Sensor Deployment for Wireless Sensor Networks: A Conjugate Learning Automata-Based Energy-Efficient Approach

Chong Di, Fangqi Li, and Shenghong Li

***Abstract*—Wireless sensor networks (WSNs) boost the development of the Internet of Things (IoT) by its ability to monitor environments and its flexibility. It is desirable to study an efficient configuration of a WSN that balances its energy consumption and its functionality. In this article, we propose to formulate the sensor deployment task as a combinatorial optimization problem and introduce an effective sensor deployment paradigm in which both the randomness and the dynamics of the environment are captured. Following the activity scheduling mechanism, we adopt a powerful non-associative reinforcement learning method, conjugate learning automata (CLA), to learn the optimal sensor deployment strategy. Compared with conventional methods, the proposed CLA-based sensor deployment method yields good performance by activated only a subset of all sensors and does not lean into prior expertise about the environments. Meanwhile, the learning process is efficient and thus the energy is saved in multiple aspects. Comprehensive experiments under different settings demonstrate the effectiveness of the proposed method.**

I. Introduction

The past decade has witnessed the revolutionary emergence of wireless communication technologies [1]. Nowadays, users, developers, and researchers are confident to assert that wireless communication technologies can boost the application of the Internet of Things (IoT), which is going to become an indispensable part of our life [2]. By connecting electronic devices, people and the environment in a smarter way, IoT provides much convenience for our daily life, however, it also demands more intelligent ways of data processing and more resources including transducers, terminals, and servers.

One of the attempts to handle the challenges released by IoT is wireless sensor networks (WSNs) [3]. A WSN is a collection of sensors connected wirelessly so that they can be physically deployed in a flexible pattern. Information collected from these sensors is processed and reflected by the modification of the deployment of the sensors that achieves better overall performance. For example, some sensors might be activated to replace the malfunctioned ones, some sensors might be turned off to save energy. Its flexibility, adaptivity, and efficiency make WSN a promising technology to monitor an arbitrary complex environment for an IoT system. A typical architecture of a WSN as Figure. 1 (a) consists of spatially distributed sensors and a base station. A sensor, as a data originator as well as a data router, monitors some task-oriented physical variables such as the temperature, the magnetic field intensity or the fluid velocity, and returns data to the base station. The base station collects data from sensors and processes them.

C. Di, F. Li, and S, Li (Corresponding author) are with the School of Electronic Information and Electrical Engineering, Shanghai Jiao Tong University, Shanghai, China.

Despite all its privileges, the organization of a WSN, during which a massive number of compact devices are distributed and controlled across a vast area, is a challenge in itself. A good sensor deployment strategy activates only a limited number of sensors without impairing the coverage, the connectivity, the reliability and the lifetime of the network. In many practical scenarios, sensors do not change their locations once they have been deployed, so the problem reduces to finding a method that activates a subset of all sensors so that the utility and the efficiency of the network are jointly optimized.

There have been various deployment approaches with different problem formulations and concerns that fall into two categories: random deployment and deterministic deployment. In random deployment, positions of the sensors are defined by a probability measure in the region of interest (RoI). Though the random deployment is convenient in implementation, its efficacy is sometimes unsatisfactory due to the complexity and intractability of the environment. The deterministic deployment is usually more reliable as sensors are placed at predetermined locations to lift the overall efficacy. However, its optimality relies on prior expertise about the environment which is hardly accessible in a new environment, e.g. an emerging community. A further concern is that in reality, an environment varies over time, and the individual capability of a sensor may change accordingly, such dynamics puts forwards another challenge against deterministic deployment strategies [4].

To handle the difficulties in sensor deployment for WSN, we propose a novel solution by adopting a machine learning paradigm. The objective is to adaptively find the optimal deployment of sensors with partial information from the environment. This task is challenging as the difficulty is three-folded: (1) the environment is time-varying; (2) the feedback from the environment is probabilistic; (3) the solution must be energy-efficient rather than exhaustive.

Considering the challenges above, we map the sensor deployment problem into a combinatorial optimization problem, as finding the optimal combination of activated sensors is equivalent to activating a subset of sensors deployed in all possible places to achieve the best functionality. However, the exact solution to such a problem, especially for a large number of sensors, is almost impossible to obtain as the state space of the solution grows exponentially with the number of sensors. Fortunately, the recent development of artificial intelligence

