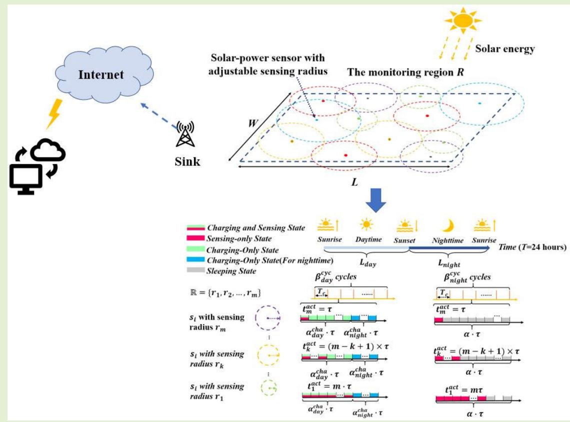
JMLA: Joint Monitoring Quality and Network Lifetime for Range-Adjustable Sensors in WRSNs

Pei Xu, Xiaofeng Wang, Chih-Yung Chang, *Member, IEEE*, Youxi Li, and Diptendu Sinha Roy

***Abstract*—Area coverage is a critical issue in wireless sensor networks (WSNs). Existing studies mainly used the Boolean sensing model (BSM) and battery-powered sensors with fixed sensing radius, potentially reducing network lifetime and monitoring quality. This article proposes an area coverage algorithm called joint monitoring quality and network lifetime with adjustable sensing radii (JMLA), which employs rechargeable sensors with adjustable sensing radii and utilizes the probabilistic sensing model (PSM) to enhance monitoring quality and perpetuate network lifetime. This will encounter several challenges: 1) more complex sensor scheduling due to the unstable of solar power; 2) more complex energy allocation because of different power consumption rates for different sensing radii; and 3) the cooperative detection between neighboring sensors resulting in different monitoring contribution when applying the PSM. To tackle these challenges, JMLA first estimates the potential solar energy available for the upcoming day. Based on this forecast, it partitions the day into multiple time slots and cycles. Sensors are scheduled to activate in different slots to balance energy consumption with the energy harvested. In addition, JMLA divides the area into several identical grids and forms multiple space–time points with each time slot. Then, sensors are scheduled to cooperatively monitor the bottleneck space–time point with the weakest cooperative detection probability, aiming to maximize the monitoring quality. Performance results show that the proposed JMLA outperforms the existing algorithms in terms of monitoring quality, stability of monitoring quality, and solar energy utilization (SEU).**



***Index Terms*— Area coverage, monitoring quality, network lifetime, range-adjustable sensor, wireless rechargeable sensor networks (WRSNs).**

NOMENCLATURE

R	Monitoring region.
S	Set of deployed sensors, $S = \{s_1, s_2, \ldots, s_n\}$.
$\mathbb{R}$	Set of selectable sensing radii of each sensor, $\mathbb{R} = \{r_1, r_2, \ldots, r_m\}$
$p(s_i, v)$	Detection probability of sensor s_i to point v.
$u_{i,k}^{\mathrm{dis}}$	Discharging rate of the sensor s_i when its sensing radius is r_k.
T_c	Length of one cycle.
τ	Length of one time slot.
$t_{i,k}^{\mathrm{act}}$	Active time of the sensor s_i when it selects the sensing radius r_k.
$c(t)$	Boolean variables represent whether the channel state is idle or busy at time t.
T_i^w	Random backoff waiting time for the sensor s_i when the channel state is busy.
B_{unit}	Backoff time unit.
$Q(\mathbb{S})$	Monitoring quality of the area coverage achieved by the algorithm $\mathbb{S}$.
$p^{\mathrm{PV}}(t)$	PV power function.
$E_{\mathrm{tot}}^{\mathrm{con}}$	Total amount of consumed energy of each sensor in one day.
$g_{(x,y)}$	Grid with the coordinates (x, y) in R.
$M_{\mathrm{met}}^{\mathrm{hist}}$	Set of historical daily meteorological data matrices $M_{\mathrm{met}}^{\mathrm{hist}} = \{\mathbb{M}_{\mathrm{met}}^{1\text{-hist}}, \mathbb{M}_{\mathrm{met}}^{2\text{-hist}}, \ldots, \mathbb{M}_{\mathrm{met}}^{365\text{-hist}}\}$.

Manuscript received 18 August 2024; accepted 25 August 2024. Date of publication 5 September 2024; date of current version 16 October 2024. This work was supported in part by the Program for Key Scientific Research Projects in Colleges and Universities of Anhui Province under Grant 2022AH051781, in part by the Program for Scientific Research Fund for Talents in Hefei University under Grant 21-22RC16, and in part by the Program for Scientific Research Innovation Team in Colleges and Universities of Anhui Province under Grant 2022AH010095. The associate editor coordinating the review of this article and approving it for publication was Prof. Chen Yang. *(Corresponding author: Chih-Yung Chang.)*

Pei Xu and Xiaofeng Wang are with the School of Artificial Intelligence and Big Data, Hefei University, Hefei 230601, China (e-mail: xupei@hfuu.edu.cn; wangxiaofeng_hfuu@126.com).

Chih-Yung Chang and Youxi Li are with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei 25137, Taiwan (e-mail: cychang@mail.tku.edu.tw; 808414022@o365.tku.edu.tw).

Diptendu Sinha Roy is with the Department of Computer Science and Engineering, National Institute of Technology Meghalaya, Shillong 793003, India (e-mail: diptendu.sr@nitm.ac.in).

Digital Object Identifier 10.1109/JSEN.2024.3451521

$\mathbb{M}_{\text{met}}^{j\text{-fore}}$	Forecast meteorological data matrix for the next day.
$E_{\text{tot}}^{\text{acq-est}}$	Estimated total amount of energy that can be acquired from solar power in the next day.
E_j^d	Energy allocated for each sensor during daytime.
E_j^n	Energy allocated for each sensor during nighttime.
α	Number of cycles in the daytime.
β	Number of cycles in the nighttime.
$\Phi_{(x,y)}^{\text{cov}}$	Set of active sensors that fully covers the grid $g_{(x,y)}$.
LMR_i^{st}	Local monitoring space–time region of sensor s_i.
$S_{(x,y),h}^{\text{sch}}$	Set of sensors that can fully cover $g_{(x,y)}$ and are scheduled to be active at the time slot t_h.
$o_{i,\text{loc}}^{\text{bott}}$	Local bottleneck space–time points in LMR_i^{st}.
$\mathbb{N}(s_i)$	Set of neighboring sensors of sensor s_i.
$M_{i,\text{loc}}^{\text{sch}}$	Matrix stores the local current sensor scheduling results.
$M_{i,\text{loc}}^{\text{QoM}}$	Matrix stores the local monitoring quality in LMR_i^{st}.
$\mathbb{S}_i^{\text{bott}}$	Set of feasible task schedules for $\mathbb{s}_i$, $\mathbb{S}_i^{\text{bott}} = \{\mathbb{s}_i^1, \mathbb{s}_i^2, \ldots, \mathbb{s}_i^m\}$.
$b^{\text{bott}}(\mathbb{s}_i^k)$	Benefit of s_i monitoring the local space–time point by adopting the feasible task schedule $\mathbb{s}_i^k \in \mathbb{S}_i^{\text{bott}}$.
$\mathbb{s}_i^{\text{best}}$	Best task schedule for sensor s_i.
$\mathbb{T}_i^{\text{wait}}$	Waiting time for s_i before initiating its monitoring task according to its task schedule.

I. Introduction

Wireless sensor networks (WSNs) have been deployed for automation of various tasks, such as environmental monitoring, intrusion detection, and intelligent farming [1], [2], [3]. The coverage problem is one of the important issues that draw a lot of attention in WSNs. It is noticed that the detection capacity and energy of each sensor are both limited, and therefore, the surveillance quality and network lifetime are two core indicators to assess the performance of the coverage problem [4] and most studies are concerned with these two issues. The coverage problem can be categorized into target coverage [5], [6], barrier coverage [7], [8], and area coverage [9], [10]. Target coverage aims usually to monitor a set of given objects, such as some important exhibition in museums and warehouses. Barrier coverage aims to surveil the boundary and detect illegal intruders. Unlike them, area coverage aims to monitor a given region, such as an agricultural test field or a chemical plant [28], without any coverage holes. This article aims to address the network lifetime and monitoring quality issues in area coverage scenarios.

The monitoring quality mainly depends on the sensing model of the sensor and the scheduling algorithm that schedules all the sensors to effectively monitor the given area. Most existing studies employed the Boolean sensing model (BSM) to design the area coverage algorithm due to its simplicity and ease of analysis, which aims to guarantee that every point in the monitoring region is covered by no fewer than the required number of sensors [10], [11]. The BSM is a deterministic model that considers the detection capability of a sensor as perfect, guaranteeing that any event occurring within its sensing region will be detected. It is noted that the sensing signal of any kind of sensor decreases with distance. The scheduling algorithm designed based on BSM may not be accurate enough because the BSM cannot reflect the physical features of sensing. Different from BSM, the probabilistic sensing model (PSM) posits that a sensor's sensing ability decreases with the increasing distance from the event, a relationship quantified by the detection probability. These studies designed algorithms based on PSM that addressed the area coverage problem and aimed to detect any point within the monitoring region with a probability greater than the predefined requirement [12], [13]. The PSM is more practical compared with the BSM because it matches real sensing behavior. In the PSM, the detection probability of a certain point, which is monitored by multiple sensors, is greater than that which is monitored by one sensor. Therefore, an effective scheduling algorithm that adopts PSM should facilitate sensor cooperation.

However, most previous studies employed sensors with fixed sensing radii. According to the PSM model, the detection capability of a sensor is enhanced as its sensing radius expands. Therefore, scheduling algorithms designed with fixed sensing radii might not be flexible enough to fully exploit the detection capacity of each sensor. In recent years, the usage of sensors with adjustable sensing radii has become more prevalent [5], [14], [15]. This poses new challenges to the sensor scheduling algorithm. With an adjustable sensing radius, sensors with different sensing radii consume different amounts of energy for the same length of time, which makes energy allocation more complex.

The network lifetime is another core indicator of coverage issue. A lot of previous studies assumed that sensors are battery-powered and developed efficient sleep-active scheduling algorithms for sensors to maximize the network lifetime [16], [17], [18]. However, the algorithms designed for extending the network lifetime of the WSNs still have the energy constraint. To overcome the energy limitation and perpetuate the network lifetime, wireless rechargeable sensor networks (WRSNs) have been widely applied, and plenty of energy-replenishing algorithms have been proposed. These algorithms can be classified into two types, including mobile charger charging [19], [20] and environmental energy harvesting [14], [21]. Although mobile chargers can provide stable and sufficient electrical energy. Area coverage is usually applied in a large outdoor area, and in some application scenarios, the terrain can present considerable complexity, leading to substantial difficulties in deploying mobile chargers for area coverage. In addition, planning paths for a large number of mobile chargers are highly complex. Therefore, environmental energy harvesting is more suitable for area coverage.

The environmental energy harvesting technology primarily derives energy from wind and solar sources. Compared to solar energy, wind energy is more unstable, and the locations with sufficient wind resources are fewer. Most existing studies have overlooked the time variation of solar power and the impact of different weather conditions on solar energy while assuming solar energy to be a constant value. Under such assumptions, there may be significant discrepancies between the energy consumption of each sensor and the actual amount of energy obtained from solar power, leading to low energy utilization efficiency or sensor failure due to battery depletion.

As discussed above, the adoption of solar-powered sensors with adjustable sensing radii poses some new challenges: 1) solar power energy is a dynamic energy resource and is affected by weather conditions, making the scheduling of sensor operations more complex; 2) sensors with different sensing radii consume varying amounts of power, complicating energy allocation due to the multiple possible schedules; and 3) the collaborative detection between sensors must be considered, as the monitoring contribution of each sensor varies with and without cooperation with neighboring sensors under the PSM. Despite these pioneering efforts, an effective and efficient algorithm for area coverage to maximize the monitoring quality while maintaining the perpetual network lifetime is still lacking.

To address these challenges and achieve the objectives of maximizing the monitoring quality while perpetuating the network lifetime, this article proposed a novel area coverage algorithm, termed joint monitoring quality and network lifetime with adjustable sensing radii (JMLA) for area coverage. The proposed JMLA algorithm utilizes solar-powered sensors with adjustable sensing radii and PSM. To address the first challenge, the JMLA initially utilizes the similarity-based calculation to select the day from the previous year that most closely matches the expected weather conditions of the next day. The algorithm then applies the photovoltaic (PV) power function from that selected day previously provided to estimate the PV power for the upcoming day, aiming to achieve an accurate estimation of potential acquired energy from solar power.

To tackle the second challenge, JMLA partitions one day into several equal-length cycles that consist of a fixed number of identical time slots. Based on the estimated acquired energy, JMLA allocates the equal available energy for each cycle. The active time of each cycle can be determined according to the corresponding sensing radius. To guarantee stable monitoring quality, each sensor has the same task schedule in every cycle.

To address the third challenge, the JMLA partitions the monitoring region into several identical grids, and the detection probability of each sensor to all points within the same grid is considered to be identical. These grids and time slots form multiple space–time points. The calculation of the cooperative detection probability of active sensors at each space–time point is straightforward. In addition, this partitioning facilitates consideration of monitoring quality in both spatial and temporal dimensions. Initially, the JMLA algorithm identifies the bottleneck space–time point with the lowest detection probability. The algorithm then prioritizes scheduling the sensor with the highest monitoring contribution among its neighbors to the bottleneck space–time point. Under this approach, the monitoring quality can be enhanced through cooperative detection.

The principle contributions of this article are itemized as follows.

1) *Considering the Variation of PV Power:* The PV power derived from solar energy is subject to instability and influenced by weather conditions. To allocate energy accurately to each sensor, the proposed JMLA algorithm utilizes weather forecast data. It identifies the day from the previous year most similar to the forecast conditions of the next day. JMLA then uses the PV power function from that selected day to estimate the solar power for the upcoming day. Based on this scheme, the algorithm can design better task schedules for each sensor with high solar power utilization.
2) *Balancing Consumed and Acquired Energy in Each Sensor:* The proposed JMLA algorithm optimizes the management of each sensor's energy by accurately calculating its available energy. Subsequently, each sensor calculates its active time under different sensing radii based on the accurately allocated available energy. Consequently, it ensures a daily balance between the total energy acquired from solar power and the total energy consumed by each sensor, thereby prolonging the network lifetime.
3) *Developing the Contribution-Based Strategy for Cooperative Sensing:* Considering the PSM, the JMLA algorithm acknowledges varying contributions of each sensor, with or without cooperation. If multiple neighboring sensors are scheduled to activate simultaneously, they are treated as a combined set. The algorithm calculates the cooperative detection probability for each sensor at each space–time point and identifies the point with the weakest detection probability. Subsequently, the sensor with the highest contribution is scheduled to monitor this bottleneck point, aiming to enhance the area's monitoring quality according to the cooperative detection between neighboring sensors.
4) *Enhancing Monitoring Quality Through Sensors With Adjustable Sensing Radii:* Given that sensors are often randomly deployed in monitoring area, sensor density varies across different areas. Areas with high deployment density may have redundant detection capabilities. To leverage this redundancy, the proposed JMLA algorithm dynamically adjusts the sensing radius and activation time for each sensor through a tailored task schedule, enhancing the utilization of detection capability. As a result, the JMLA algorithm achieves superior monitoring quality compared to existing algorithms that utilize sensors with a fixed sensing radius.

The remainder of this article is organized as follows. Section II summarizes the related work. Section III formalizes the assumptions and problem statement of the investigated issue. In Section IV, the JMLA algorithm is introduced. Section V presents the performance study and Section VI gives conclusions and future work of this article.

II. Related Work

In literature, several area coverage algorithms have been proposed in recent years. In general, these studies can be partitioned into four categories: WSNs with fixed sensing radii sensors, WSNs with adjustable sensing radii sensors, WSNs with battery-powered sensors, and WSNs with rechargeable sensors. The following reviews these studies and compares them with our work.

