSRCEA, the sensing radius should be in range of 40m-60m. After all sensors keep stable, LAASRS is adopted to shut off the redundant sensors.

Fig. 12 shows the effect of different sensing radius on the number of closed sensors. When the sensing radius is greater than 40m, LAASRS obtains better performance in shutting off redundant sensors than OSRCEA. With the increase of sensing radius, more sensing blanks may be appeared after using OSRCEA to closing redundant sensors with highest priorities simultaneously.

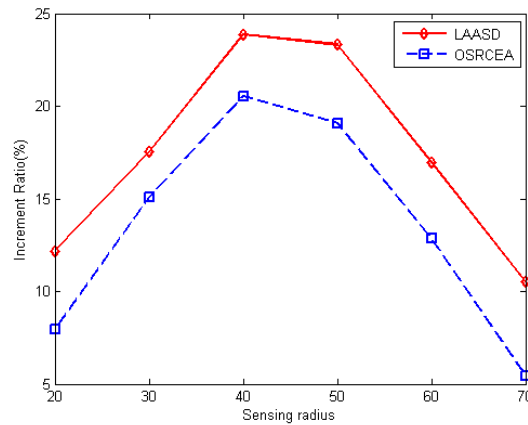


Fig. 11. Influence of sensing radius on the increment ratio

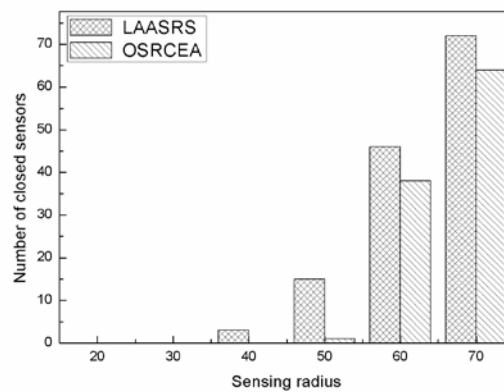


Fig. 12. Influence of sensing radius on the number of closed sensors

### 6.3 Impact of the sensing Fov

The sensing Fov angle is ranged between  $15^\circ$  and  $105^\circ$  with incremental step  $15^\circ$ , the number of sensors is 200, and the sensing radius is 60m. Similar to Experiment 2, Fig. 13 shows that LAASD achieves better increment ratio than OSRCEA. Base on analysis of increase of sensing Fov, it can be seen that the sensing region of sensor could fall outside monitoring region with a high probability. LAASD is proposed to reduce IEFOV and EOSA of sensor by adjustment of its working direction, but algorithm OSRCEA only focus on EOSA of sensor and neglect the IEFOV of sensor so that it achieves a worse performance than LASSD in coverage enhancement. When the Fov angle is in the range of  $15^\circ$  and  $75^\circ$ , the increment ratio of LAASD is more than 20%. In order to achieve a coverage growth ratio of 10% in algorithm OSRCEA, the range of Fov angle should be  $35^\circ$  and  $50^\circ$ .

Fig. 14 indicates the effect of different sensing Fov on the number of closed sensors. When the sensing Fov is greater than  $45^\circ$ , LAASRS obtains better performance in shutting off redundant sensors than OSRCEA. With the increase of sensing Fov, more sensing blanks may be appeared after using OSRCEA to closing redundant sensors with highest priorities simultaneously.