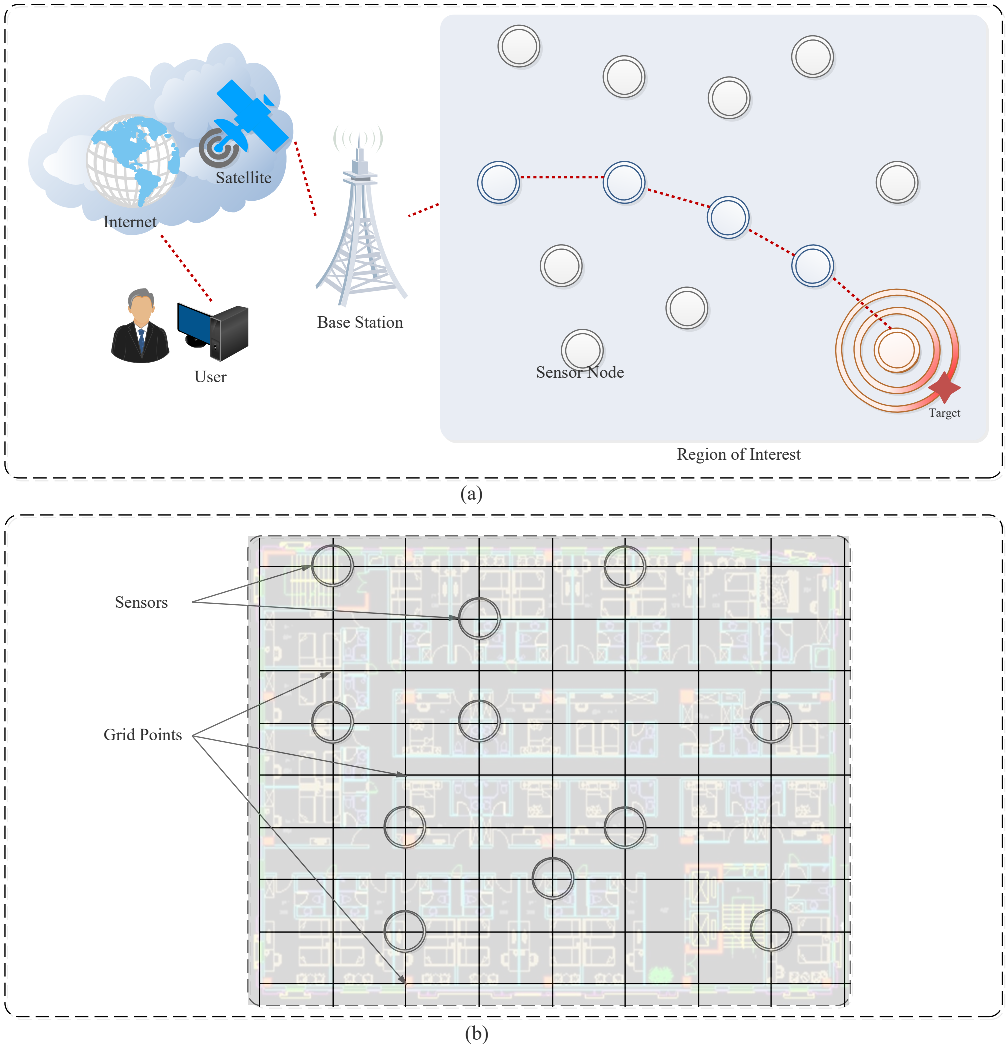

Fig. 1. A typical WSN architecture for target detection. It consists of a base station and a collection of wirelessly connected sensors deployed in a region of interest. Sensors send data to the base station when an interested target is detected, and then the base station communicates with users. (b) is an abstract version of the RoI and deployed sensors.

(AI) methodologies has brought significant breakthroughs for such problems. Several combinatorial optimization problems have been studied in the AI community and state-of-the-art performance is achieved. It has been reported that learning automaton (LA), a classical paradigm in reinforcement learning, can cooperate to achieve combinatorial optimization tasks with acceptable time complexity. As a self-adaptive unit, LA can learn the most appropriate action/decision in a stochastic environment [5]. Inspired by the latest results in this domain, we propose a conjugate learning automata (CLA) [6] based solution to address the sensor deployment problem. When a collection of LAs are trained jointly, the CLA can find the solution to the combinatorial optimization problem. Based on this fascinating property, we elaborate on a framework for determining the energy-efficient deployment of sensors that supports high-quality WSN-based services.

In the next section we review some preliminaries in the sensor deployment in WSN, in section III we propose our solution based on CLA, section IV presents experimental results and section V concludes this paper.

II. Sensor Deployment in Wireless Sensor Networks

A. Metrics of Sensor Deployment

There are three crucial metrics to evaluate a sensor deployment strategy: the coverage, the network lifetime and the energy efficiency [7, 8].

1) Coverage: The coverage of a deployment strategy is the percentage of the RoI that activated sensors can cover, it is the most fundamental metric of the quality of service (QoS). The models of coverage are diversified, in this article, we adopt the *probabilistic coverage model* that captures the stochastic nature of the environment. Given the location of a sensor and a target in the RoI, the probability that the target being detected by the sensor is inversely proportional to the distance between them. The coverage is then measured by the percentage of detected targets among all appeared targets.

2) Network Lifetime: Network lifetime is another major concern in the design of a WSN. From a practical point of view, a deployed sensor may encounter ineluctable damages and fails to work. Therefore a good deployment strategy should be responsive in the sense that even some sensors fail to function, the coverage of the entire network would not decline sharply. This could be done by activating backup sensors. In this manner we do not risk the sustainability of the network on the shortest lifetime of an individual sensor, so the network can provide more reliable and robust service.