A. WSNs With Fixed Sensing Radii Sensors

Because the fixed sensing radius sensor is comparatively easier to analyze and design the coverage algorithm, several studies applied the fixed sensing radius to investigate the barrier coverage issue, several studies applied the fixed sensing radius to investigate the barrier coverage issue. Tossa et al. [10] proposed a genetic algorithm-based strategy, called genetic algorithm for area coverage maximization (GAFACM), for optimal sensor node placement in WSNs, aimed at maximizing both coverage and connectivity. The approach began with a linear programming (LP) formulation of the problem, followed by detailed descriptions of genetic procedures such as chromosome representation, selection, crossover, and mutation. In addition, it developed a comprehensive fitness function derived from the percentage of coverage and connectivity metrics between sensors. However, the algorithm proposed in this study is the centralized algorithm, which leads to a high control overhead. Gao et al. [11] addressed the coverage problem in mobile sensor networks, aiming to minimize sensor movement while ensuring continuous target coverage in an Euclidean space. Initially, sensors were stationed at k base stations, termed k-sink minimum movement target coverage. A polynomial-time approximation scheme, energy effective movement algorithm (EEMA), was proposed, with an approximation ratio of $1 + \varepsilon$ and a time complexity of $n^{O(1/\varepsilon^2)}$. In addition, a distributed version (D-EEMA) was introduced for large-scale networks. Both of the abovementioned studies adopted the BSM, which might not be accurate in real applications. Some other studies adopted the PSM to design the barrier coverage algorithm.

Tossa et al. [10] first defined the concept of the safe cell and then designed algorithms to find safe cells and barrier gaps. Second, they designed an algorithm to improve the barrier coverage with the minimum number of mobile sensors. Finally, they applied the Kuhn–Munkres (KM) algorithm to solve the minimum movement of mobile sensors problem in the general case. The abovementioned studies assumed that the sensors are densely deployed. Yang et al. [13] investigated the full area coverage issue by using the PSM in sensor networks. They investigated the mathematical relation between the two adjacent points in the given area. According to this assumption, the desired area coverage is transformed into a point coverage. Thereby, they designed the area coverage algorithm by dynamically selecting a group of sensors. Although sensors with a fixed sensing radius are easier to analyze, the fixed sensing radius lacks flexibility, which limits the full utilization of each sensor's detection capabilities, especially when sensors are randomly deployed.

B. WSNs With Adjustable Sensing Radii Sensors

To further improve the performance of WSNs, some studies adopted sensors with adjustable sensing radii. Chen et al. [5] addressed the coverage problem of WSNs by considering the range-adjustable sensors. Based on the BSM, they proposed a neighborhood-based estimation of distribution algorithm (NEDA) to address it in a recursive manner. In NEDA, each individual represented a coverage scheme in which the sensors were selectively activated to monitor all the targets. An LP model was built to assign activation time so that the network lifetime can be maximized, while the surveillance quality can be guaranteed. However, this study is a centralized algorithm. Fan et al. [14] investigated the coverage problem in directional sensor networks (DSNs), considering the nodes' capability to adjust their working directions and sensing ranges. This article proposed a barrier construction scheme that schedules nodes to collaboratively form multiple coverages, determining their optimal working directions and sensing ranges, which extends the service lifetime of the network. However, directional sensors, such as cameras, represent a specialized category of sensors. As they are not universally prevalent, the scheme proposed in this article may not be suitable for the majority of scenarios. Dande et al. [15] presented a decentralized area coverage algorithm, called maximizing the surveillance quality of area coverage (MSQAC), integrating rechargeable sensors with solar energy to enhance surveillance quality. It tackled energy utilization imbalances and sensor cooperation challenges using the PSM. MSQAC optimized sensor scheduling to maximize monitoring quality by balancing energy availability and cooperation among neighboring sensors. However, this study did not consider the time variation of solar power and the impact of weather conditions on solar power. However, this study did not consider the time variation of solar power and the effect of weather conditions on solar power.

C. WSNs With Battery-Powered Sensors

For the WSNs with battery-powered sensors, the energy is limited, and therefore, most of the studies aim to develop energy conservation or energy-efficient schemes to extend the network lifetime. Mostafaei et al. [16] focused on the problem of partial coverage, crucial for scenarios where continuous monitoring of a limited area was sufficient. This study introduced a novel algorithm, partial coverage with learning automata (PCLA), which utilized learning automata to manage sensor sleep scheduling. The primary goal was to minimize sensor activation while ensuring connectivity among the sensors to cover the desired area effectively. Building on the advancements in partial coverage, Yu et al. [17] addressed the more complex k-coverage problem. The aim was to select a minimal subset of nodes such that each point in a target region was covered by at least k nodes. This study developed both centralized and distributed protocols, grounded in the innovative concept of coverage contribution area. This approach not only facilitated a reduction in sensor spatial density but also incorporated the residual energies of the sensors. Consequently, these protocols significantly extended the network's lifetime, a critical advancement given the energy constraints and recharging challenges inherent to WSNs. Further expanding on these concepts, another aspect of [18] tackled the maximum coverage set scheduling problem. This involved finding feasible scheduling for the coverage set collection to maximize network lifetime. The problem was formulated as an integer LP problem, and an approximation algorithm with a theoretical performance guarantee was proposed. This study also highlighted the potential enhancements brought by mobile sensors in improving the performance of barrier coverage, although the persistent issue of energy limitations continued to constrain the overall performance of WSNs. With the development of mobile sensors, some other studies further designed algorithms by adopting mobile sensors to improve the performance of the barrier coverage. However, the energy constraint still limits the performance of WSNs.

D. WSNs With Rechargeable Sensors

To address the power limitations in WSNs, various studies developed recharging algorithms for mobile chargers in WSNs. These kinds of studies are typically categorized into two main types: offline and online. For the offline type, mobile chargers operate on a planned recharging route, determined by the recharge requests from sensors with low energy levels. Kan et al. [19] proposed an energy recharging mechanism, called energy charging mechanism for surveillance quality, network connectivity and perpetual lifetime in WRSNs (CSCP), which adopted the mobile charger to recharge the sensors according to the recharging requests. The proposed CSCP considered the contribution of each grid in terms of network connectivity, surveillance quality, as well as path cost, aiming to recharge the grids where the sensors could maximize the surveillance quality. Different from the offline recharging algorithms, the online ones dynamically adjusted the recharging schedule based on the received recharging requests from sensors in real time. Shang et al. [20] proposed a novel reinforcement learning strategy, reinforcement learning recharging (RLR), for optimizing sensor coverage in WRSNs. It employed reinforcement learning to analyze energy trends, recharging costs, and coverage benefits. The mobile charger prioritized sensors with high independent coverage and proximity, enhancing overall network efficiency. However, the large-scale deployment of mobile chargers leads to a very high cost, and the charging path scheduling of mobile chargers is complex.

Environmental energy harvesting is another category of energy replenishing algorithm. Chen et al. [21] proposed a reinforcement learning-based sleep scheduling for coverage (RLSSC) algorithm for solar-powered WSNs. It combined a two-stage sleep scheduling algorithm integrating precedence operator-based group formation and Q learning-based active node selection. Factors, such as precedence operator, residual energy, node recharging frequency, and environmental states, were considered. However, this study did not consider the monitoring quality of area coverage. Dande et al. [15] proposed an area coverage algorithm called MSQAC, which considers the rechargeable sensors and utilizes solar-powered energy. The proposed mechanism aims to maximize the surveillance quality for a given monitoring area by applying the PSM. However, the time variation of solar power and the effect of weather conditions on solar power were not considered in this study.

Table I gives a comparison of the main characteristics of the proposed JMLA algorithm and the existing works.

III. Preliminary and System Molde

The considered network model for the WRSNs and the problem statement of this article are discussed in this section.

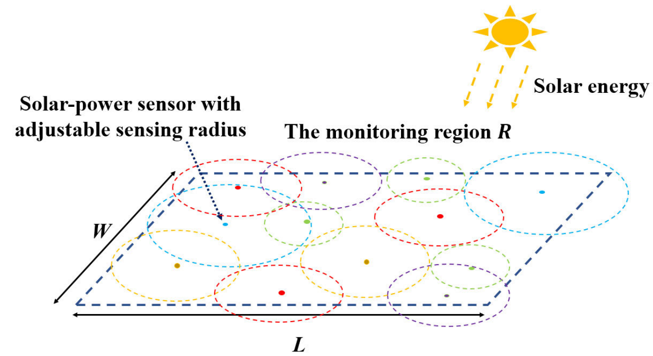


Fig. 1. Scenario of the considered WRSNs.

A. Network Environment

This article considers a rectangle region R with the size $L \times W$, where L and W are the length and width of R, respectively. A set of solar-powered sensors $S = \{s_1, s_2, \ldots, s_n\}$ are randomly deployed in R. The sensing radius of each sensor is identical and adjustable with several fixed levels. Let $\mathbb{R} = \{r_1, r_2, \ldots, r_m\}$ denote the set of selectable sensing radii of each sensor, where $r_i < r_j$ if $i < j$. The power consumption rate and detection capability of each sensor will be increased with its sensing radius. Each sensor has a unique ID and is aware of its location, remaining energy, and the boundaries of R. Let (x_i, y_i) represent the coordinate of the sensor s_i. The communication radius of each sensor is denoted by r_c. In general, the communication radius is at least twice the maximum sensing radius, that is, $r_c \geq 2r_m$.

In the considered WRSNs, channel characteristics significantly impact the overall network performance. Among these, the packet drop rate is one of the key factors affecting the overall performance of WRSNs, especially in area coverage algorithms designed for long-term monitoring such as the proposed JMLA algorithm. A high packet drop rate can significantly degrade monitoring quality as crucial monitoring data may fail to be transmitted reliably [26]. For the JMLA algorithm, which relies on cooperative detection among multiple sensors to enhance monitoring quality, packet drop reduces the network's collaborative detection probability at various space–time points, ultimately decreasing the overall monitoring quality. In addition, a high packet drop rate leads to repeated transmission attempts and increased communication energy consumption, which reduces the energy available for sensing operations. This further diminishes the sensors' detection capabilities.

Fig. 1 gives a scenario of the considered network.

B. Sensing Model

As shown in Fig. 2, the coverage area of the sensor s_i can be divided into two subareas, the inner area is the guaranteed detecting area, and the outer area is the probabilistic detecting area. Let A_i^g and A_i^p denote the guaranteed detecting area and probabilistic detecting area, respectively. The radii of A_i^g and A_i^p are r_g and r_m, respectively. If a point v falls in A_i^g, the detection probability of s_i to v is 100%. In contrast, if a point v falls in A_i^p, the detection probability of s_i to v is reduced

TABLE I
COMPARISONS OF THE PROPOSED JMLA AND RELATED WORKS

Related works	*PSM*	Adjustable sensing radius sensor	Rechargeable sensors	Considering the variation of solar power	Maximizing the monitoring quality
[5]	×	○	×	×	×
[10]	×	×	×	×	○
[11]	×	×	×	×	×
[12]	○	×	×	×	×
[13]	○	×	×	×	×
[14]	○	○	×	×	×
[15]	○	×	○	×	○
[16]	×	×	×	×	×
[17]	×	×	×	×	×
[18]	×	×	×	×	×
[19]	×	×	○	×	×
[20]	×	×	○	×	×
[21]	×	×	○	×	×
JMLA	○	○	○	○	○

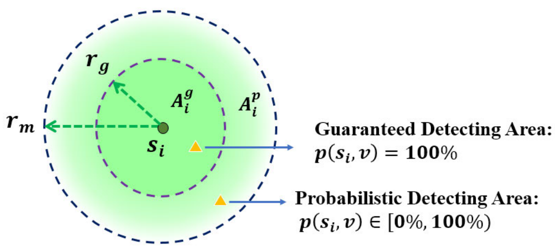


Fig. 2. Logic view of the PSM.

based on the distance between them, ranging from 100% to 0%.

Let $p(s_i, v)$ denote the detection probability of s_i to v. The value of $p(s_i, v)$ can be calculated by applying the following equation:

$$p(s_i, v) = \begin{cases} 1, & d(s_i, v) \le r_g \\ e^{-\lambda(d(s_i,v)-r_g)^{\gamma}}, & r_g < d(s_i, v) \le r_m \\ 0, & d(s_i, v) > r_m \end{cases} \quad (1)$$

where λ and γ represent the path loss exponents of the sensing signal strength of the sensor, respectively.

In the considered WRSNs, path loss is another crucial channel characteristic that impacts the overall performance of the proposed algorithm. Path loss refers to the attenuation of signal energy as it propagates through space, which directly affects the communication range and quality between sensors. Higher path loss requires sensors to use increased transmission power or to attempt communications more frequently, both of which can lead to higher energy consumption [27]. Consequently, this reduces the amount of energy available for sensing, thereby degrading monitoring quality. The PSM utilized in the JMLA algorithm depends on (1), in which the parameters λ and γ are key to impact the attenuation model of path loss. The parameter λ impacts the rate at which signal strength decreases with distance; γ impacts the shape of the attenuation function, further refining the dynamic range of signal attenuation.

To specifically investigate the impact of channel characteristics on area coverage performance in the considered WRSNs, this article extensively discusses the effects of packet drop rate and path loss on the overall performance of the JMLA algorithm according to the simulation experiments in Section V.

C. Charging and Discharging Model

In the considered WRSNs, all sensors are solar-powered, which has four possible states. During the daytime, each sensor can stay in one of two states: charging and sensing state or charging-only state. Because there is no solar energy during nighttime, each sensor can stay in either sensing-only state or sleeping state.

Sensors consume energy when they are in either charging and sensing state or sensing-only state. Sensors operating in these two states are referred to as active sensors. In the charging and sensing state, each sensor can perform charging and sensing operations simultaneously. In addition, sensors can be charged in both the charging and sensing state and the charging-only state. Conversely, sensors in the sleeping state do not perform any operation or consume energy.

Let E denote total battery capacity, which is the maximum available energy for each sensor. Let $u_{i,k}^{\mathrm{dis}}$ denote the discharging rate of the sensor s_i when its sensing radius is r_k. According to [22], the relationship between the sensing radius and the discharging rate can be quantified by the following equation:

$$u_{i,k}^{\mathrm{dis}} = u_{i,m}^{\mathrm{dis}} \times (r_k/r_m)^2 \quad (2)$$

where $u_{i,m}^{\mathrm{dis}}$ is the discharging rate of s_i when it selects the maximum sensing radius r_m. The value of $u_{i,m}^{\mathrm{dis}}$, which is given in advance, is dependent on the basic features of the sensor.

Recall that the proposed algorithm partitions one day into several equal-length cycles, denoted by T_c, and each cycle consists of a fixed number of identical time slots. Then, the algorithm allocates the equal available energy for each cycle based on the estimated acquired energy. Given that the discharging rate of a sensor reaches its maximum when the sensor selects the maximum sensing radius r_m, it is assumed that the energy allocated for each cycle is only sufficient to

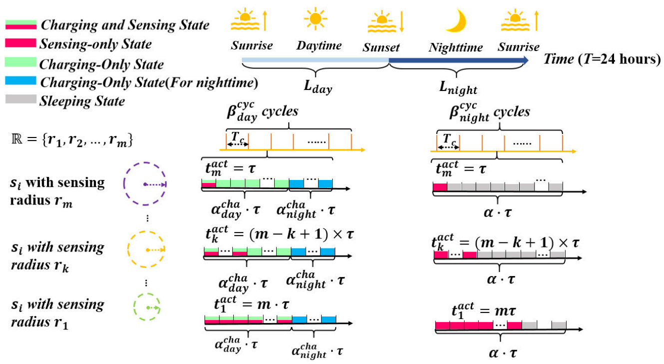


Fig. 3. Example of the daytime and nighttime cycle and the charging and discharging model.

sustain the sensor being activated at r_m for one time slot per cycle. This time slot, denoted by τ, serves as the fundamental time unit in each cycle for task scheduling. The lengths of one time slot and one cycle are further discussed in Section IV.