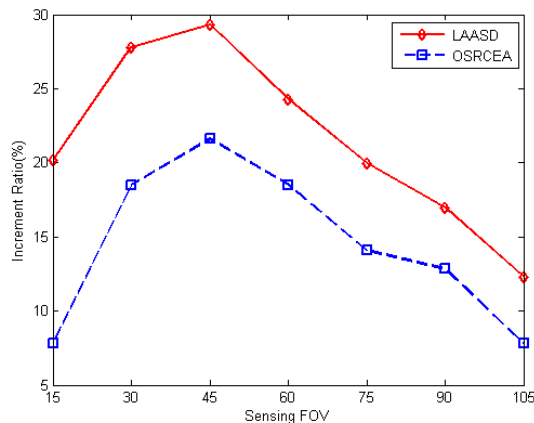


Fig. 13. Influence of sensing FOV on the increment ratio

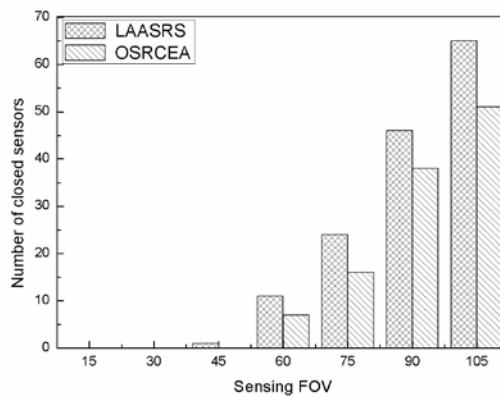


Fig. 14. Influence of sensing Fov on the number of closed sensors



## 7. Conclusion

In this paper, a learning automata-based algorithm for area coverage is presented, which consists of two phrases: area coverage enhancement(LAASD) and sleeping redundant sensors(LAASRS). In order to solve this problem more effectively, some new concepts(EFOV, IEFOV, EOSA and OSR) are proposed to measure the effects of neighboring sensors. In the phrase of LAASD, the selection of sensing directions of sensor is obtained according to the OSR. Subsequently, For closing more redundant sensors to save more energy, the second phrase of LAASRS intends to shut off redundant sensors based on the OSR and the current working states of sensors. Compared with OSRCEA, the proposed algorithm are more effective to enhance arec coverage and reduce more energy consumption. The simulation results demonstrate that the proposed algorithm outperform the existing algorithm in terms of coverage enhancement and energy consumption.

## References

- [1] I.F. Akyildiz, T. Melodia and K.R. Chowdhury, "A survey on wireless multimedia sensor networks," *Computer Networks*, vol. 51, no. 4, pp. 921-960, 2007. [Article \(CrossRef Link\)](#).
- [2] Y.H. Kim, Y.H. Han, Y.S. Jeong and D.S. Park, "Lifetime maximization considering target coverage and connectivity in directional image/video sensor networks," *Journal Supercomputing*, vol. 65, no.1, pp. 365-382, 2013. [Article \(CrossRef Link\)](#).
- [3] 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. [Article \(CrossRef Link\)](#).  
M.A. Guvensan and A.G. Yavuz, "On coverage issues in directional sensor networks: A survey," *Ad Hoc Networks*, vol. 9, no. 7, pp. 1238-1255, 2011. [Article \(CrossRef Link\)](#).
- [4] M. Cardei and D.Z. Du, "Improving wireless sensor network lifetime through power aware organization," *Wireless Networks*, vol. 11, no. 3, pp. 333-340, 2005. [Article \(CrossRef Link\)](#).
- [5] D. Zorbas, D. Glynos, P. Kotzanikolaou and C. Douligeris, "Solving coverage problems in wireless sensor networks using cover sets," *Ad Hoc Networks*, vol. 8, no. 4, pp. 400-415, 2010. [Article \(CrossRef Link\)](#).
- [6] H.P. Gupta, S.V. Rao and V. Tamarapalli, "Analysis of Stochastic k-Coverage and Connectivity in Sensor Networks With Boundary Deployment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 1861-1871, 2015. [Article \(CrossRef Link\)](#).
- [7] H. Mohamadi, A.S. Ismail, S. Salleh and A. Nodhei, "Learning automata-based algorithms for finding cover sets in wireless sensor networks," *Journal of Supercomputing*, vol. 66, no. 3, pp. 1533-1552, 2013. [Article \(CrossRef Link\)](#).
- [8] H. Mostafaei and M.R. Meybodi, "Maximizing lifetime of target coverage in wireless sensor networks using learning automata," *Wireless Personal Communications*, vol. 71, no. 2, pp. 1461-1477, 2013. [Article \(CrossRef Link\)](#).
- [9] H. Mostafaei, "Stochastic barrier coverage in wireless sensor networks based on distributed learning automata," *Computer Communications*, vol. 55, no. C, pp. 51-61, 2015. [Article \(CrossRef Link\)](#).
- [10] H. Mohamadi, S. Salleh, M.N. Razali and S. Marouf, "A new learning automata-based approach for maximizing network lifetime in wireless sensor networks with adjustable sensing ranges," *Neurocomputing*, vol.153, pp. 11-19, 2015. [Article \(CrossRef Link\)](#).
- [11] M. Esnaashari and M.R. Meybodi, "A learning automata based scheduling solution to the dynamic point coverage problem in wireless sensor networks," *Computer Networks*, vol. 54, no. 14, pp. 2410-2438, 2010. [Article \(CrossRef Link\)](#).
- [12] H. Mostafaei, A. Montieri, V. Persico and A. Pescape, "A sleep scheduling approach based on learning automata for WSN partial coverage," *Journal of Network & Computer Applications*, vol. 80, pp. 67-78, 2017. [Article \(CrossRef Link\)](#).