3) Energy Efficiency: The energy consumption during the deploying process is expected to be as little as possible. We further assume that all activated sensors consume energy with an identical velocity, so the problem reduces to preserving good coverage while activating as few sensors as possible.

B. Paradigm of Energy-Efficient Sensor Deployment

Following the discussions above, we propose an energy-efficient deployment inspired by the activity scheduling mechanism proposed by [9], in which a vast number of sensors are distributed into the RoI but only a small subset of them is activated so energy is saved. Figure. 1 (b) illustrates an RoI discretized by a grid where any intersection of axes is a potential position for a sensor. It should be noted that the heterogeneity of the RoI, the heterogeneity of the probability density of the emergence of targets and the change of the environment can be simulated by distorting the grid or randomly erasing some sensors from their cells. So this simplification hardly hurts the generality of the discussion hitherto.

After distributing sensors along the grid, selecting the subset of activated sensors is essentially a combinatorial optimization problem. Utilizing CLA, our strategy is capable of: (1) determining the optimal subset of active sensors efficiently; (2) deciding the alternatives of unusable sensors efficiently; (3) updating the set of activated sensors after detecting the change of the environment.

C. Related Works

To achieve energy-efficient sensor deployment while preserving good performance, aspects from sensing model to problem formulations have been well studied by forerunners.

In WSNs, a sensor's coverage is a function that evaluates its ability to detect objects. Binary coverage models used to be widely adopted because of its simplicity. In this model, the coverage function is defined as a binary indicator with a predetermined parameter, i.e., the range of effectiveness [10]. Coverage within the range is assumed to be effective while coverage outside of the range is assumed to fail. However, the binary model does not consider the stochastic nature in the real world which could disturb the sensing process. For that, the probability coverage models that are capable of capturing the stochastic nature of sensing are adopted in recent research. The sensing ability for a sensor in probability coverage models is a random variable conditioned on the characteristics of a sensor's location in the RoI [11].

WSNs have a wide range of applications including military, environmental and medical scenarios [12, 13]. Abstracting from various applications, the WSNs are formalized by point coverage, target coverage, and barrier coverage problems, where a point coverage-focused WSN aims to look for stationary objects, a target coverage-focused one seeks targets moving through the RoI while WSNs with barrier coverage missions detect the movement across a barrier [4].

The proposed energy-efficient paradigm is not limited to some specific coverage models or applications and can be generally applied in any formulation above. In this article, we adopt point coverage formulation under probability coverage models as a typical example to present the mechanism of CLA-based sensor deployment, where the coverage is captured by the percentage of detected targets (i.e., detection rate) among all emerged targets within the RoI.

III. Conjugate Learning Automata-Based Solution

A straightforward solution to this combinatorial optimization problem is the greedy scheme which activates the local optimal sensor one at a time. By locality, we mean that we take the selected activated sensors as given and look for a new sensor that maximizes the marginal increment of the coverage, i.e., the detection rate. Since the environment is stochastic, this marginal increment can only be measured by simulation. In fact, the greedy-based scheme has been generally used in deterministic deployment approaches. However, the greedy approach is not guaranteed to end up with the global optimal solution. In a similar combinatorial optimization problem, the information maximization in social networks, researchers have proved that the ratio between the outcome of greedy approaches and the optimal one is lower bounded by $(1 - \frac{1}{e})$ for submodular propagation models [14]. Thus, the greedy approach leaves plentiful space for adversary constructions that could drastically hinder the efficacy. At this point, conjugate optimization can outperform greedy ones by selecting multiple active sensors simultaneously instead of selecting the one that maximizes the coverage performance greedily. However, this line of reasoning has been suppressed by the formidable cost in energy consumption as the state space of the solution grows exponentially with the total number of sensors.

It has been reported that learning automaton (LA) can cooperate to achieve combinatorial optimization with acceptable simulation times. When a collection of LAs are trained jointly, they can avoid the pitfall designed to trap greedy methods and obtain a better performance while preserving efficiency. Thus it is intuitively feasible to achieve energy-efficient sensor deployment with conjugate trained LAs.

A. Conjugate Learning Automata

Learning automaton, inspired by the behavior modification of living organisms adjusting to the environments, is a powerful non-associative reinforcement learning method. It is an adaptive decision-maker unit that learns the optimal one among all possible choices through interactions with the environment without manual intervention. LA possesses many valuable properties, such as the simple implementation and guaranteed convergence in stationary environments that facilitate plentiful applications in various engineering areas.