For ease of calculation, assume that the relation between any sensing radius r_k and the maximum sensing radius r_m is given by the following equation:

$$\frac{r_m}{r_k} = \sqrt{m-k+1}. \quad (3)$$

Let $t_{i,k}^{\text{act}}$ denote the active time of the sensor s_i when it selects the sensing radius r_k, where the active time refers to the maximum duration that the energy allocated for each cycle allows the sensor to perform sensing operations. According to (2) and (3), the value of $t_{i,k}^{\text{act}}$ can be calculated by applying the following equation:

$$t_{i,k}^{\text{act}} = (m-k+1) \cdot \tau. \quad (4)$$

Fig. 3 gives an example of daytime and nighttime cycle, which reflects the considered charging and discharging model. In this example, we have $\mathbb{R} = \{r_1, r_2, \ldots, r_m\}$. According to (3), we have

$$r_k = \frac{r_m}{\sqrt{m-k+1}}, \quad r_1 = \frac{r_m}{\sqrt{m}}.$$

As shown in Fig. 3, the sensor s_i can be activated in one time slot when it adopts the maximum sensing radius r_m. The active time of s_i selecting sensing radii r_k and r_1 is $(m-k+1)\cdot\tau$ and $m \cdot \tau$, respectively.

D. Channel Selection and Transmission Scheduling Protocol

In the considered WRSNs, the channel is assumed to be a quasi-static fading channel. This channel model presumes that the channel remains unchanged over a short period but undergoes changes over a longer duration. The primary reasons for assuming the channel to be a quasi-static fading channel are given as follows. First, although the sensors are solar-powered, the daily harvested energy is still limited. The quasi-static fading channel assumption reduces the need for frequent channel state updates, which helps conserve energy and prolongs the lifespan of the nodes. Second, under a quasi-static fading channel, the channel remains relatively stable over a short period, and sensor nodes do not require frequent channel estimation. This simplifies the algorithm implementation, reduces computational overhead, and is suitable for resource-constrained WRSNs. Finally, area coverage is mainly applied in long-term monitoring of large areas, such as environmental and industrial facility monitoring. These application scenarios typically have stable environments. In this study, the sensors are static with fixed deployment locations, resulting in slow channel condition changes, that is, the channel remains stable over a longer period and only undergoes significant changes overextended durations. The quasi-static fading channel model can better reflect the channel characteristics in actual environments, enhancing the practical application effectiveness of the algorithm.

Under this assumption, the carrier sense multiple access with collision avoidance (CSMA/CA) protocol is selected to optimize communication efficiency and reduce data collisions. This is primarily because, in a quasi-static fading channel, the channel remains stable over a short period, making it well suited for the collision avoidance mechanism of CSMA/CA. Moreover, the implementation of the CSMA/CA protocol is relatively simple, requiring sensor nodes to perform channel listening and basic random backoff calculations, which incurs low computational overhead. In addition, CSMA/CA reduces energy consumption by minimizing packet collisions and retransmissions through its collision avoidance mechanism. On the other hand, compared to time division multiple access (TDMA) and code division multiple access (CDMA), CSMA/CA does not require global clock synchronization or complex code allocation. The proposed JMLA is a distributed algorithm, and CSMA/CA can operate efficiently without a central control node, aligning well with the distributed algorithm application scenarios proposed in this article.

Let Boolean variables $c(t)$ represent whether the channel state is idle or busy at time t, that is,

$$c(t) = \begin{cases} 1, & \text{if the channel state is idle at time } t \\ 0, & \text{if the channel state is busy at time } t. \end{cases}$$

Each sensor s_i initial checks the current channel state $c(t)$ before attempting to transmit data. If the channel is idle ($c(t) = 1$), s_i proceeds to the transmission preparation phase. If the channel is busy ($c(t)= 0$), s_i enters the backoff phase. Let T_i^w denote the random backoff waiting time for the sensor s_i when $c(t)= 0$. In the backoff phase, the sensor s_i initially calculates the backoff time T_i^w using an exponential backoff algorithm controlled by the backoff stage k, that is,

$$T_i^w = \text{Random}(0, 2^k - 1) \times B_{\text{unit}}$$

where B_{unit} is the backoff time unit, which represents the fundamental unit of time that a network node must wait before attempting to retransmit data after detecting that the channel is busy.

After the backoff time ends, s_i rechecks the channel status. Upon confirming that the channel is idle, it sends a request to send (RTS) signal, requesting data transmission. When the receiver receives the RTS frame, it checks if the channel is currently idle and if it is ready to receive data. If both conditions are met, it sends a clear-to-send (CTS) frame back to s_i, indicating that it is CTS transmission. After receiving the CTS response, s_i begins data transmission operation. Once the transmission is completed, s_i receives an acknowledgment signal (ACK), confirming that the data have successfully reached the receiver.

E. Problem Statement

This article considers the area coverage problem in the WRSNs that adopt the adjustable sensing radii sensors. The proposed JMLA algorithm schedules sensors to cooperatively detect the monitoring region R in a distributed manner, aiming to maximize the monitoring quality while perpetuating the network lifetime.

Let $\mathbb{s}$ denote a possible scheduling algorithm to construct an area coverage. The following initially discusses the monitoring quality under the scheduling algorithm $\mathbb{s}$. It is noticed that sensors are activated sequentially, and sensors in the sensing state vary at different times. Therefore, the monitoring quality should be considered in both spatial and temporal dimensions. Let $S^{\text{act}}_{v,h}(\mathbb{s})$ denote the set of sensors that can cover the point $v \in R$ and are scheduled to be activated at the time t_h by applying algorithm $\mathbb{s}$. Consider a space–time point (v, t_h), where $v \in R$ and $t_h \in T_c$. Let $p^{\mathbb{s}}_{v,h}$ denote the cooperative detection probability of all sensors in $S^{\text{act}}_{v,h}(\mathbb{s})$ to the space–time point (v, t_h), that is,

$$p^{\mathbb{s}}_{v,h} = 1 - \prod_{s_i \in S^{\text{act}}_{v,h}(\mathbb{s})} (1 - p(s_i, v)) \tag{5}$$

where the term $\prod_{s_i \in S^{\text{act}}_{v,h}(\mathbb{s})}(1 - p(s_i, v))$ represents the probability that all sensors in $S^{\text{act}}_{v,h}(\mathbb{s})$ cooperatively monitor point v but fail to detect an occurred event.

Let $q^{\mathbb{s}}_v$ denote the monitoring quality to point v achieved by applying algorithm $\mathbb{s}$. To guarantee the monitoring quality, this article defines $q^{\mathbb{s}}_v$ as the weakest cooperative detection probability to point v across all the time slots in one cycle, that is,

$$q^{\mathbb{s}}_v = \min_{t_h \in T_c} p^{\mathbb{s}}_{v,h}.$$

Let $Q(\mathbb{s})$ denote the monitoring quality of the area coverage achieved by the algorithm $\mathbb{s}$. To ensure that the monitoring quality can be maximized to any point at any time, that is, to any space–time point, $Q(\mathbb{s})$ is represented by the weakest monitoring quality among all active sensors to any point, that is,

$$Q(\mathbb{s}) = \min_{v \in R} q^{\mathbb{s}}_v. \tag{6}$$

However, there are infinite points within monitoring region R. This article applies the grid-based approach to cope with this problem. The detail of this approach is presented in Section IV.

The proposed JMLA algorithm aims to maximize the monitoring quality of the constructed area barrier while balancing energy consumption and acquisition to perpetuate the network lifetime of deployed WRSNs. The object function of this article is given as follows:

Objective Function:

$$\text{Max}\left[Q(\mathbb{s}) = \min_{v \in R} q^{\mathbb{s}}_v\right]. \tag{7}$$

Equation (7) should be achieved under some constraints, which are presented in the following.

Let Boolean variables $M^{\text{cha\&sen}}_{i,h}$, $M^{\text{cha-only}}_{i,h}$, $M^{\text{sen-only}}_{i,h}$, and $M^{\text{slp}}_{i,h}$ represent whether sensor s_i stays in charging and sensing, charging-only, sensing-only, or sleeping states at t_h, respectively, that is,

$$M^{\text{cha\&sen}}_{i,h} = \begin{cases} 1, & \text{if } s_i \text{ stay in charging and sensing state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

$$M^{\text{cha-only}}_{i,h} = \begin{cases} 1, & \text{if } s_i \text{ stay in charging-only state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

$$M^{\text{sen-only}}_{i,h} = \begin{cases} 1, & \text{if } s_i \text{ stay in sensing-only state at } t_h \\ 0, & \text{otherwise} \end{cases}$$

$$M^{\text{slp}}_{i,h} = \begin{cases} 1, & \text{if } s_i \text{ stay in sleeping state at } t_h \\ 0, & \text{otherwise.} \end{cases}$$

Equation (8) gives the state constraint, which ensures that each sensor can only stay in one of these four states at any given time.

1) State Constraint:

$$M^{\text{cha\&sen}}_{i,h} + M^{\text{cha-only}}_{i,h} + M^{\text{sen-only}}_{i,h} + M^{\text{slp}}_{i,h} = 1. \tag{8}$$

The second constraint guarantees that each scheduled sensor should be activated at least one time slot within each cycle such that all sensors will participate in the monitoring task. Equation (9) reflects this constraint.

2) Working Constraint:

$$\sum_{h=1}^{\left\lceil \frac{T_c}{\tau} \right\rceil} M^{\text{cha\&sen}}_{i,h} + M^{\text{sen-only}}_{i,h} \geq 1 \quad \forall s_i. \tag{9}$$

Let $p^{\text{PV}}(t)$ denote the PV power function, which reflects the real-time solar power. The total amount of energy that can be obtained from solar power in one day is $\int_{t_s}^{t_e} p^{\text{PV}}(t)dt$, where t_s and t_e are the starting and ending times of daytime, respectively. Recall that the active time of the sensor when it selects the sensing radius r_k in one cycle is $t^{\text{sen}}_{i,k}$, and each sensor is activated in the same time slot within each cycle to maintain equal monitoring quality across cycles. Therefore, the total amount consumed energy of each sensor in one day, denoted by $E^{\text{con}}_{\text{tot}}$, can be calculated by applying the following equation:

$$E^{\text{con}}_{\text{tot}} = \left(u^{\text{dis}}_{i,k} \cdot t^{\text{sen}}_{i,k}\right) \cdot \left\lceil \frac{T}{T_c} \right\rceil \tag{10}$$

where T and T_c are the length of one day and one cycle, respectively. To guarantee the perpetual network lifetime, the acquired energy and the consumed energy of each sensor must be balanced each day. Equation (11) reflects this constraint.

3) Perpetual Lifetime Constraint:

$$\int_{t_s}^{t_e} p^{\mathrm{PV}}(t)dt = \left(u_{i,k}^{\mathrm{dis}} \cdot t_{i,k}^{\mathrm{sen}}\right) \cdot \left\lceil \frac{T}{T_c} \right\rceil. \tag{11}$$

Nomenclature summarizes the parameters used in this article. Then, Section IV presents the proposed JMLA algorithm, which aims to achieve the objective function shown in (7) while satisfying constraints given in (8), (9), and (11).

IV. Proposed JMLA Algorithm

This article proposed a novel area coverage algorithm, called JMLA for area coverage. The JMLA algorithm aims to efficiently schedule each sensor to perform sensing, charging, and sleeping operations in different time slots according to its sensing radius to cooperatively monitor the given region, which helps maximize the monitoring quality while perpetuating the network lifetime.

The JMLA algorithm mainly consists of three phases, including the space–time partitioning phase, the monitoring contribution computation phase, and the scheduling phase.

Initially, the monitoring region R is divided into several grids with equal size. Moreover, considering the variability of solar power and its susceptibility to weather conditions, the JMLA utilizes calculations based on similarity to pinpoint the day from the previous year that best matches the expected meteorological conditions of the forthcoming day. Then, the PV power function of that day is applied to predict the actual PV power for the forthcoming day, thus estimating potential solar energy availability. Following this, the JMLA segments the continuous time into multiple time slots of uniform length. Each cycle consists of a fixed number of time slots. The duration of each slot and cycle is determined by the predicted PV power for the next day and the discharging rate of sensors to ensure a balance between energy consumption and supply. These organized time slots and uniformly sized grids form a structured series of space–time points, enabling straightforward assessment of monitoring quality for each grid during each time interval. The second phase involves calculating the detection probability of each sensor for each grid, representing the monitoring contribution of each sensor.

In the last phase, the algorithm identifies the bottleneck space–time point that has the weakest cooperative detection probability. Then, it designs the task schedule in a distributed manner, prioritizing the sensors with the maximum contribution to the bottleneck space–time point, aiming to maximize monitoring quality for area coverage.

A. Space–Time Partitioning Phase

The primary objective of the proposed algorithm is to initially identify the bottleneck space–time point, consisting of a specific time point (when) within a cycle and a location (where) in monitoring region R, where the cooperative detection probability is the weakest. Subsequently, the JMLA schedules the sensor with the maximum monitoring contribution to enhance the cooperative detection probability of bottleneck space–time point, aiming to maximize the monitoring quality of area coverage. However, both the time points and locations in R are infinite. To simplify the investigated

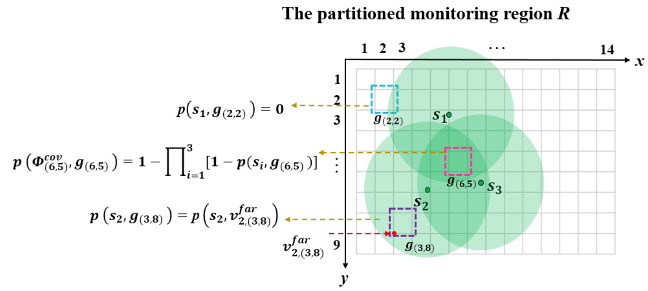


Fig. 4. Scenario of partitioned region R monitored by three active sensors.

issue and reduce the computation complexity, this phase first partitions the R into several identical grids by applying a grid-based approach. The detection probability of each sensor to any point in one grid is considered to be identical. A grid with the coordinates (x, y) represented by $g_{(x,y)}$. The top-left grid is labeled $g_{(1,1)}$. When the grid is shifted one position to the right, the value of x increases by one, and when it is shifted down, the value of y increases by one. Fig. 4 shows an example of a monitoring region R, partitioned into 14×9 identical grids, monitored by three active sensors s_1–s_3.

Based on the same considerations for space partitioning, the time axis then will be partitioned into several equal-length cycles and each cycle consists of several equal-length time slots. Sensors will be scheduled to operate in different states across various time slots within one cycle, and their task schedules will be consistent across all cycles, ensuring uniform monitoring quality at all times. The length of each time slot and cycle is determined by evaluating the potential for energy acquisition of the next day and the discharging rate of each sensor with the corresponding sensing radius, where the potential for energy acquisition can be derived from the PV power of the next day. However, the PV power of the next day is unknown, which needs to be forecast.

The operation of this phase consists of two steps. The first step is to employ the similarity-based calculation to identify the day in the previous year that most closely resembles the meteorological scenario of the next day. It then uses the PV power function from that day to estimate the PV power for the next day. The second step is to calculate the length of one time slot and one cycle in both daytime and nighttime, based on the assessment of the PV power and the discharging rate of each sensor, aiming to balance the acquired and consumed energy.

Step 1 (PV Power Evaluation for the Next Day): The total amount of solar energy in a certain period is the integral of the PV power function [23]. Therefore, the future PV power function should be forecast in advance. Most existing studies adopt deep learning algorithms to forecast PV power, which requires a high-performance computing device to train the forecasting model. However, since most devices in the WRSNs have limited power and computing capabilities, it is impractical to forecast future PV power using deep learning algorithms.

To forecast the next day's PV power within an acceptable error margin, while conserving computational resources, the

JMLA algorithm first stores the previous year's daily PV power and related meteorological data at the base station, followed by the acquisition of the next day's meteorological data forecasts through the network. The main weather factors that influence PV power generation include horizontal radiation, land surface temperature, surface wind speed, relative humidity, and total cloud cover [24]. JMLA then creates two matrices to systematically record historical and future meteorological data. This structured approach provides an organized framework that ensures effective storage and access to meteorological data across various time intervals. Assume that meteorological data are recorded n times per day, with equal time intervals between each recording point. The ith recording time point is represented by $\mathcal{f}_i$. Let m_1–m_5 represent the identifiers for horizontal radiation, land surface temperature, surface wind speed, relative humidity, and total cloud cover, respectively. Let $\mathbb{M}_{\text{met}}^{i\text{-his}}$ denote the historical daily meteorological data matrix that records the meteorological data for ith day of the previous year, which is shown as follows:

$$\mathbb{M}_{\text{met}}^{i\text{-hist}} = \begin{bmatrix} d_{m_1,\mathcal{f}_1}^{i\text{-hist}} & \cdots & d_{m_1,\mathcal{f}_n}^{i\text{-hist}} \\ \vdots & \ddots & \vdots \\ d_{m_5,\mathcal{f}_1}^{i\text{-hist}} & \cdots & d_{m_5,\mathcal{f}_n}^{i\text{-hist}} \end{bmatrix} \tag{12}$$

where $d_{m_p,\mathcal{f}_q}^{i\text{-hist}}$ represents the corresponding meteorological data at the qth recording time point on the ith day of the previous year.