- [13] M. Li, C. Miao and C. Leung, "A Coral Reef Algorithm Based on Learning Automata for the Coverage Control Problem of Heterogeneous Directional Sensor Networks," *Sensors*, vol. 15, no. 12, pp. 30617-30635, 2015. [Article \(CrossRef Link\)](#).
- [14] Y. Cai, W. Lou, M. Li and X.Y. Li, "Target-Oriented Scheduling in Directional Sensor Networks," in *Proc. Of IEEE INFOCOM*, vol. 58, no. 9, pp. 1550-1558, May 6-12, 2007. [Article \(CrossRef Link\)](#).
- [15] Y.H. Kim, Y.H. Han, Y.S. Jeong and D.S. Park, "Lifetime maximization considering target coverage and connectivity in directional image/video sensor networks," *Journal of Supercomputing*, vol. 65, no. 1, pp. 365-382, 2013. [Article \(CrossRef Link\)](#).
- [16] J. Wang, C. Niu and R. Shen, "Priority-based target coverage in directional sensor networks using a genetic algorithm," *Computers and Mathematics with Applications*, vol. 57, no.11-12, pp. 1915-1922, 2009. [Article \(CrossRef Link\)](#).
- [17] M.M. Islam, M. Ahasanuzzaman, M.A. Razzaque and M. Hassan, "A distributed clustering algorithm for target coverage in Directional Sensor Networks," in *Proc of IEEE APWiMob*, pp. 42-47, August 28-30, 2014. [Article \(CrossRef Link\)](#).
- [18] H. Mohamadi, A.S. Ismail and S. Salleh, "A learning automata-based algorithm for solving coverage problem in directional sensor networks," *Computing*, vol. 95, no. 1, pp. 1-24, 2013. [Article \(CrossRef Link\)](#).
- H. Mohamadi, S. Salleh and A.S. Ismail, "A Learning Automata-Based Solution to the Priority-Based Target Coverage Problem in Directional Sensor Networks," *Wireless Personal Communications*, vol. 79, no. 3, pp. 2323-2338, 2014. [Article \(CrossRef Link\)](#).
- [19] H. Mohamadi, A.S. Ismail and S. Salleh, "Utilizing distributed learning automata to solve the connected target coverage problem in directional sensor networks," *Sensors and Actuators A Physical*, vol. 198, no. 16, pp. 21-30, 2013. [Article \(CrossRef Link\)](#).
- [20] Z.M. Liu, G.J. Wang and W.J. Jia, "Coverage Prediction Model and Number Estimation for Directional Sensor Networks," *Journal of Software*, vol. 27, no. 12, pp. 3120-3130, 2016. [Article \(CrossRef Link\)](#).
- [21] W. Cheng, S. Li, X. Liao, C. Shen and H. Chen, "Maximal coverage scheduling in randomly deployed directional sensor networks," in *Proc. Of IEEE Parallel Processing*, pp. 68-73, September 10-14, 2007. [Article \(CrossRef Link\)](#).
- [22] N. Tezan and W. Wang, "Self-Orienting wireless multimedia sensor networks for maximizing multimedia coverage," in *Proc. Of IEEE ICC*, vol. 52, no. 13, pp. 2206-2210, May 19-23, 2008. [Article \(CrossRef Link\)](#).
- [23] D. Tao, H.D. Ma and L. Liu, "A virtual potential field based coverage-enhancing algorithm for directional sensor networks," *Journal of Software*, vol. 18, no. 5, pp. 1152-1163, 2007. [Article \(CrossRef Link\)](#).
- [24] J. Zhao and J.C. Zeng, "A Virtual Centripetal Force-Based Coverage Enhancing Algorithm for Wireless Multimedia Sensor Networks," *IEEE Sensors Journal*, vol. 10, no.8, pp. 1328-1334, 2010. [Article \(CrossRef Link\)](#).
- [25] J. Chen, L. Zhang and Y. Kuo, "Coverage-Enhancing Algorithm Based on Overlap-Sense Ratio in Wireless Multimedia Sensor Networks," *IEEE Sensors Journal*, vol. 13, no. 6, pp. 2077-2083, 2013. [Article \(CrossRef Link\)](#).
- [26] K.S. Narendra and M.A.L. Thathachar, *Learning Automata: An Introduction*, Prentice-Hall, New York, 2012. [Article \(CrossRef Link\)](#).
- [27] M.A.L. Thathacha and P.S. Sastry, "A hierarchical system of learning automata that can learn the globally optimal path," *Information Sciences: An International Journal*, vol. 42, no. 2, pp. 143-166, 1987. [Article \(CrossRef Link\)](#).
- [28] B. Hamid and M. Mohammad, "Cellular learning automata with multiple learning automata in each cell and its applications," *IEEE Transactions on System Man and Cybernetics*, vol. 40, no. 1, pp. 54-65, 2010. [Article \(CrossRef Link\)](#).



**Zhimin Liu** born in 1983. He received the B.E. degree from Northeastern University, M.E. degree from South China Agricultural University in 2006 and 2010 respectively. Now he is a doctoral student, in school of Information Science and Engineering, Central South University, Changsha, China. His current interests include wireless multimedia sensor network.



**Zhangdong Ouyang** born in 1981. He received the B.E. degree from Hunan University of Arts and Science, M.E. degree and Ph.D. degree from Hunan Normal University, in 2005, 2008, 2011, respectively. He is an associate professor of School of Mathematics and Computational Science at Hunan First Normal University. His research interests include topological graph theory, graph theory algorithm and complex network, etc.