An LA with its environment can be formalized by three components $A, B, \mathbf{D}$, where A is the set of optional actions, B is the set of possible feedback from the environment, and $\mathbf{D}$ is the reward matrix of the environment whose component $d_{a,b}$ is the probability that the environment returns the feedback $b \in B$ when the agent sends the action $a \in A$. An LA looks for the action with the largest expected reward by interacting with the environment. At the t-th iteration, the LA selects action $a(t)$ according to an action probability vector $\mathbf{P}(t)$. The environment receives $a(t)$ and returns the feedback $b(t)$. The LA receives $b(t)$ and updates $\mathbf{P}(t)$ into $\mathbf{P}(t+1)$ according to some specific strategy, during which the action with higher estimated reward obtains a higher probability to be chosen again. The LA gets converged and terminates learning when its probability of choosing the estimated optimal action exceeds some threshold, i.e., $\max_r \{\mathbf{P}_r\} > \mathcal{T}$, where $\mathcal{T}$ is a predefined terminal threshold. If the action on which major probability weight concentrates is the action with the highest expected reward then an LA is said to be correctly converged. As for an unstationary environment where $\mathbf{D}$ varies with time, a uniformly optimal action does not exist, neither does a correct convergence. Ordinary LA has been thoroughly studied and its almost sure convergence in stationary environment has been established as its benchmark property [5]. Formally, for an arbitrary positive ϵ, there exists an $N(\epsilon)$ such that after no less than $N(\epsilon)$ rounds of interactions, the LA converges to the optimal action with probability no less than $(1-\epsilon)$.

The CLA originates from learning automata games where multiple independent LAs co-operate or compete with each other to obtain the Nash equilibrium. It has been demonstrated that in combinatorial optimization problems such as influence maximization, CLA can overcome some coined pitfalls and produce results better than greedy solutions [6].

A CLA comprises multiple LAs. The learning of a CLA is similar to that of an individual LA. At the t-th iteration, the k-th LA selects an action $a_k(t)$ according to its probability vector $\mathbf{P}^k(t)$. The CLA subsequently combines the actions of its components to form a vector $\vec{a}(t)$ and sends it to the environment. The stochastic environment then reacts in $b(t)$ with respect to $\vec{a}(t)$. Upon receiving the feedback, each LA independently updates its probability vector.

B. CLA-Based Sensor Deployment in Static Environments

To apply CLA to the problem of sensor deployment, we define the action set, the feedback and the external environment as Figure. 2:

- *CLA structure*. We assume that no more than K sensors are to be activated simultaneously, so we have K LAs operate conjugately.
- *Action*. The action set A_k for the k-th LA is defined as all possible sensors that have been distributed in the RoI. During each iteration, each LA chooses one sensor to activate, the collection of which is the subset of activated sensors.

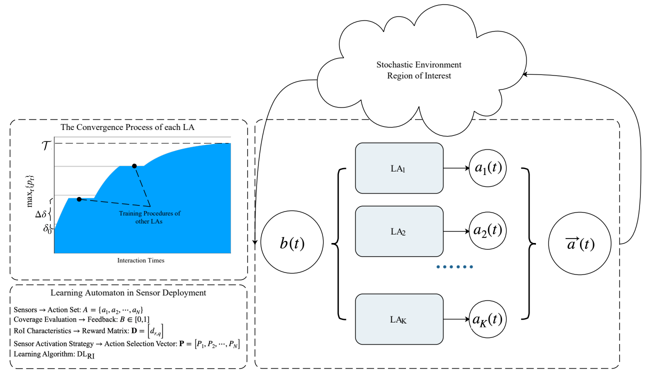

Fig. 2. Integration of CLA-based sensor deployment. CLA consists of a set of homogeneously operating LAs, whose actions form an associative action vector $\vec{\alpha}(t)$ and share the identical feedback $\beta(t)$ from the stochastic environment. Mapping the formulated sensor deployment task into a CLA learning process, LAs are jointly trained to learn the favorable subset of activated sensors.

- *Feedback*. Given the subset of activated sensors, the feedback is the result of coverage estimated by numerical simulations. Some targets are randomly generated in the RoI and the percentage of detected targets is returned.
- *Environment*. The environment of each individual LA includes not only the stochastic environment but also other automata, hence is non-stationary. Thus the ϵ-optimal convergence theorem for LA in stationary environments fails to hold. To deal with this problem, we have only one LA update its probability vector during a batch of iterations while other LAs keep their probability vector fixed. In this way each LA is trained in a stationary environment, hence stable convergence is secured. Meanwhile, we prohibit an LA to converge to its optimal action while leaving other LAs untrained because this is tantamount to greedy solutions and yields no more interest. So we set a temporary terminal threshold δ. In each batch, an LA is trained until one component of its probability vector exceeds the terminal threshold, then we start another batch and train another LA. If all K LAs have met the temporary terminal condition then we increase δ by $\Delta\delta$ and start another round of batches. This process terminates until δ is the same as $\mathcal{T}$, the left side of Figure. 2 demonstrates the change of the internal state of one individual LA.