Because there are 365 days in one year, let $M_{\text{met}}^{\text{hist}}$ denote the set of historical daily meteorological data matrices, which is defined as

$$M_{\text{met}}^{\text{hist}} = \left\{\mathbb{M}_{\text{met}}^{1\text{-hist}}, \mathbb{M}_{\text{met}}^{2\text{-hist}}, \ldots, \mathbb{M}_{\text{met}}^{365\text{-hist}}\right\}.$$

Let $\mathbb{M}_{\text{met}}^{j\text{-fore}}$ denote the forecast meteorological data matrix that records the meteorological data for the next day, which is the jth day of this year, that is,

$$\mathbb{M}_{\text{met}}^{i\text{-fore}} = \begin{bmatrix} d_{m_1,\mathcal{f}_1}^{j\text{-fore}} & \cdots & d_{m_1,\mathcal{f}_n}^{j\text{-fore}} \\ \vdots & \ddots & \vdots \\ d_{m_5,\mathcal{f}_1}^{j\text{-fore}} & \cdots & d_{m_5,\mathcal{f}_n}^{j\text{-fore}} \end{bmatrix} \tag{13}$$

where $d_{m_p,\mathcal{f}_q}^{j\text{-fore}}$ represents the corresponding meteorological data at the qth time point in the next day.

Subsequently, the JLMA algorithm aims to identify the day in the previous year that most closely matches the meteorological conditions expected for the next day. For convenience, this day is referred to as the "template day" for the next day. It then selects the PV power function from this template day to serve as the template PV power function for the next day. To achieve this, JMLA first calculates the difference between $\mathbb{M}_{\text{met}}^{j\text{-fore}}$ and each matrix $\mathbb{M}_{\text{met}}^{i\text{-hist}} \in M_{\text{met}}^{\text{hist}}$ by applying the Frobenius norm, or F-norm in short [25]. Let $\mathbb{M}_{i,j}^{\text{met-hist}}$ denote the difference matrix between $\mathbb{M}_{\text{met}}^{j\text{-fore}}$ and $\mathbb{M}_{\text{met}}^{i\text{-hist}}$, that is,

$$\begin{aligned} \mathbb{M}_{i,j}^{\text{met-hist}} &= \mathbb{M}_{\text{met}}^{j\text{-fore}} - \mathbb{M}_{\text{met}}^{i\text{-hist}} \\ &= \begin{bmatrix} d_{m_1,\mathcal{f}_1}^{j\text{-fore}} - d_{m_1,\mathcal{f}_1}^{i\text{-hist}} & \cdots & d_{m_1,\mathcal{f}_n}^{j\text{-fore}} - d_{m_1,\mathcal{f}_n}^{i\text{-hist}} \\ \vdots & \ddots & \vdots \\ d_{m_5\mathcal{f}_1}^{j\text{-fore}} - d_{m_5\mathcal{f}_1}^{i\text{-hist}} & \cdots & d_{m_5\mathcal{f}_n}^{j\text{-fore}} - d_{m_5\mathcal{f}_n}^{i\text{-hist.}} \end{bmatrix}. \end{aligned} \tag{14}$$

The F-norm of $\mathbb{M}_{i,j}^{\text{met-hist}}$ can be calculated by applying the following equation:

$$\left\|\mathbb{M}_{i,j}^{\text{met-hist}}\right\|_F = \sqrt{\sum_{p=1}^{5}\sum_{q=1}^{n}\left|d_{m_p,\mathcal{f}_q}^{j\text{-fore}} - d_{m_p,\mathcal{f}_q}^{i\text{-hist}}\right|^2}. \tag{15}$$

Let $\mathbb{M}_{\text{met}}^{\text{tem-hist}}$ denote the matrix that records the meteorological data from the template day, that is,

$$\mathbb{M}_{\text{met}}^{\text{tem-hist}} = \arg\min_{\mathbb{M}_{\text{met}}^{1\text{-hist}} \in M_{\text{met}}^{\text{hist}}} \left\|\mathbb{M}_{i,j}^{\text{met-hist}}\right\|_F. \tag{16}$$

The PV power function of the template day, provided in advance, is used to approximate the actual PV power function of the next day. This function serves to estimate the potential for solar energy acquisition on the next day.

Step 2 (Calculating the Length of One Time Slot and Cycle): Based on the template PV power function for the next day, this step involves calculating the lengths of each time slot and cycle to maintain a balance between the acquired energy and consumed energy of each sensor during both daytime and nighttime periods.

It is noticed that the solar power is available only during the daytime. Recall that t_s and t_e are the starting and ending times of daytime, respectively. Let $p_{j\text{-tem}}^{\text{PV}}(t)$ denote the template PV power function of the next day, which is the jth day of this year. Let $E_{\text{tot}}^{\text{acq-est}}$ denote the estimated total amount of energy that can be acquired from solar power in the next day, and the value of $E_{\text{tot}}^{\text{acq-est}}$ is

$$E_{\text{tot}}^{\text{acq-est}} = \int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt. \tag{17}$$

By considering the sensing operation of each sensor during the nighttime, the JMLA algorithm reserves enough energy for each sensor in advance to support each sensor performing sensing operation during nighttime. Let E_j^d and E_j^n denote the energy allocated for each sensor during daytime and nighttime, respectively. To achieve the balance between the obtained energy and consumed energy throughout the day, the following equation should be satisfied:

$$E_j^d + E_j^n = \int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt. \tag{18}$$

Since each sensor can be continuously charged from solar power during the daytime, the balance between the obtained energy and consumed energy through the daytime can be achieved if this balance is maintained in each cycle. Therefore, the energy allocated for each sensor consumed in each cycle is E_j^d. Recall the charging and discharging model, and the energy allocated for each cycle is only sufficient to sustain the sensor being activated with the maximum sensing radius r_m for one time slot per cycle. Therefore, the length of each cycle is

$$\tau = \frac{E_j^d}{u_{i,m}^{\text{dis}}} \tag{19}$$

where $u_{i,m}^{\text{dis}}$ is the discharging rate of the sensor s_i when its sensing radius is r_m.

The following further discusses the length of one cycle. As shown in Fig. 3, there are $\alpha_{\text{day}}^{\text{cha}}$ and $\alpha_{\text{night}}^{\text{cha}}$ time slots that are allocated to charge the energy for each sensor consuming during daytime and nighttime, respectively. The length of each cycle can be calculated once the values of $\alpha_{\text{day}}^{\text{cha}}$ and $\alpha_{\text{night}}^{\text{cha}}$ are determined. To achieve the balance between the obtained energy and consumed energy in each cycle, the following equation should be satisfied:

$$\frac{\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt}{(t_e - t_s)} \times \tau \times \alpha_{\text{day}}^{\text{cha}} = u_{i,m}^{\text{dis}} \times \tau \tag{20}$$

where $((\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt)/(t_e - t_s))$ is the average PV power during the daytime. Based on (20), the value of $\alpha_{\text{day}}^{\text{cha}}$ can be calculated, that is,

$$\alpha_{\text{day}}^{\text{cha}} = \frac{u_{i,m}^{\text{dis}}(t_e - t_s)}{\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt}. \tag{21}$$

The following further calculated the value of $\alpha_{\text{night}}^{\text{cha}}$. As shown in Fig. 3, there are $\beta_{\text{day}}^{\text{cyc}}$ cycles during the nighttime. Therefore, the total amount of energy charged from solar power to support each sensor performing sensing operations during nighttime is given as follows:

$$E_j^n = \beta_{\text{day}}^{\text{cyc}} \times \alpha_{\text{night}}^{\text{cha}} \times \tau \times \frac{\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt}{(t_e - t_s)} \tag{22}$$

where the value of $\beta_{\text{day}}^{\text{cyc}}$ is

$$\beta_{\text{day}}^{\text{cyc}} = \frac{t_e - t_s}{\left(\alpha_{\text{day}}^{\text{cha}} + \alpha_{\text{night}}^{\text{cha}}\right) \times \tau}. \tag{23}$$

As shown in Fig. 3, there are $\beta_{\text{night}}^{\text{cyc}}$ cycles during the nighttime, that is,

$$\beta_{\text{night}}^{\text{cyc}} = \frac{T - t_e + t_s}{\left(\left(\alpha_{\text{day}}^{\text{cha}} + \alpha_{\text{night}}^{\text{cha}}\right) \times \tau\right)} \tag{24}$$

where T is the length of one day.

Since the task schedules for each sensor are identical across different cycles, each sensor consumes the same amount of energy in every cycle. Based on the number of cycles during the nighttime, the total amount of consumed energy of each sensor during the nighttime is $\beta_{\text{night}}^{\text{cyc}} \times \tau \times u_{i,m}^{\text{dis}}$. To balance the charged energy to support each sensor performing sensing operations during nighttime and the consumed energy during the nighttime, the following equation should be satisfied:

$$\beta_{\text{day}}^{\text{cyc}} \times \alpha_{\text{night}}^{\text{cha}} \times \tau \times \frac{\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt}{(t_e - t_s)} = \beta_{\text{night}}^{\text{cyc}} \times \tau \times u_{i,m}^{\text{dis}}. \tag{25}$$

According to (25), the value of $\alpha_{\text{night}}^{\text{cha}}$ is

$$\alpha_{\text{night}}^{\text{cha}} = \frac{u_{i,m}^{\text{dis}}(T - t_e + t_s)}{\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt}. \tag{26}$$

Consequently, the length of one cycle T_c is

$$T_c = \frac{\left(T \times u_{i,m}^{\text{dis}}\right)}{\int_{t_s}^{t_e} p_{j\text{-tem}}^{\text{PV}}(t)dt} \times \tau. \tag{27}$$

Until now, the continuous time can be partitioned into several equal-length time slots, with the length of one time slot provided in (19). Each cycle consists of $(T \times u_{i,m}^{\text{dis}})$/time slots consists.

In the next phase, the detection probability of each sensor to each covered grid is calculated, representing the monitoring contribution of each sensor.

B. Monitoring Contribution Computation Phase

The detection probability of each sensor to each grid is calculated in this phase. Assume that grid $g_{(x,y)}$ is covered by the sensor s_i, and the detection probability of s_i $g_{(x,y)}$ depends on whether or not $g_{(x,y)}$ is completely covered by s_i and the positions of $g_{(x,y)}$ and s_i. In case that $g_{(x,y)}$ is partially covered by $g_{(x,y)}$, events occurring within the uncovered part in $g_{(x,y)}$ will not be detected by s_i. Therefore, the detection probability of s_i to $g_{(x,y)}$ is 0 to ensure the monitoring quality.

In case that $g_{(x,y)}$ is completely covered by s_i, the detection probability of s_i to $g_{(x,y)}$ depends on the distance between s_i and $g_{(x,y)}$. Let $v_{i,(x,y)}^{\text{far}}$ denote the farthest point in $g_{(x,y)}$ to s_i, that is,

$$v_{i,(x,y)}^{\text{far}} = \arg \max_{v \in g_{(x,y)}} d(v, s_i). \tag{28}$$

To ensure the monitoring quality, the point $v_{i,(x,y)}^{\text{far}}$ is selected as the representative point of the grid $g_{(x,y)}$. Therefore, the detection probability of s_i to $g_{(x,y)}$ can be calculated by applying the following equation:

$$p\left(s_i, g_{(x,y)}\right) = p\left(s_i, v_{i,(x,y)}^{\text{far}}\right). \tag{29}$$

The following further discusses the case that two or more active sensors can completely cover the grid $g_{(x,y)}$. Let $\Phi_{(x,y)}^{\text{cov}} = \{s_{(x,y),1}^{\text{cov}}, s_{(x,y),2}^{\text{cov}}, \ldots, s_{(x,y),n}^{\text{cov}}\}$ denote the set of active sensors that fully covers the grid $g_{(x,y)}$. Let $p(\Phi_{(x,y)}^{\text{cov}}, g_{(x,y)})$ denote the cooperative detection probability of all sensors in $\Phi_{x,y}^{\text{act}}$ to $g_{(x,y)}$. The value of $p(\Phi_{(x,y)}^{\text{cov}}, g_{(x,y)})$ can be calculated by applying the following equation:

$$p\left(\Phi_{(x,y)}^{\text{cov}}, g_{(x,y)}\right) = 1 - \prod_{i=1}^{n}\left[1 - p\left(s_{(x,y),i}^{\text{cov}}, g_{(x,y)}\right)\right]. \tag{30}$$

As shown in Fig. 4, $g_{(2,2)}$ is only partially covered by s_1, and we have $p(s_1, g_{(2,2)}) = 0$. The grid $g_{(3,8)}$ is completely covered by the sensor s_2, and the farthest point in $g_{(3,8)}$ to s_2 is marked by the red point, as shown in Fig. 4. The detection probability of s_2 to $g_{(3,8)}$ depends on the distance between s_2

and $v^{\text{far}}_{2,(3,8)}$. By applying (1), the value of $p(s_2, g_{(3,8)})$ is

$$\begin{aligned} &p\left(s_2, g_{(3,8)}\right)\\ &= p\left(s_2, v^{\text{far}}_{2,(3,8)}\right)\\ &= \begin{cases} 100\%, & d\left(s_2, v^{\text{far}}_{2,(3,8)}\right) \le r_g \\ e^{-\lambda\left(d\left(s_2, v^{\text{far}}_{2,(3,8)}\right)-r_g\right)^{\gamma}}, & r_g < d\left(s_2, v^{\text{far}}_{2,(3,8)}\right) \le r_m. \end{cases} \end{aligned}$$

Furthermore, the grid $g_{(6,5)}$ is completely covered by s_1–s_3, that is, $\Phi^{\text{cov}}_{(6,5)} = \{s_1, s_2, s_3\}$. The cooperative detection probability of all the sensors in $\Phi^{\text{cov}}_{(6,5)}$ to $g_{(6,5)}$ is

$$p\left(\Phi^{\text{cov}}_{(6,5)}, g_{(6,5)}\right) = 1 - \prod_{i=1}^{3}\left[1 - p\left(s_i, g_{(6,5)}\right)\right].$$

In this phase, the detection probability of each sensor to each grid is calculated, which is considered as the monitoring contribution of each sensor to each grid. If the detection of a sensor to a grid is 0, it is considered to have no monitoring contribution to this grid.

The next phase aims to first identify the bottleneck space–time point that has the weakest cooperative detection probability and then schedule the sensor with the largest monitoring contribution to the bottleneck space–time point, aiming to maximize the monitoring quality.

C. Scheduling Phase

The main task of this phase is to schedule a set of active sensors to be activated in different time slots to monitor the given area R, aiming to maximize the monitoring quality of constructed area coverage while perpetuating the network lifetime. To achieve this, the scheduling phase consists of three steps. The first step is to initially identify the bottleneck space–time point that has the weakest cooperative detection probability based on the current schedule of all sensors. In the second step, the sensor with the maximum monitoring contribution is scheduled to participate in monitoring the bottleneck space–time point in a distributed manner while satisfying the constraints given in (8), (9), and (11). In the last step, each sensor cooperates with its neighbors, aiming to achieve the maximal monitoring quality.

The operations mentioned above are performed repeatedly until all sensors are scheduled, and the area coverage is achieved.