For an individual LA in CLA, we adopt the classical $\mathtt{DL}_{\mathtt{RI}}$ scheme [5]. A $\mathtt{DL}_{\mathtt{RI}}$ updates its probability vector only if it receives a reward from the environment. Given enough rounds of interaction, it converges to the optimal behavior with respect to a stationary environment by probability.

Remarkably, CLA preserves the characteristics of both the exhaustive method and the greedy method. If the size of each batch is set to one then CLA reduces to an exhaustive search, if δ is initialized to $\mathcal{T}$ then CLA gracefully degenerates to a greedy method. The step $\Delta\delta$, by which δ gradually increases to $\mathcal{T}$ naturally reflects the trade-off between exploration (when $\Delta\delta$ is small) and exploitation (when $\Delta\delta$ is large), an innate duality in reinforcement learning.

C. CLA-Based Sensor Deployment in Dynamic Environments

Through this article, we have been considering two kinds of dynamics in sensor deployment. The first kind of dynamics is the breakdown or malfunction of an activated sensor. The second kind of dynamics is the change of the environment such as the weather and electromagnetic interference. Both of them can be reflected by the change in the expected coverage range of a given set of activated sensors. To include the dynamics of the environment into our framework, we extend the CLA into Dynamic CLA (DyCLA). Before formalizing DyCLA, consider two extreme cases: if the coverage of the current choice of active sensors is hardly affected, then we should only make a slight revision of the activated set of sensors instead of restarting from the beginning. If the coverage of the current set of activated sensors is significantly affected, then we should erase a larger portion of (potentially all) memory and restart in this new scenario.

To conclude, the dynamics of the environment results in the variation of the sensing abilities, i.e., the coverage, and is reflected by the variation of the coverage of the current set of activated sensors. Since CLA has a collection of probability vectors as a model of memory, it is straightforward to incorporate the observations from the two cases above into CLA:

1) Firstly, after the variation of the external environment takes place, the estimated coverage of the current set of activated sensors $\mathcal{S}$ changes from $\sigma(\mathcal{S})$ to $\sigma'(\mathcal{S})$. The significance of this variation is measured by the difference in detection rate $\Delta\sigma = |\sigma'(\mathcal{S}) - \sigma(\mathcal{S})|$. The function $\sigma(\mathcal{S})$ is evaluated using a series of Monte Carlo simulations, during which targets appear randomly in the ROI and sensors in $\mathcal{S}$ try to capture them under the probability coverage model. The average percentage of targets captured is the value of $\sigma(\mathcal{S})$.
2) Secondly, the convergence of any individual LA in the CLA is reversed by a parameterized *smoothing function* $f(\Delta\sigma, \cdot)$ that maps an N-dimensional simplex to another N-dimensional simplex, during which the maximal component of the input is reduced. The larger $\Delta\sigma$ is, the more closely the output turns to be a uniform distribution. Essentially, $f(\Delta\sigma, \cdot)$ is the inverse of the update process of an LA. A small $\Delta\sigma$ cancels only the influence of the latest few batches of previous learning, and CLA should be able to find another optimal solution quickly. A large $\Delta\sigma$ cancels almost all information as if CLA has just been initialized. The lower half of Figure. 3 visualizes how the smoothing function acts upon the action probability vector $\mathbf{P}$ of a specific LA.

An example of dynamics is illustrated in the upper half of Figure. 3. In case the sensitivity of a sensor decreases so that some nearby target can no longer be captured, we erase our confidence in this set of activated set of sensors and choose another set. The more target we lose, the more sensors are to be chosen from scratch.

To summarize, the CLA-based sensor deployment strategy is collected in Algorithm 1.

Algorithm 1 The CLA-based sensor deployment strategy.

1: **Input** A gridding RoI that with targets.
2: **Input** Model hyperparameters:
  the number of activated sensors K;
  the terminal threshold $(\delta, \Delta\delta, \mathcal{T})$;
  the smoothing functional $f(\cdot, \cdot)$.
3: **Training Phase**
  While $\delta < \mathcal{T}$ **do**:
    Sequentially train K LAs with terminal threshold δ.
    $\delta += \Delta\delta$.
  End while
  Output Activate the collection of sensors $\mathcal{S}$ corresponding to the largest components of K LAs' action probability vectors, the coverage is $\sigma(\mathcal{S})$.
4: **Dynamic Response Phase**
  Estimate the coverage after environment variation as $\sigma'(\mathcal{S})$.
  Evaluate the significance of environment variation as $\Delta\sigma = |\sigma(\mathcal{S}) - \sigma'(\mathcal{S})|$.
  Recover the probability vector of K LAs with $f(\Delta\sigma, \cdot)$.
  Goto the **Training Phase**.