Step 1 (Finding the "bottleneck space–time point"): In the space–time partitioning phase, the continuous time has been partitioned into several identical time slots and a fixed number of time slots consist of each cycle. These time slots and partitioned grids in monitoring region R can form multiple space–time points. Each space–time point is a 2-D data structure and can be represented by $(g_{(x,y)}, t_h)$, which corresponds to the grid $g_{(x,y)}$ at time slot t_h. Based on this scheme, the cooperative detection probability of all active sensors to $g_{(x,y)}$ at time slot t_h can be easily represented by the cooperative detection probability to $(g_{(x,y)}, t_h)$. This allows for the evaluation of monitoring quality in both spatial and temporal dimensions.

This step is to initially calculate the cooperative detection probability of current active sensors to each space–time point and then identify the bottleneck space–time point that has the weakest cooperative detection probability, which is the point that needs to be monitored by more sensors. Because the proposed JMLA algorithm schedules all the sensors in a distributed manner, each sensor is responsible for its local monitoring region (LMR), which consists of the grids located within its sensing radius. Let LMR_i denote the LMR of s_i, that is,

$$\text{LMR}_i = \{g_{(x,y)} | d\left(s_i, g_{(x,y)}\right) \le r_m\} \quad \forall g_{(x,y)} \in R \tag{31}$$

where r_m is the maximum sensing radius of each sensor.

Considering the space–time points, let LMR^{st}_i denote the local monitoring space–time region of s_i, which expands the LMR^{st}_i into two dimensions. This means that each sensor s_i only focuses on the monitoring quality of its LMR in the spatial dimension and the time slots within one cycle in the temporal dimension. LMR^{st}_i is given as follows:

$$\text{LMR}^{\text{st}}_i = \{(g_{(x,y)}, t_h) | g_{(x,y)} \in \text{LMR}_i, t_h \in T_c\} \tag{32}$$

where T_c is one cycle.

Let $p_{(x,y),h}$ denote the cooperative detection probability of all the active sensors to the space–time point $(g_{(x,y)}, t_h)$, which reflects the cooperative detection probability contributed by the set of active sensors that can fully cover the grid $g_{(x,y)}$ at time slot t_h. The value of $p_{(x,y),h}$ can be calculated by applying the following equation:

$$p_{(x,y),h} = 1 - \prod_{s_j \in S^{\text{sch}}_{(x,y),h}} [1 - p(s_j, g_{(x,y)})] \tag{33}$$

where $S^{\text{sch}}_{(x,y),h}$ is the set of sensors that can fully cover $g_{(x,y)}$ and are scheduled to be active at the time slot t_h.

Let $o^{\text{bott}}_{i,\text{loc}}$ denote the local bottleneck space–time points in LMR^{st}_i, that is,

$$o^{\text{bott}}_{i,\text{loc}} = \arg \min_{(g_{(x,y)}, t_h) \in \text{LMR}^{\text{st}}_i} p_{(x,y),h}. \tag{34}$$

To obtain the details of $S^{\text{sch}}_{(x,y),h}$, the matrices $M^{\text{sch}}_{i,\text{loc}}$ and $M^{\text{QoM}}_{i,\text{loc}}$ are adopted to store the local current sensor scheduling results and local monitoring quality in LMR^{st}_i, respectively. Both of these two matrices are maintained by s_i. $M^{\text{sch}}_{i,\text{loc}}$ is applied to store the current task schedule of all the neighboring sensors of s_i. A sensor s_j is considered the neighboring sensors of s_i if the Euclidean distance between s_i and s_j is less than their communication radius, which is twice the maximum sensing radius. Let $\mathbb{N}(s_i)$ denote the set of neighboring sensors of s_i, and we have

$$\mathbb{N}(s_i) = \left\{s_j \mid d\left(s_i, s_j\right) \le 2r_m\right\}.$$

Assume that each cycle has n time slots, and the structure of $M^{\text{sch}}_{i,\text{loc}}$ is

$$M^{\text{sch}}_{i,\text{loc}} = \begin{pmatrix} S^{\text{sch}}_{(x_1,y_1),1} & \cdots & S^{\text{sch}}_{(x_1,y_1),n} \\ \vdots & \ddots & \vdots \\ S^{\text{sch}}_{(x_m,y_m),1} & \cdots & S^{\text{sch}}_{(x_m,y_m),n} \end{pmatrix}. \tag{35}$$

The matrix $M^{\text{QoM}}_{i,\text{loc}}$ is applied to store the cooperative detection probability of all the space–time points in LMR^{st}_i. The

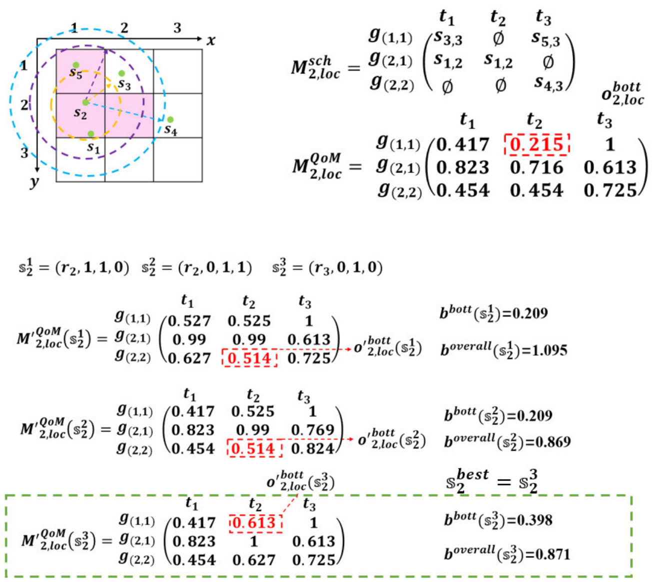


Fig. 5. Example of finding the bottleneck space–time point and sensor scheduling.

structure of $M_{i,\text{loc}}^{\text{QoM}}$ is

$$M_{i,\text{loc}}^{\text{QoM}} = \begin{pmatrix} p_{(x_1,y_1),1} & \cdots & p_{(x_1,y_1),n} \\ \vdots & \ddots & \vdots \\ p_{(x_m,y_m),1} & \cdots & p_{(x_m,y_m),n} \end{pmatrix}. \tag{36}$$

Each sensor will broadcast its scheduling result to all its neighboring sensors, and then, $M_{i,\text{loc}}^{\text{sch}}$ can be initially obtained. Each element $p_{(x_v,y_v),h}$ in $M_{i,\text{loc}}^{\text{QoM}}$ can be calculated according to (33).

Based on $M_{i,\text{loc}}^{\text{QoM}}$, the local bottleneck space–time point $o_{i,\text{loc}}^{\text{bott}}$ can be found by applying (34). In case multiple space–time points satisfy (34), let $O_{i,\text{loc}}^{\text{bott}}$ denote the set of local space–time points in LMR_i^{st}. The next step aims to schedule the sensor with the maximum monitoring contribution to enhance the monitoring quality of LMR_i^{st}.

Fig. 5 depicts an example to illustrate the operation of finding the bottleneck space–time point. As shown in Fig. 5, a set of sensors $S = \{s_1, s_2, \ldots, s_5\}$ are randomly deployed in the partitioned monitoring region. For an unscheduled sensor s_2, the set of available sensing radii is $\mathbb{R} = \{r_1, r_2, r_3\}$. The local monitoring space–time region of s_2 is

$$\text{LMR}_2^{\text{st}} = \left\{\left(g_{(x,y)}, t_h\right) \mid d\left(s_2, g_{(x,y)}\right) \le r_3, t_h \in T_c\right\}.$$

The set of neighboring sensors of s_2 is $\mathbb{N}(s_2) = \{s_1, s_2, s_3, s_4, s_5\}$. Assume that each cycle T_c has three time slots, and the local current sensor scheduling results in LMR_2^{st} is

$$M_{2,\text{loc}}^{\text{sch}} = \begin{pmatrix} s_{3,3} & \emptyset & s_{5,3} \\ s_{1,2} & s_{1,2} & \emptyset \\ \emptyset & \emptyset & s_{4,3} \end{pmatrix}$$

where each element $s_{i,k}$ in $M_{2,\text{loc}}^{\text{sch}}$ represents the sensor s_i with sensing radius r_k.

Then, the matrix $M_{2,\text{loc}}^{\text{QoM}}$ can be calculated based on the matrix $M_{2,\text{loc}}^{\text{sch}}$, which is given as follows:

$$M_{2,\text{loc}}^{\text{QoM}} = \begin{pmatrix} 0.417 & 0.215 & 1 \\ 0.823 & 0.716 & 0.613 \\ 0.454 & 0.454 & 0.725 \end{pmatrix}.$$

$o_{2,\text{loc}}^{\text{bott}}$ can be found by applying (34), which is $(g_{(1,1)}, t_2)$ and is marked in red ink in $M_{2,\text{loc}}^{\text{QoM}}$.

Step 2 (Scheduling Sensor With the Maximum Contribution to Monitor the Bottleneck Space–Time Point): Since the detection probability of each sensor to each grid is considered as the monitoring contribution of each sensor, sensor s_i is considered to have a monitoring contribution to the space–time point $(g_{(x,y)}, t_h)$ if sensor s_i can completely cover the grid $g_{(x,y)}$ that is activated in time slot t_h. The main task of this step is to schedule the sensor with the maximum monitoring contribution to the local space–time point, which is referred to as the best task schedule, in a distributed manner, aiming to maximize the monitoring quality.

The task schedule of each sensor includes its sensing radius and corresponding active time in each cycle. Let $\mathbb{s}_i^k$ denote the kth possible task schedule for the sensor s_i, which can be expressed as follows:

$$\mathbb{s}_i^k = \left(r_k, t_1^k, t_2^k, \ldots, t_n^k\right)$$

where each t_h^k in $\mathbb{s}_i^k$ is a Boolean variable that represents whether s_i is activated at the time slot t_h, that is,

$$t_h^k = \begin{cases} 1, & \text{if } s_i \text{is activated at } t_h \text{ under task schedule } \mathbb{s}_i^k \\ 0, & \text{otherwise.} \end{cases}$$

Recall that the active time of each sensor in one cycle should satisfy (4), and therefore, any feasible task schedule for the sensor s_i should satisfy the following constraint:

$$\sum_{h=1}^{n} t_h^k = m - k + 1.$$

Let $\mathbb{S}_i^{\text{bott}}$ denote the set of feasible task schedules for s_i that has the monitoring contribution to the local bottleneck space–time points in $O_{i,\text{loc}}^{\text{bott}}$. When sensor s_i adopts the feasible task schedule $\mathbb{s}_i^k \in \mathbb{S}_i^{\text{bott}}$, the local monitoring quality matrix $M_{i,\text{loc}}^{\text{QoM}}$ should be updated accordingly. Let $M'_{i,\text{loc}}(\mathbb{s}_i^k)$ denote the updated local monitoring quality matrix when s_i adopts the task schedule $\mathbb{s}_i^k$, which is given as follows:

$$M'^{\text{QoM}}_{i,\text{loc}}\left(\mathbb{s}_i^k\right) = \begin{pmatrix} p'_{(x_1 y_1)1}\left(\mathbb{s}_i^k\right) & \cdots & p'_{(x_1 y_1)n}\left(\mathbb{s}_i^k\right) \\ \vdots & \ddots & \vdots \\ p'_{(x_m y_m)1}\left(\mathbb{s}_i^k\right) & \cdots & p'_{(x_m y_m)n}\left(\mathbb{s}_i^k\right) \end{pmatrix} \tag{37}$$

where each element $p'_{(x_v,y_v),h}(\mathbb{s}_i^k)$ in $M'^{\text{QoM}}_{i,\text{loc}}(\mathbb{s}_i^k)$ represents the updated cooperative detection probability to space–time point $(g_{(x_v,y_v)}, t_h)$ after s_i adopting the task schedule $\mathbb{s}_i^k$. The value of $p'_{(x_v,y_v),h}(\mathbb{s}_i^k)$ can be calculated by applying the following equation:

$$p'_{(x_v,y_v),h}\left(\mathbb{s}_i^k\right) = 1 - \left(1 - p_{(x_v,y_v),h}\right)\left(1 - p\left(s_i, g_{(x_v,y_v)}\right) \times t_h^k\right).$$

Based on $M'^{\mathrm{QoM}}_{i,\mathrm{loc}}(\mathbb{s}_i^k)$, the new local bottleneck space–time point in $\mathrm{LMR}_i^{\mathrm{st}}$, denoted by $o'^{\mathrm{bott}}_{i,\mathrm{loc}}(\mathbb{s}_i^k)$, can be obtained according to the following equation:

$$o'^{\mathrm{bott}}_{i,\mathrm{loc}}\left(\mathbb{s}_i^k\right) = \arg \min_{(g_{(x_v,y_v)},t_h)\in \mathrm{LMR}_i^{\mathrm{st}}} p'_{(x_v,y_v),h}\left(\mathbb{s}_i^k\right). \tag{38}$$

The primary purpose of this step is to enhance the cooperative detection probability of local bottleneck space–time points by scheduling the sensor with the maximum monitoring contribution, aiming to maximize the monitoring quality. Let $b^{\mathrm{bott}}(\mathbb{s}_i^k)$ denote the benefit of s_i monitoring the local space–time point by adopting the feasible task schedule $\mathbb{s}_i^k \in \mathbb{S}_i^{\mathrm{bott}}$, and the value of $b^{\mathrm{bott}}(\mathbb{s}_i^k)$ can be calculated as follows:

$$b^{\mathrm{bott}}\left(\mathbb{s}_i^k\right) = p'^{\mathrm{bott}}_{(x_v,y_v),h}\left(\mathbb{s}_i^k\right) - p_{(x_v,y_v),}h^{\mathrm{bott}} \tag{39}$$

where $p'^{\mathrm{bott}}_{(x_v,y_v),h}(\mathbb{s}_i^k)$ and $p^{\mathrm{bott}}_{(x_v,y_v),h}$ are the cooperative detection probability of $o'^{\mathrm{bott}}_{i,\mathrm{loc}}(\mathbb{s}_i^k)$ and $o^{\mathrm{bott}}_{i,\mathrm{loc}}$, respectively. The value of $b^{\mathrm{bott}}(\mathbb{s}_i^k)$ reflects the improved local monitoring quality when s_i participates in monitoring by adopting the task schedule $\mathbb{s}_i^k$.

Let $\mathbb{s}_i^{\mathrm{best}}$ denote the best task schedule for s_i, which aims to maximize the improved local monitoring quality. $\mathbb{s}_i^{\mathrm{best}}$ can be derived by the following equation:

$$\mathbb{s}_i^{\mathrm{best}} = \arg \max_{\mathbb{s}_i^k\in \mathbb{s}_i^{\mathrm{bott}}} \left[b^{\mathrm{bott}}\left(\mathbb{s}_i^k\right)\right]. \tag{40}$$

However, there may exist more than one task schedule for s_i that satisfies (40). To finally select the best task schedule that can maximize the monitoring quality of the constructed area coverage, the JMLA algorithm further investigates the other advantages of the task schedules that satisfy (40). Let $\mathbb{S}_i^{\mathrm{cand}}$ denote the set of candidate best task schedules for s_i. To fully utilize the detection capability of each sensor, the proposed algorithm aims not only to enhance the cooperative detection probability of the bottleneck space–time point but also to improve the cooperative detection probability of all space–time points. Let $b^{\mathrm{overall}}(\mathbb{s}_i^k)$ denote the overall benefit of s_i monitoring the local space–time point by adopting the feasible task schedule $\mathbb{s}_i^k$, which is the total amount of improved cooperative detection probability of all space–time points in $\mathrm{LMR}_i^{\mathrm{st}}$. To calculate the value of $b^{\mathrm{overall}}(\mathbb{s}_i^k)$, a matrix, which is denoted by $M^{\mathrm{QoM\text{-}Im}}_{i,\mathrm{loc}}(\mathbb{s}_i^k)$, is initially established to store the improved cooperative detection probability of all space–time points in $\mathrm{LMR}_i^{\mathrm{st}}$. $M^{\mathrm{QoM\text{-}Im}}_{i,\mathrm{loc}}(\mathbb{s}_i^k)$ is given as follows:

$$\begin{aligned} &M^{\mathrm{QoM\text{-}Im}}_{i,\mathrm{loc}}\left(\mathbb{s}_i^k\right)\\ &= M'^{\mathrm{QoM}}_{i,\mathrm{loc}}\left(\mathbb{s}_i^k\right) - M^{\mathrm{QoM}}_{i,\mathrm{loc}}\\ &\cdot\begin{pmatrix} p'_{(x_1,y_1),1}\left(\mathbb{s}i^k\right) - p_{(x_1,y_1),1} & \cdots & p'_{(x_1,y_1),n}\left(\mathbb{s}_i^k\right) - p_{(x_1,y_1),n}\\ \vdots & \ddots & \vdots\\ p'_{(x_m,y_m),1}\left(\mathbb{s}_i^k\right) - p_{(x_m,y_m),1} & \cdots & p'_{(x_m,y_m),n}\left(\mathbb{s}_i^k\right) - p_{(x_m,y_m),n}\end{pmatrix}.\end{aligned}$$

Subsequently, the value of $b^{\mathrm{overall}}(\mathbb{s}_i^k)$ can be calculated according to the following equation:

$$b^{\mathrm{overall}}\left(\mathbb{s}_i^k\right) = \sum_{v=1}^{m}\sum_{h=1}^{n}\left[p'_{(x_v,y_v),h}\left(\mathbb{s}_i^k\right) - p_{(x_v,y_v),h}\right]. \tag{41}$$

Finally, the candidate best task schedule for s_i that satisfies the following equation will be selected as the best task schedule, that is,

$$\mathbb{s}_i^{\mathrm{best}} = \arg \max_{\mathbb{s}_i^k\in \mathbb{s}_i^{\mathrm{cand}}} \left[b^{\mathrm{overall}}\left(\mathbb{s}_i^k\right)\right]. \tag{42}$$

However, the best task schedules for neighboring sensors might contradict each other. To address this problem and achieve cooperative monitoring, the random back-off and decision-announcement scheme is designed and presented in the third step.

Fig. 5 further depicts an example to illustrate the operations in this step. Recall that in the example given in step 1, the unscheduled sensor is s_2, the current local bottleneck space–time point $o^{\mathrm{bott}}_{2,\mathrm{loc}}$ is $(g_{(1,1)}, t_2)$. Therefore, the set of feasible task schedules for s_i that has the monitoring contribution to $o^{\mathrm{bott}}_{2,\mathrm{loc}}$, which is denoted by $\mathbb{S}_2^{\mathrm{bott}}$, is listed as follows:

$$\begin{aligned} \mathbb{s}_2^1 &= (r_2, 1, 1, 0)\\ \mathbb{s}_2^2 &= (r_2, 0, 1, 1)\\ \mathbb{s}_2^3 &= (r_3, 0, 1, 0).\end{aligned}$$

For each task schedule $\mathbb{s}_2^k \in \mathbb{S}_2^{\mathrm{bott}}$, the local monitoring quality matrix $M^{\mathrm{QoM}}_{2,\mathrm{loc}}$ is updated accordingly, and each corresponding matrix $M'^{\mathrm{QoM}}_{2,\mathrm{loc}}(\mathbb{s}_2^k)$ can be obtained by applying (37), which is given in Fig. 5. The new local bottleneck space–time point $o'^{\mathrm{bott}}_{2,\mathrm{loc}}(\mathbb{s}_2^k)$ when s_2 selects $\mathbb{s}_2^k$ can be obtained according to (38), which is marked in red ink in each matrix $M'^{\mathrm{QoM}}_{2,\mathrm{loc}}(\mathbb{s}_2^k)$ shown in Fig. 5. Each of benefit $b^{\mathrm{bott}}(\mathbb{s}_2^k)$ and overall benefit $b^{\mathrm{overall}}(\mathbb{s}_2^k)$ of s_2 that selects $\mathbb{s}_2^k$ is calculated and given in Fig. 5. Finally, the best task schedule for s_2 uniquely determined according to (40) and (42), which is $\mathbb{s}_2^2$, and it is marked by a green dotted box in Fig. 5.

Step 3 (The random back-off and decision-announcement scheme): After obtaining the best task schedule for each sensor s_i, each s_i should wait for a specified amount of time to avoid conflicts between the task schedules of neighboring sensors and to ensure that the better task schedule is prioritized. To achieve this, the waiting time for each sensor s_i is determined based on the benefit and the overall benefit of its task schedule. Let $\mathbb{T}_i^{\mathrm{wait}}$ denote the waiting time for s_i, which is given in the following equation:

$$\mathbb{T}_i^{\mathrm{wait}} = \left[\omega_1 \cdot b^{\mathrm{bott}}\left(\mathbb{s}_i^{\mathrm{best}}\right) + \omega_1 \cdot b^{\mathrm{overall}}\left(\mathbb{s}_i^{\mathrm{best}}\right)\right]^{-1} \tag{43}$$

where ω_1 is much larger than ω_2, and reflecting the primary objective of the proposed algorithm is to maximize the cooperative detection probability of bottleneck space–time point. In addition, it aims to enhance the cooperative detection probability across all space–time points.

Each sensor s_i waits for the waiting time before initiating its monitoring task according to its task schedule. If the sensor s_i does not receive a task schedule broadcast from its neighbors after the waiting time has elapsed, s_i can broadcast its task schedule $\mathbb{s}_i^{\mathrm{best}}$ and begin to monitor the given region. Each sensor s_j that is loser in the random back-off procedure should update its local monitoring quality matrix $M^{\mathrm{QoM}}_{j,\mathrm{loc}}$ and local current sensor scheduling results matrix $M^{\mathrm{sch}}_{j,\mathrm{loc}}$ according to the received task schedule and then redesign its task schedule according to the operations presented in step 2.

Table II summarizes the process of the proposed JMLA algorithm. The proposed JMLA algorithm consists of three phases. The space–time partitioning phase is summarized in steps 1–3. Subsequently, the monitoring contribution computation phase is summarized in step 4. Finally, steps 5–8 summarize the scheduling phase.

To demonstrate the feasibility and efficiency of the proposed JMLA algorithm, a time complexity analysis is provided as follows. During the space–time partitioning phase, the monitoring region is initially divided into $c_1 \times c_2$ grids, and then, the PV power for the next day is evaluated. The time complexity of the space partitioning process is $O(C)$, where $C = c_1 \times c_2$. The PV power evaluation involves similarity-based calculation using matrices. Because there are 365 days in one year and the number of selected weather factors that influence PV power generation is five, the complexity of PV power evaluation is $O(365 \cdot n \cdot 5)$, where n is the number of meteorological data records per day. Considering that 365 and 5 are constants, the actual time complexity of PV power evaluation is $O(n)$. Finally, the length of one time slot and cycle calculation involves dividing a day into β cycles, each containing time α slots. The process in this step is straightforward, with a time complexity of $O(\alpha \cdot \beta)$.

Subsequently, during the monitoring contribution computation phase, each sensor s_i initially identifies the grids it can fully cover and simultaneously checks whether these grids are fully covered by one or more sensors. The time complexity of this step is $O(m \cdot C)$, where m is the number of deployed sensors and C is the number of grids. Then, each sensor s_i calculates the detection probability to the grids that can be fully covered by s_i. Assume that s_i can fully cover k_i grids. The time complexity of this step is $O(m \cdot k_i)$.

Finally, during the scheduling phase, each sensor independently determines its priority for entering the sensing state based on its monitoring contribution. Therefore, the decision-making time complexity for each sensor is $O(1)$ as it only needs to make a decision based on its own information and that of its neighboring sensors. When the number of deployed sensors is m, the overall time complexity of this phase is $O(m)$.

In summary, the time complexity of the space–time partitioning phase is $O(C + n + \alpha \cdot \beta)$, the time complexity of the monitoring contribution computation phase is $O(m \cdot C + m \cdot k_i)$, and the time complexity of the scheduling phase is $O(m)$. Therefore, the overall time complexity of the JMLA algorithm is $O(C + n + \alpha \cdot \beta + m(C + k_i + 1))$. This indicates that the time complexity of the algorithm is manageable and scales linearly with the key parameters.

The JMLA algorithm effectively optimizes energy consumption and network communication through dynamic scheduling and self-organizing mechanisms. Sensors determine their status and activation time based on their monitoring contribution, and the sensor with the largest monitoring contribution will be prioritized for activation, thereby enhancing the monitoring quality. The base station, equipped with robust computational and storage capabilities, performs data aggregation and preprocessing to reduce redundant data and transmission load. The use of multihop communication further enhances data transmission efficiency, reducing the energy consumption associated with long-distance communication. In practical applications, the JMLA algorithm precisely calculates the

TABLE II
JMLA ALGORITHM

Inputs:

1. A set of solar power sensors with adjustable sensing radius $S = \{s_1, s_2, \ldots, s_n\}$.
2. The set of historical meteorological data M_{met}^{hist}
3. Some main features of the sensor: E, $\mathbb{R}$, $u_{i,m}^{dis}$.

Outputs:

1. The set of scheduled sensors that form an area coverage

Step 1. /* ***Space Partitioning*** */
  Partition the R into $c_1 \times c_2$ identical grids;
Step 2. /* ***PV Power Evaluation for the Next Day*** */
  Evaluating the PV power for the next day based on similarity-based calculation;
Step 3. /* ***Calculating the Length of One Time Slot and Cycle*** */
  Calculating the length of one time slot and cycle according to Exp. (19) and Exp. (27), respectively
Step 4. /* ***Monitoring Contribution Computation*** */
  If (sensor s_i can completely cover the grid $g_{(x,y)}$)
    Calculating $p(s_i, g_{(x,y)})$ according to Exp. (29)
  else if (the set of sensors $\Phi_{(x,y)}^{cov}$ can completely cover the grid $g_{(x,y)}$)
    Calculates $p(\Phi_{(x,y)}^{cov}, g_{(x,y)})$ according to Exp. (30)
  else if (sensor s_i partially cover the grid $g_{(x,y)}$**)**
    $p(s_i, g_{(x,y)}) = 0$
Step 5. /* ***Finding the "Bottleneck Space-Time Point"*** */
  For(each unscheduled sensor s_i) {
    Establishing local current sensor scheduling result matrix $M_{i,loc}^{sch}$ according to Exp. (35)
  Calculating the local monitoring quality matrix $M_{i,loc}^{QoM}$ based on $M_{i,loc}^{sch}$
    Finding the local bottleneck space-time point $o_{i,loc}^{bott}$ according to Exp. (34)
}
Step 7. /* ***Scheduling Sensor with the Maximum Contribution to Monitor the Bottleneck Space-Time Point*** */
  For (each unscheduled sensor s_i **)** {
    For (each feasible task schedule $\mathbb{s}_i^k \in \mathbb{S}_i^{bott}$) {
      Calculating the updated local monitoring quality matrix $M'^{QoM}_{i,loc}(\mathbb{s}_i^k)$ according to Exp. (37)
      Identifying the new local bottleneck space-time point $o'^{bott}_{i,loc}(\mathbb{s}_i^k)$ according to Exp. (38)
      Selecting the best task schedule $\mathbb{s}_i^{best}$ according to Exps. (40) and (42).
      }
    }
Step 8. /* ***The Random Back-off and Decision-Announcement*** */
  For (each s_i)
  Calculating the waiting time $\mathbb{T}_i^{wait}$ according to Exp. (43)
  s_i countdown from $\mathbb{T}_i^{wait}$ to 0
If (there is no schedule broadcasted from $\mathbb{N}(s_i)$**)**
  s_i broadcast its task schedule $\mathbb{s}_{best}^{s_i}$ and begins the monitoring task
Else
  s_i receive schedule broadcasted from $\mathbb{N}(s_i)$ and updates its $M_{i,loc}^{sch}$
  Executes the operations from step 7 to step 8 again;
Until all the sensors are scheduled

TABLE III
SIMULATION PARAMETERS

Parameters	Value
Monitoring Region	400m×400m
Number of Sensors	200-1000
Grid Size	2m-10m
Minimum Sensing Radius	10m
Maximum Sensing Radius	20-40m
Communication Radius	40-80m
Battery Capacity	10.3 KJ
Maximum Power Consumption Rate	0.2 J/s
PV Power and Meteorological Data	Dataset from *DKA* Solar Center
Deployment Strategy	Randomly

monitoring contribution of each sensor, improving data collection accuracy. Its self-organizing nature and distributed computing approach ensure good scalability, making it suitable for large-scale sensor networks. In addition, the dynamic scheduling mechanism enhances system reliability, allowing the network to maintain effective monitoring capabilities even when some sensors fail or deplete their energy. In summary, the JMLA algorithm demonstrates efficiency, energy savings, reliability, and scalability in practical applications, making it highly valuable for real-world use.

To provide a detailed explanation of the operational mechanism of our proposed system in practical applications, the main operational steps of the JMLA algorithm and their execution details across different nodes are described as follows. The system first performs initialization. WRSNs are first deployed in the monitoring region, ensuring that the positions and coverage areas of all sensors are known. The specialized base stations within the system are responsible not only for data collection but also for acquiring historical meteorological data and weather forecasts via an internet connection. Subsequently, during daily operations, the base station selects a day from historical data that resembles the current weather forecast and applies the PV power function to calculate the solar energy supply for each time period of the day, which is then distributed to the sensors through multihop transmission. Each sensor, based on the solar energy estimation, divides the day into multiple periods and time slots and independently calculates its energy budget. In addition, sensors autonomously form space–time points based on predefined parameters and calculate cooperative detection probabilities for each space–time point within their local monitoring space–time region. Each sensor independently identifies its local bottleneck space–time points, and the sensor with the largest monitoring contribution will be prioritized for activation. Sensors regularly collect environmental data and transmit it to the base station via multihop transmission. The sink node then performs data fusion and analysis, generating regional monitoring reports, and ultimately transmits the processed data to a data processing terminal for further analysis and storage. Through these steps, the JMLA algorithm not only operates efficiently and reliably in practical applications but also ensures effective data transmission and processing through a distributed approach, thereby achieving high motoring quality.

V. PERFORMANCE EVALUATIONS

This section investigates the performance improvement of the proposed JMLA algorithm by comparing the existing modified improved genetic algorithm for maximizing area coverage in WSNs (MIGA) [9] and MSQAC [15]. The MIGA is a novel and efficient genetic algorithm, which aims to maximize area coverage in WRSNs. This study adopted the battery-powered sensors with fixed sensing radius and utilized the BSM. MSQAC further adopted solar-powered sensors with fixed sensing radius and utilized the PSM. This study assumed that solar power is constant and optimized sensor scheduling to maximize monitoring quality by balancing energy availability. The following presents the simulation environment, performance metrics, and simulation results.

A. Simulation Environment

The simulation parameters are given in Table III. The size of the monitoring region R is 400 m $\times$ 400 m. The deployed WRSNs consist of 200–1000 sensors with adjustable sensing radii which are randomly deployed in R. The grid size ranges from 2 to 10 m. The minimum sensing radius and the maximum sensing radius of each sensor are 10 and 40 m, respectively. The communication radius is twice the maximum sensing radius. The battery capacity E of each sensor is set at 10.3 kJ [29]. The maximum power consumption rate is 0.2 J/s. The historical daily PV power data and the corresponding meteorological data are from the dataset of Desert Knowledge Australia (DKA) solar center. This dataset provides high-quality data related to PV power and corresponding meteorological data in Northern Australia, which is an authoritative dataset widely used in the analysis and prediction of PV power [30].

B. Simulation Results

It is noted that weather conditions are generally divided into three main categories: clear, cloudy, and rainy days. This classification has been validated through the analysis of the dataset. Subsequently, Fig. 6(a)–(c) presents the comparison of JMLA, MSQAC, and MIGA in terms of the monitoring quality on clear, cloudy, and rainy days, respectively. In this experiment, the number of deployed sensors ranges from 200 to 1000, and the maximum sensing radius of each sensor is set from 20 to 40 m. Since both the MSQAC and MIGA algorithms adopt fixed sensing radius sensors, these two algorithms only adjust the sensing radius of each sensor from 20 to 40 m. Besides, the MIGA algorithm did not adopt the rechargeable sensors and schedules all the sensors to be activated all the time. It is assumed that in each experiment, the number of deployed sensors is $|S|$, and each cycle has α time slots. To ensure fairness, the MIGA algorithm schedules $|S|/\alpha$ sensors to be activated in one time slot. Furthermore, to accurately reflect the actual monitoring quality of the area coverage constructed by the three compared algorithms in different scenarios, the monitoring quality results presented in each experiment are the averages obtained from multiple repeated experiments conducted in the same scenario.