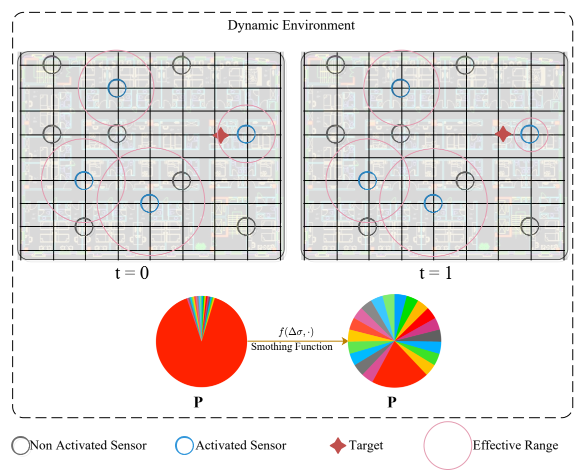

Fig. 3. CLA-based sensor deployment in a dynamic environment. An example of environment changing is illustrated, where a detected target can no longer be captured due to the decrease of the sensing ability of some specific sensor. The smoothing function then backtracks the training process of CLA as a response.

IV. Performance Evaluations

In this section, we conduct numerical simulations to verify the efficacy of our proposed CLA-based sensor deployment method. We consider WSNs of different scales, where the RoI is discretized by a two-dimensional grid. Sensors are randomly distributed at 50% of all possible cells in the RoI to simulate the geographical heterogeneity. Meanwhile, targets are generated in the whole RoI subject to a uniform probability distribution. Following the probability coverage models [10], the probability of detection of a target by a sensor varies exponentially with the distance between the target and the sensor, a target with distance d from a sensor will be detected by that sensor with probability $e^{-\alpha d}$, where α is a scaling parameter and is set to 1 uniformly. The distinctive perturbation in the stochastic environment on each sensor is represented by a proportional parameter $\mathcal{C}_n$. Thence, in our simulations, the probability that a target is detected by a sensor is given by: $\Pr = \mathcal{C}_n e^{-d}$. Note that the proportional parameter $\mathcal{C}_n$ is usually assumed to be identical in conventional works. In this article, we get rid of this assumption by allowing different values of $\mathcal{C}_n$ for different sensors, as the perturbation of stochastic environments might exert different influences on different sensors. Based on this model, we generate various scenarios for evaluating the performance of our approach. The parameters of the different experimental settings are summarized in Table I. Compared with experiments 1, 3, 5, the triplet 2, 4, 6 maintains the same sensor distribution but adopts a different set of $\{\mathcal{C}_n\}$. So pairwise comparison between them can demonstrate the strategy's ability to reuse learned knowledge against a dynamic environment.

Now we discuss the three major concerns in sensor deployment. The network lifetime is well guaranteed using the proposed paradigm wherein sensors have been distributed in all possible positions and only a subset of selected sensors are activated to enable the effectiveness of WSN. It is straightforwardly advantageous as one can dynamically switch the status of sensors to ensure longer network lifetime through energy conservation and balancing among sensors. As for the coverage and energy efficiency, we conduct the following two metrics for evaluation:

- Detection Rate. For a given sensor deployment setting, the coverage in a target-detection WSN is presented by the *detection rate* which is obtained by calculating the ratio between the number of successful detections and the total number of simulations. In this article, the number of simulations is set to $10,000$.
- Interaction Times. It is critical to minimize the number of simulated detections to achieve energy-efficient sensor deployment while determining the optimal subset of active sensors, as the status switch and detection simulation both consume energy. This metric, *interaction times* emphasizes the complexity of an algorithm.

In summary, our objective is to find the optimal subset of active sensors with predefined cardinality K which yields the highest detection rate within a few times of interaction.

As a comparison, we also solve the testing cases with the random-based approach [15], the particle swarm optimization-based method (PSODA)[16] and the greedy-based approach [17]. The idea of the applied random-based approach is to randomly select the subset of sensors to be activated. The greedy-based approach selects the *optimal* sensor one by one using Monte-Carlo simulations, which always selects the sensor with the estimated highest coverage gain at each step. The estimation results of simulation results are affected by the number of Monte-Carlo simulations, where a larger number of interactions with the environments leads to better accuracy. In our experiments, we set the number of Monte-Carlo simulations to $100, 1000, 10000$ separately to obtain comprehensive experimental results. For CLA, the learning parameters $(\delta_0, \Delta\delta, \mathcal{T})$ are set to $(\frac{1}{K}, \frac{1}{2K}, 0.999)$ and the step-size for updating the probability vector is $\frac{1}{\mathcal{R}\mathcal{N}}$, where $\mathcal{R}$ is the number of actions and $\mathcal{N} = 100K$ is the resolution parameter.

As shown in Figure.4 (a), compared with the random-based approach, CLA always has a better detection rate. While from Figure.4(a) and Figure.4(b), it can be concluded that CLA outperforms greedy-based in learning speed. For the greedy-based approach, its performance grows with the number of Monte-Carlo simulations. When the number of Monte-Carlo simulations is small (e.g. 100), the detection rate is fairly low, therefore to ensure satisfying performance, the greedy-based approach requires a large number of interactions with the environment. Meanwhile, CLA achieves the same or even higher detection rate with much fewer interactions with the environment. Taking the dynamics of external environments into consideration, simulation results shown in Figure.4(c) suggest that dynamic CLA always converges fast than naive CLA when environmental characteristics change slightly. Note that the number of extra interactions (i.e., the evaluation cost) for generating the smoothing function is set to 10000, which is added to the interaction times of dynamic CLA.