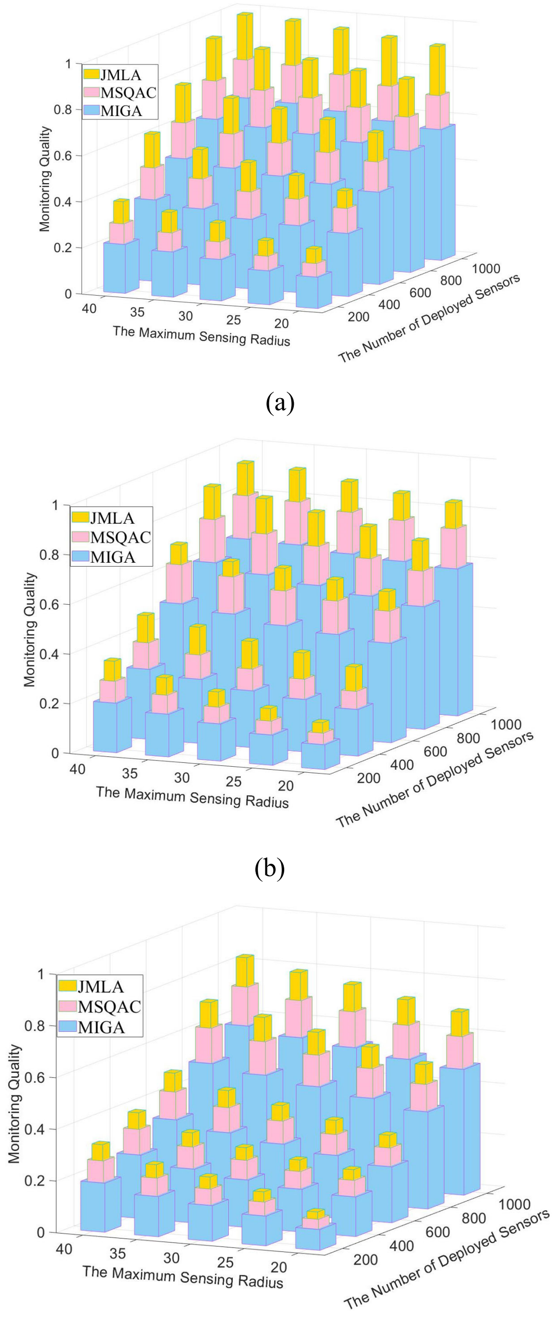


Fig. 6. Comparison of JMLA, MSQAC, and MIGA in terms of monitoring quality by varying numbers of deployed sensors and the maximum sensing radius. (a) On clear days. (b) On cloudy days. (c) On rainy days.

As depicted in Fig. 6, a common trend shows that the monitoring quality of the three compared algorithms increases with the number of deployed sensors and the sensing radius of each sensor. This is because the larger number of deployed sensors will increase the number of active sensors in each time slot, leading to higher monitoring quality. In addition, sensors with a larger sensing radius can cover more grids and support a higher detection capacity. The other trend of the three compared algorithms is that the monitoring quality on clear days is higher than those on cloudy and rainy days. This is because compared to cloudy and rainy days, solar power can generate higher and more stable PV power on clear days, allowing solar-powered sensors to harvest more energy. Sensors with a more abundant power supply can operate for longer durations or select the larger sensing radius, thereby achieving higher monitoring quality.

In comparison, the proposed JMLA algorithm outperforms the MSQAC and MIGA algorithms in terms of monitoring quality in all cases. This occurs because, unlike MSQAC and MIGA, JMLA considers the variation of PV power affected by weather conditions. It initially evaluates the available solar energy by forecasting the PV power for the next day. The scheduling scheme of each sensor is based on the evaluated available solar energy, which supports higher detection capacity while avoiding power depletion. On the contrary, the MSQAC algorithm assumes that the PV power is constant. Consequently, the evaluated available solar energy might be overly conservative to prevent some sensors from becoming non-operational due to power depletion. This leads to a conservative scheduling scheme for each sensor. The MIGA algorithm utilizes the BSM model to evaluate the detection capacity of each sensor and the monitoring quality of all active sensors to each grid. However, it cannot accurately identify the bottleneck space–time points, and the selected sensors might not have the maximum monitoring contribution to these points, thereby limiting the performance of the constructed area coverage.

Fig. 7 further compares the there compared algorithms in terms of monitoring quality by varying the number of deployed sensors and the grid size on clear, cloudy, and rainy days. The common trend, similar to the results shown in Fig. 6, is that the monitoring quality of the three algorithms increases with the number of deployed sensors, and the monitoring quality obtained by the three algorithms on clear days is higher than that on cloudy and rainy days. Besides, the monitoring qualities of the three algorithms decrease with the grid size increase. This occurs because more sensors are required to cover one grid with the larger size and hence leads to lower monitoring quality when the grid size grows.

In comparison, the proposed JMLA algorithm achieves the best performance in terms of monitoring quality. This occurs because the proposed JMLA algorithm can adjust the sensing radius of each sensor during the sensor scheduling phase, which can not only completely coverer more grids with larger sizes according to extend the sensing radius of each sensor but also have a higher utilization of detection capacity of each sensor than that of MSQAC and MIGA algorithms. This scheme allows JMLA to exploit more opportunities for enhancing monitoring quality to bottleneck space–time points. In addition, the JMLA adopts the PSM and considers the cooperative detection between neighboring sensors, which schedules sensors depending on a contribution-based strategy for cooperative sensing. Therefore, the sensor with the maximum monitoring contribution can be prior scheduled to monitor the bottleneck space–time point, resulting in a higher monitoring quality.

In Fig. 8, the experiment randomly selects nine locations from the monitoring region R, aiming to have a closer

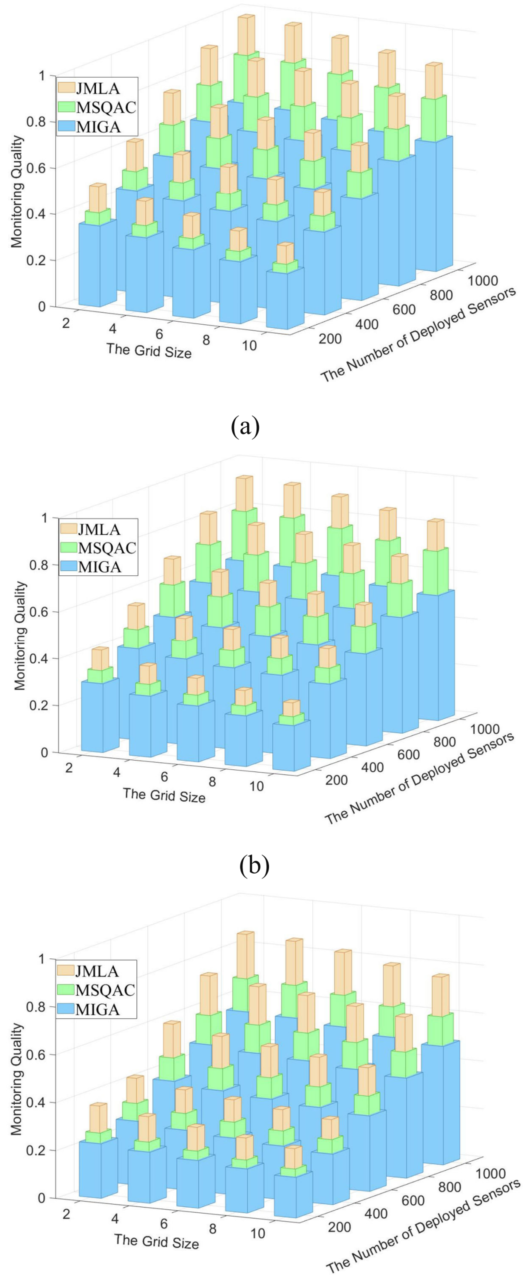


Fig. 7. Comparison of JMLA, MSQAC, and MIGA in terms of monitoring quality by varying numbers of deployed sensors and the grid size. (a) On the clear days. (b) On cloudy days. (c) On rainy days.

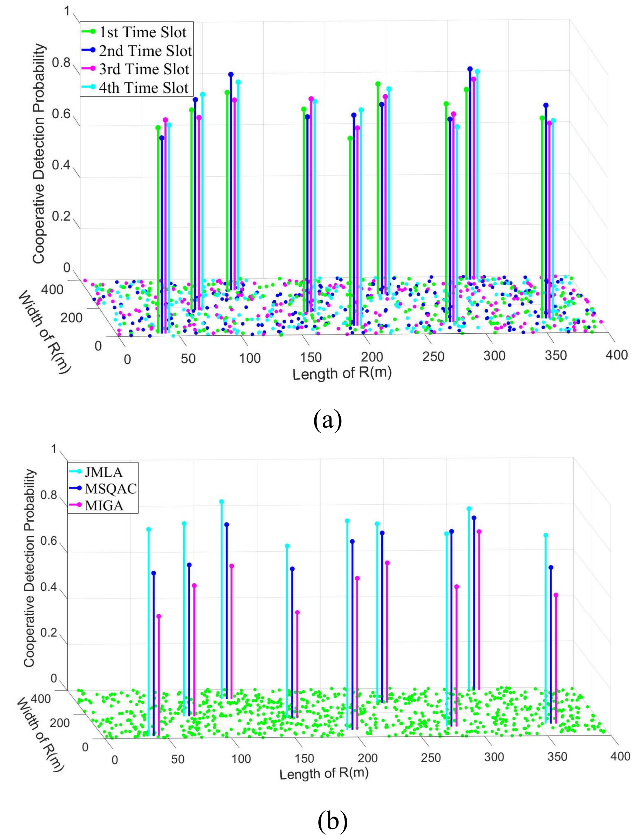


Fig. 8. Comparison of cooperative detection probability on nine randomly selected observed grids. (a) Cooperative detection probability of JMLA in different time slots within one cycle. (b) Comparison of three compared algorithms in terms of monitoring quality.

observation of the performance of three compared algorithms in terms of the monitoring quality. Fig. 8(a) first shows the cooperative detection probability on nine randomly selected locations on four time slots within one cycle by applying the proposed JMLA algorithm. As shown in Fig. 8(a), the cooperative detection probabilities of the selected locations on four time slots within one cycle are similar. It implies that the proposed JMLA algorithm supports good stability of cooperative detection probability on different space–time points. This is because the JMLA initially identifies the local bottleneck space–time point within the local monitoring space–time region of each sensor and then tries to schedule the sensors with the maximum monitoring contribution to be activated, which aims to enhance the cooperative detection probability of the bottleneck space–time point while concurrently enhancing the cooperative detection probability of all space–time points. Fig. 8(b) further compares the monitoring quality of nine selected locations by applying three compared algorithms. The selected locations are identical to those in Fig. 8(a). The performance of the proposed JMLA algorithm is superior to that of the MSQAC and MIGA algorithms in terms of both monitoring quality and the stability of monitoring quality across different locations.

Fig. 9 finally investigates solar energy utilization (SEU) by applying JMLA and MSQAC algorithms on nine randomly selected clear, cloudy, and rainy days. Because the MIGA algorithm adopts the battery-powered sensor, it is not included in this experiment. In Fig. 9, the practical solar energy (PSE) and the total amount of charged energy (TCE) of each sensor on each day are presented. It is noted that the MSQAC algorithm schedules sensors under the assumption that the PV power is constant; consequently, the MSQAC algorithm uses the average PV power of each month as the daily PV power for each sensor during that month in this experiment. Because the SEU of each sensor differs within a given area coverage, the SEUs presented in Fig. 9 in this study are the averages

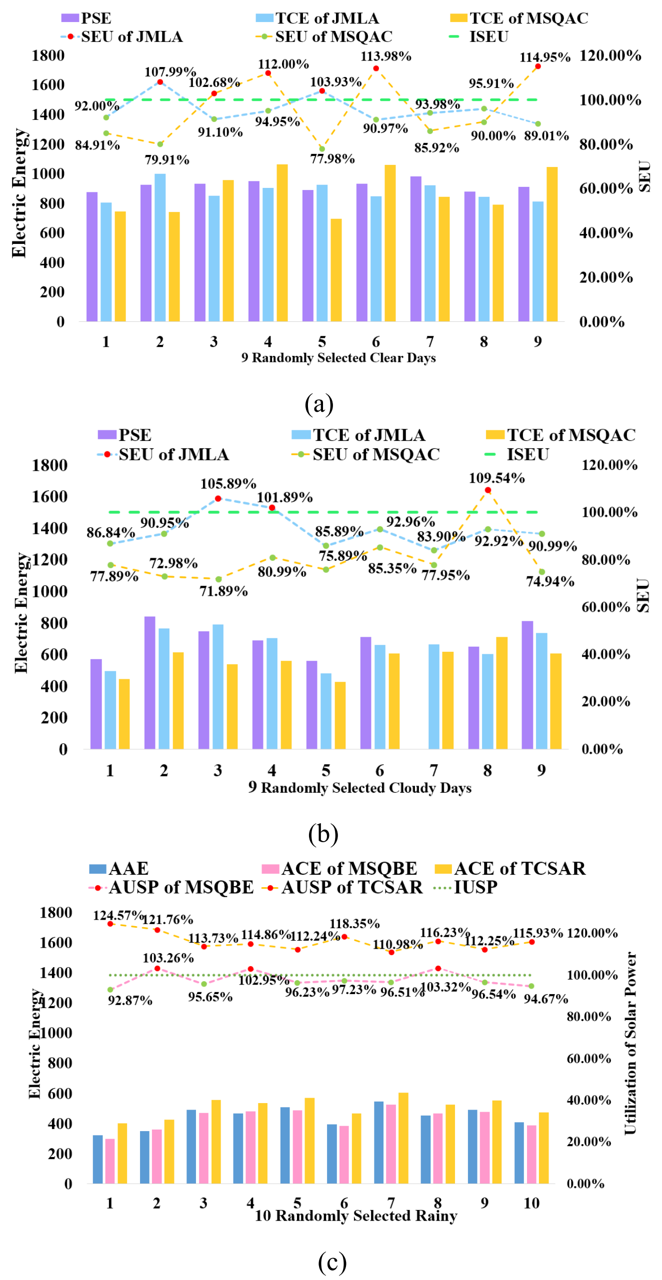


Fig. 9. Comparison of JMLA and MSQAC in terms of SEU. (a) On clear days. (b) On cloudy days. (c) On rainy days.

across all sensors within the area coverage formed by the two compared algorithms.

The value of SEU can be calculated by applying the following equation:

$$\mathrm{SEU} = \frac{\mathrm{TCE}}{\mathrm{PSE}}. \tag{44}$$

The ideal SEU (ISEU) is achieved when SEU reaches 100%, indicating a balance between the TCE and the PSE. In Fig. 9, the red point indicates that the SEU on that day is less than 100%. This reflects that the total energy consumed by each sensor in a day exceeds the total energy obtained from solar power, posing a risk of sensor failure due to battery depletion. The green point represents that the actual utilization of solar power (AUSP) is less than 100%. This reflects that the total energy consumed by each sensor in a day is less than the total energy that can be obtained from solar power, leading to a waste of solar power and consequently limiting the performance of the network. Therefore, the closer the SEU is to the ISEU, the better the performance in terms of SEU.

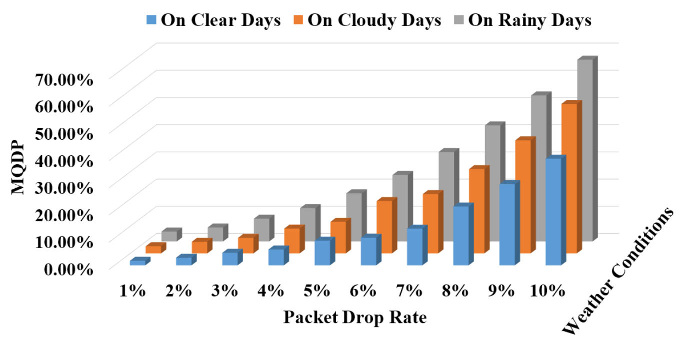


Fig. 10. Comparison of the overall performance of the JMLA algorithm by varying the packet drop rate.