V. Summary and Future Direction

In this article, we present a novel energy-efficient CLA-based paradigm to solve the sensor deployment problem. The proposed method ensures longer network lifetime, optimal coverage as well as less energy consumption. Apart from solving the combinatorial optimization problem in static circumstances, the proposed method can also capture the changing of sensing abilities of sensors in dynamic environments and accelerate the modification of deployment.

As a beneficial research topic, there are many interesting future research directions left for consideration. One of them is the adaptive deployment of movement-assisted sensors. The emerging hardware techniques have promoted the development of movement-assisted sensors, i.e., mobile sensors, giving much flexibility to WSNs. Therefore, it would be interesting and valuable to investigate how movement-assisted sensors can be trained to achieve energy-efficient self-controlling. Moreover, many practical scenarios, e.g. surveillance systems, involve heterogeneous WSNs that integrate multiple types of sensors with different functions and capabilities. The sensor deployment approaches in such collaborative environments is also a thriving topic for future investigation.

Acknowledgment

This research work is funded by the National Nature Science Foundation of China under Grant 61971283 and 2019 Industrial Internet Innovation Development Project of Ministry of Industry and Information Technology of P.R. China Comprehensive Security Defense Platform Project for Industrial/Enterprise Networks.

TABLE I
PARAMETERS OF SIMULATED WSNS.

Index	Scale of WSN	Total Number of Sensors (N)	Number of Activated Sensors (K)	Proportional Parameter ($\mathcal{C}$)
1	6×6	18	6	U(0,1)
2	6×6	18	6	U(0,1) with a different random seed
3	8×8	32	8	U(0,1)
4	8×8	32	8	U(0,1) with a different random seed
5	10×10	50	10	U(0,1)
6	10×10	50	10	U(0,1) with a different random seed

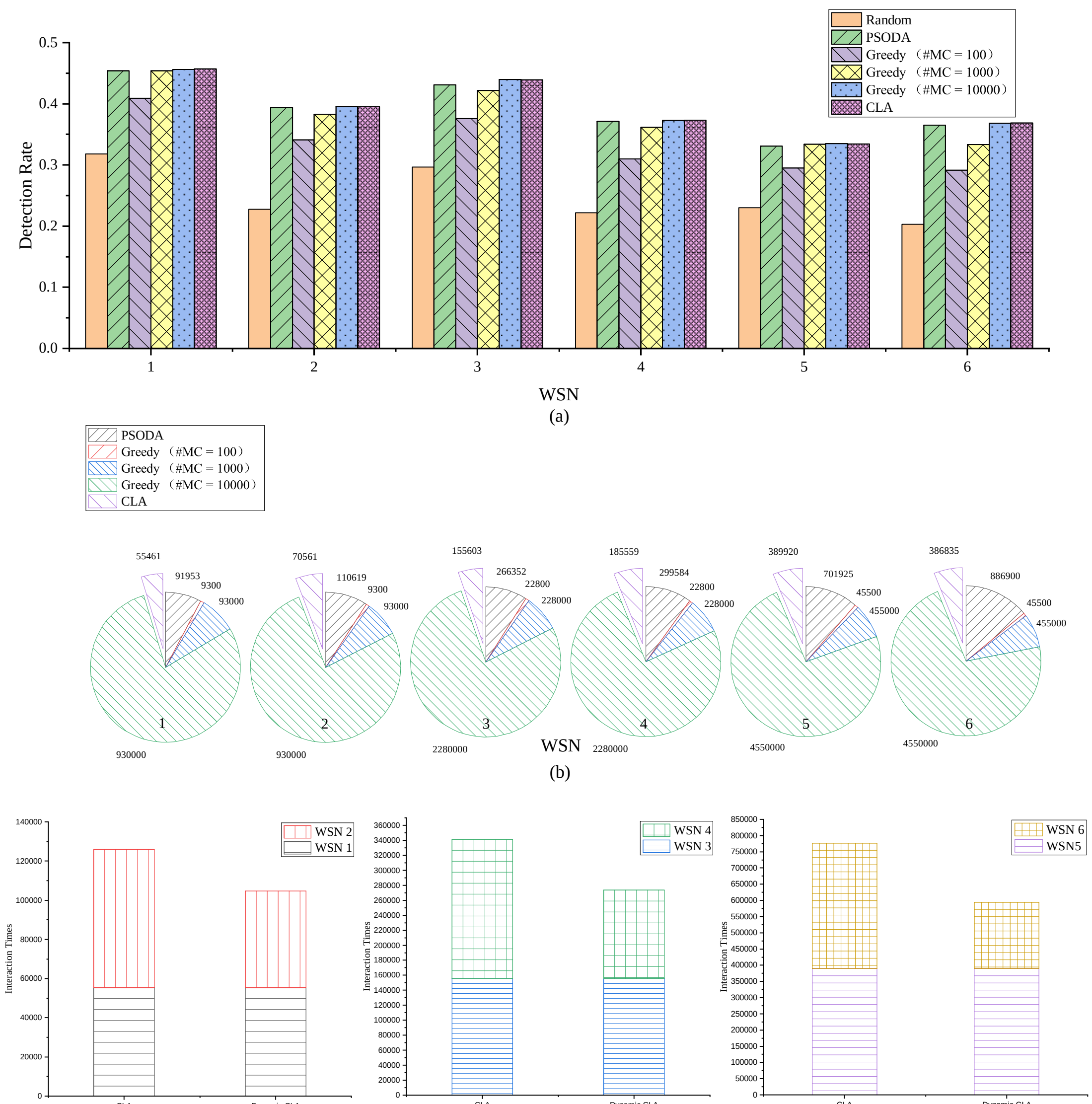

Fig. 4. Simulation results in different experimental settings. In (a), the x-axis denotes the index of WSN, and the y-axis presents the average detection rates of the compared methods. In. In (b), the pie charts show the deployment cost, i.e., the interaction times of the CLA-based approach and the greedy-based approach with different Monte-Carlo parameters. In (c), the x-axis denotes the index of WSNs, and the y-axis presents the interaction times of naive CLA-based approach and DyCLA approach in dynamic environments.

References

[1] Bi, Suzhi, Yong Zeng, and Rui Zhang. "Wireless powered communication networks: An overview." *IEEE Wireless Communications* 23.2 (2016): 10-18.

[2] Zhang, Lin, Ying-Chang Liang, and Ming Xiao. "Spectrum sharing for Internet of Things: A survey." *IEEE Wireless Communications* (2018).

[3] Al-Turjman, Fadi, and Ayman Radwan. "Data delivery in wireless multimedia sensor networks: Challenging and defying in the IoT era." *IEEE Wireless Communications* 24.5 (2017): 126-131.

[4] Senouci, Mustapha Reda, and Abdelhamid Mellouk. *Deploying wireless sensor networks: theory and practice*. Elsevier, 2016.

[5] Narendra, Kumpati S., and Mandayam AL Thathachar. *Learning automata: an introduction*. Courier Corporation, 2012.

[6] C Di, F Li, K Qi, and S Li. Maximizing Influence on Social Networks with Conjugate Learning Automata. *2019 IEEE Global Communications Conference*. IEEE, 2019.

[7] Li, He, et al. "Efficient energy transport in 60 GHz for wireless industrial sensor networks." *IEEE Wireless Communications* 24.5 (2017): 143-149.

[8] Van Hoesel, Lodewijk, et al. "Prolonging the lifetime of wireless sensor networks by cross-layer interaction." *IEEE Wireless Communications* 11.6 (2004): 78-86.

[9] Xing, Guoliang, et al. "Integrated coverage and connectivity configuration for energy conservation in sensor networks." ACM Transactions on Sensor Networks (TOSN) 1.1 (2005): 36-72.

[10] Zou, Yi, and Krishnendu Chakrabarty. "A distributed coverage-and connectivity-centric technique for selecting active nodes in wireless sensor networks." *IEEE Transactions on Computers* 54.8 (2005): 978-991.

[11] Wang, Bang. "Coverage problems in sensor networks: A survey." *ACM Computing Surveys (CSUR)* 43.4 (2011): 32.

[12] He, Daojing, Sammy Chan, and Mohsen Guizani. "Cyber security analysis and protection of wireless sensor networks for smart grid monitoring." *IEEE Wireless Communications* 24.6 (2017): 98-103.

[13] Xu, Wenjun, et al. "Data-cognition-empowered intelligent wireless networks: Data, utilities, cognition brain, and architecture." *IEEE Wireless Communications* 25.1 (2018): 56-63.

[14] Zhu, Jianming, Smita Ghosh, and Weili Wu. "Group Influence Maximization Problem in Social Networks." *IEEE Transactions on Computational Social Systems* 6.6 (2019): 1156-1164.

[15] Senouci, Mustapha Reda, Abdelhamid Mellouk, and Amar Aissani. "Random deployment of wireless sensor networks: a survey and approach." *International Journal of Ad Hoc and Ubiquitous Computing* 15.1-3 (2014): 133-146.

[16] Senouci, Mustapha R., et al. "Static wireless sensor networks deployment using an improved binary PSO." *International Journal of Communication Systems* 29.5 (2016): 1026-1041.

[17] Krause, Andreas, et al. "Optimizing sensor placements in water distribution systems using submodular function maximization." *Water Distribution Systems Analysis Symposium* 2006. 2008.