As shown in Fig. 9, the performance of JMLA in terms of SEU on rainy days is better than that on clear and cloudy days. This occurs because the PV power on rainy days is relatively low but more stable than that on clear and cloudy days. The performance of JMLA in terms of SEU on cloudy is the worst. This occurs because solar energy on cloudy days is influenced by random cloud cover, leading to relatively more unstable PV power compared to rainy and cloudy days. In general, the performance of JMLA is better than that of MSQAC in all cases. This occurs because the proposed JMLA employs similarity-based calculation to estimate the PV power on the next day, aiming to achieve an accurate estimation of the next day's available solar energy. On the contrary, the MSQAC assumes that the PV power is constant throughout each day, which will lead to significant discrepancies with the actual solar power, that is, the discrepancies between PSE and TCE by applying MSQAC. Therefore, the area coverage formed by the MSQAC poses a significant risk of sensor failure due to battery depletion on rainy days and results in a substantial waste of solar energy on clear and cloudy days.

Fig. 10 investigates the impact of packet drop rate on the overall performance of the JMLA algorithm. Recall that the objective function in the problem statement, the main metric for measuring the overall performance of the JMLA algorithm, is the monitoring quality of the constructed area coverage. Therefore, the monitoring quality degradation percentage (MQDP) is adopted to measure the impact of packet drop rate on the overall performance of the area coverage achieved by the JMLA algorithm. The value of MQDP can be calculated by applying the following equation:

$$\mathrm{MQDP} = \left(1 - \frac{Q^{\mathrm{JMLA}}_{\mathrm{pd}=0}}{Q^{\mathrm{JMLA}}_{\mathrm{pd}=k}}\right) \times 100\% \tag{45}$$

where $Q^{\mathrm{JMLA}}_{\mathrm{pd}=k}$ represents the monitoring quality of the area coverage achieved by the JMLA algorithm when the packet drop rate is $k\%$.

Despite influences such as network conditions, the packet drop rate is temporally variable and may differ at different times. However, in area coverage applications such as smart agriculture and environmental monitoring, data collection and transmission frequencies are relatively stable, and the physical environments of the deployed WRSNs are comparatively stable. Therefore, over shorter periods, such as 1–2 h, the network conditions and packet drop rates are typically stable. In this study, considering the battery capacity and the discharging rate of sensors, the length of one time slot is generally around 20–40 min. Consequently, it can be assumed that the packet drop rate remains stable within one time slot. Based on this reasonable assumption, the cooperative detection probability of active sensors scheduled by JMLA to one grid at all time points within the same time slot is considered to be identical. Thus, despite time-varying packet drop rates, the monitoring quality is still regarded as the weakest cooperative detection probability among all the space–time points.

In this experiment, the number of deployed sensors is 800, the maximum sensing radius of each sensor is 30 m, and the grid size is 6 m. As shown in Fig. 10, a common trend shows that the MQDP of the area coverage achieved by the JMLA algorithm increases with the packet drop rate under all weather conditions. This occurs because, when the packet drop rate increases, the cooperative detection among sensors may be affected. Sensors rely on the exchange of information with each other to determine which areas require more intensive monitoring. If packets are lost, this information may not be accurately transmitted, thus impacting the overall monitoring quality. Furthermore, a higher packet drop rate necessitates more frequent retransmissions of data by sensors, which increases the energy consumed for data transmission. This scenario compels sensors to either shorten their sensing radii or reduce their active periods to conserve energy, consequently leading to a decline in detection capabilities and a degradation of monitoring quality. The other trend is that the performance of JMLA in terms of MQDP on clear days is better than that on cloudy and rainy days. This occurs because sensors can obtain more energy from solar power on clear days than on cloudy days and rainy days. Therefore, although data retransmission induced by packet drop consumes additional energy, reducing the amount of power available for sensing per cycle and weakening the sensor's detection capability, the impact is relatively minor on sunny days due to the increased energy acquisition by the sensors.

Fig. 11 further investigates the impact of path loss on JMLA in terms of monitoring quality. Similar to the experiment shown in Fig. 10, the MQDP is adopted to measure the impact of path loss on the monitoring quality of the area coverage achieved by the JMLA algorithm. Recall that λ and γ represent the path loss exponents of the sensing signal strength of the sensor, respectively. In this experiment, the value of MQDP can be calculated by applying the following equation:

$$\mathrm{MQDP} = \left(1 - \frac{Q^{\mathrm{JMLA}}_{\lambda=0.001,\gamma=1}}{Q^{\mathrm{JMLA}}_{\lambda=a,\gamma=b}}\right) \times 100\%$$

where $Q^{\mathrm{JMLA}}_{\lambda=a,\gamma=b}$ represents the monitoring quality of the area coverage achieved by the JMLA algorithm when $\lambda = a$ and $\gamma = b$, respectively. In this experiment, values of $\lambda =$ 0.001 and $\gamma = 1$ are considered ideal, as they result in very low path loss.

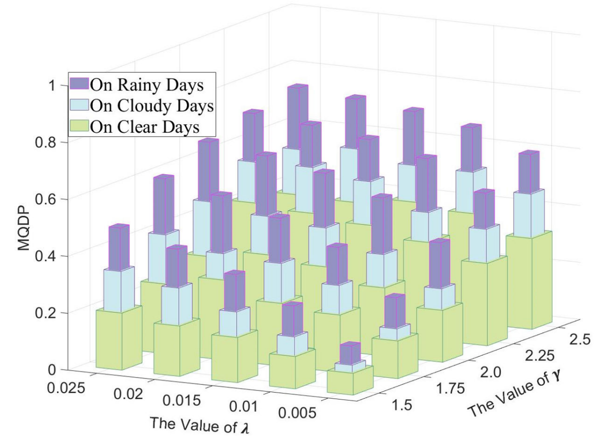


Fig. 11. Comparison of the overall performance of the JMLA algorithm by varying the values of λ and γ.

The number of deployed sensors, the maximum sensing radius of each sensor, and the grid size are the same as those in the experiment shown in Fig. 10. As shown in Fig. 11, under all weather conditions, the value of MQDP increases with the values of λ and γ. This occurs because higher λ values intensify path loss, leading to a more rapid decay in sensor signal strength with increasing distance, which reduces the effective coverage area and thus decreases monitoring quality. In addition, a higher γ accelerates the rate at which detection probability decays, further degrading the monitoring quality. Furthermore, the increase in MQDP is the least on clear days and the most on rainy days. This difference is mainly because sensors can obtain more energy from solar power on clear days, enabling them to maintain larger sensing radii and longer active time, which is conducive to reducing the impact of path attenuation on the sensor's detection capabilities.

VI. Conclusion

In this article, a novel area coverage algorithm, called JMLA, is proposed for WRSNs. The proposed JMLA adopts the adjustable sensing radius sensors, aiming to maximize the monitoring quality of the constructed area coverage. Two main issues have been investigated while designing the JMLA algorithm. The first one is to maximize the monitoring quality of all locations within the given monitoring region at any time by considering the cooperative detection of all the deployed sensors. To achieve this, the JMLA initially partitions the monitoring region R and the continuous time into several identical grids and time slots. These grids and time slots can form multiple space–time points, which enables the JMLA to evaluate the cooperative detection probability of active sensors to any location at any time. Then, the JMLA identifies the bottleneck space–time point that has the weakest cooperative detection probability and schedules the sensor with the maximum monitoring contribution to enhance the cooperative detection probability of the bottleneck space–time point while concurrently enhancing the cooperative detection probability of all space–time points. The second issue is to ensure the balance between the acquired energy and consumed energy of each sensor in each day, aiming to perpetuate the network lifetime. To achieve this, the JMLA algorithm designs an energy allocation scheme to balance the acquired and consumed energy of each sensor. It selects a day in the previous year with the most similar weather conditions to estimate the acquired energy on the following day. Then, it divides the time into several equal-length cycles and time slots and allocates the same amount of energy for each sensor within each cycle. This allocation includes a reserve for nighttime sensor operation.

The experiment study demonstrates that the proposed JMLA outperforms the compared algorithms in terms of monitoring quality, stability of the monitoring quality, and SEU. However, the proposed JMLA algorithm only adopts the static range-adjustable sensors to enhance the utilization of detection capability of each sensor. Compared to static range-adjustable sensors, mobile sensors with adjustable sensing radius have a greater capability to enhance surveillance quality. Therefore, our future work will further employ the heterogeneous WRSNs, which include some mobile sensors to further improve the network's monitoring quality.

References

[1] X. Li et al., "Using sensor network for tracing and locating air pollution sources," *IEEE Sensors J.*, vol. 21, no. 10, pp. 12162–12170, May 2021.

[2] A. Sharma and S. Chauhan, "Sensor fusion for distributed detection of mobile intruders in surveillance wireless sensor networks," *IEEE Sensors J.*, vol. 20, no. 24, pp. 15224–15231, Dec. 2020.

[3] S. N. Daskalakis, G. Goussetis, S. D. Assimonis, M. M. Tentzeris, and A. Georgiadis, "A uW Backscatter-Morse-Leaf sensor for low-power agricultural wireless sensor networks," *IEEE Sensors J.*, vol. 18, no. 19, pp. 7889–7898, Oct. 2018.

[4] H. M. Ammari, "Investigating physical security in stealthy lattice wireless sensor networks using k-barrier coverage," *Ad Hoc Netw.*, vol. 89, pp. 142–160, Jun. 2019.

[5] Z.-G. Chen, Y. Lin, Y.-J. Gong, Z.-H. Zhan, and J. Zhang, "Maximizing lifetime of range-adjustable wireless sensor networks: A neighborhood-based estimation of distribution algorithm," *IEEE Trans. Cybern.*, vol. 51, no. 11, pp. 5433–5444, Nov. 2021.

[6] P. Zhou, C. Wang, and Y. Yang, "Static and mobile target k-coverage in wireless rechargeable sensor networks," *IEEE Trans. Mobile Comput.*, vol. 18, no. 10, pp. 2430–2445, Oct. 2019.

[7] P. Xu, J. Wu, C.-Y. Chang, C. Shang, and D. S. Roy, "MCDP: Maximizing cooperative detection probability for barrier coverage in rechargeable wireless sensor networks," *IEEE Sensors J.*, vol. 21, no. 5, pp. 7080–7092, Mar. 2021.

[8] Z. Liu, G. Jiang, W. Jia, T. Wang, and Y. Wu, "Critical density for K-coverage under border effects in camera sensor networks with irregular obstacles existence," *IEEE Internet Things J.*, vol. 11, no. 4, pp. 6426–6437, Feb. 2024.

[9] N. T. Hanh, H. T. T. Binh, N. X. Hoai, and M. S. Palaniswami, "An efficient genetic algorithm for maximizing area coverage in wireless sensor networks," *Inf. Sci.*, vol. 488, pp. 58–75, Jul. 2019.

[10] F. Tossa, W. Abdou, K. Ansari, E. C. Ezin, and P. Gouton, "Area coverage maximization under connectivity constraint in wireless sensor networks," *Sensors*, vol. 22, no. 5, p. 1712, Feb. 2022.

[11] X. Gao, Z. Chen, F. Wu, and G. Chen, "Energy efficient algorithms for k-sink minimum movement target coverage problem in mobile sensor network," *IEEE/ACM Trans. Netw.*, vol. 25, no. 6, pp. 3616–3627, Dec. 2017.

[12] P. Si, J. Ma, F. Tao, Z. Fu, and L. Shu, "Energy-efficient barrier coverage with probabilistic sensors in wireless sensor networks," *IEEE Sensors J.*, vol. 20, no. 10, pp. 5624–5633, May 2020.

[13] Q. Yang, S. He, J. Li, J. Chen, and Y. Sun, "Energy-efficient probabilistic area coverage in wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 64, no. 1, pp. 367–377, Jan. 2015.

[14] X. Fan, F. Hu, T. Liu, K. Chi, and J. Xu, "Cost effective directional barrier construction based on zooming and united probabilistic detection," *IEEE Trans. Mobile Comput.*, vol. 19, no. 7, pp. 1555–1569, Jul. 2020.

[15] B. Dande, C.-Y. Chang, W.-H. Liao, and D. S. Roy, "MSQAC: Maximizing the surveillance quality of area coverage in wireless sensor networks," *IEEE Sensors J.*, vol. 22, no. 6, pp. 6150–6163, Mar. 2022.

[16] H. Mostafaei, A. Montieri, V. Persico, and A. Pescapé, "A sleep scheduling approach based on learning automata for WSN partialcoverage," *J. Netw. Comput. Appl.*, vol. 80, pp. 67–78, Feb. 2017.

[17] J. Yu, S. Wan, X. Cheng, and D. Yu, "Coverage contribution area based k-coverage for wireless sensor networks," *IEEE Trans. Veh. Technol.*, vol. 66, no. 9, pp. 8510–8523, Sep. 2017.

[18] C. Luo, Y. Hong, D. Li, Y. Wang, W. Chen, and Q. Hu, "Maximizing network lifetime using coverage sets scheduling in wireless sensor networks," *Ad Hoc Netw.*, vol. 98, Mar. 2020, Art. no. 102037.

[19] Y. Kan, C.-Y. Chang, C.-H. Kuo, and D. S. Roy, "Coverage and connectivity aware energy charging mechanism using mobile charger for WRSNs," *IEEE Syst. J.*, vol. 16, no. 3, pp. 3993–4004, Sep. 2022.

[20] C. Shang, C.-Y. Chang, W.-H. Liao, and D. S. Roy, "RLR: Joint reinforcement learning and attraction reward for mobile charger in wireless rechargeable sensor networks," *IEEE Internet Things J.*, vol. 10, no. 18, pp. 16107–16120, Sep. 2023.

[21] H. Chen, X. Li, and F. Zhao, "A reinforcement learning-based sleep scheduling algorithm for desired area coverage in solar-powered wireless sensor networks," *IEEE Sensors J.*, vol. 16, no. 8, pp. 2763–2774, Apr. 2016.

[22] Z. Zhang, W. Wu, J. Yuan, and D.-Z. Du, "Breach-free sleep-wakeup scheduling for barrier coverage with heterogeneous wireless sensors," *IEEE/ACM Trans. Netw.*, vol. 26, no. 5, pp. 2404–2413, Oct. 2018.

[23] G. K. Singh, "Solar power generation by PV (photovoltaic) technology: A review," *Energy*, vol. 53, no. 1, pp. 1–13, May 2013.

[24] Y. Li, Y. Wang, and Q. Chen, "Study on the impacts of meteorological factors on distributed photovoltaic accommodation considering dynamic line parameters," *Appl. Energy*, vol. 259, Feb. 2020, Art. no. 114133.

[25] R. Gabriel, "Minimization of the Frobenius norm of a complex matrix using planar similarities," *Appl. Numer. Math.*, vol. 40, no. 3, pp. 391–414, Feb. 2002.

[26] A. Hentati, J.-F. Frigon, and W. Ajib, "Energy harvesting wireless sensor networks with channel estimation: Delay and packet loss performance analysis," *IEEE Trans. Veh. Technol.*, vol. 69, no. 2, pp. 1956–1969, Feb. 2020.

[27] D. A. Guimarães, E. P. Frigieri, and L. J. Sakai, "Influence of node mobility, recharge, and path loss on the optimized lifetime of wireless rechargeable sensor networks," *Ad Hoc Netw.*, vol. 97, Feb. 2020, Art. no. 102025.

[28] M. N. Mowla, N. Mowla, A. F. M. S. Shah, K. M. Rabie, and T. Shongwe, "Internet of Things and wireless sensor networks for smart agriculture applications: A survey," *IEEE Access*, vol. 11, pp. 145813–145852, 2023.

[29] H. Yu, C.-Y. Chang, W. Guan, Y. Wang, and D. Sinha Roy, "MC^3: A multicharger cooperative charging mechanism for maximizing surveillance quality in WRSNs," *IEEE Sensors J.*, vol. 24, no. 10, pp. 17080–17091, May 2024.

[30] M. Zhang, Y. Han, C. Wang, P. Yang, C. Wang, and A. S. Zalhaf, "Ultra-short-term photovoltaic power prediction based on similar day clustering and temporal convolutional network with bidirectional long short-term memory model: A case study using DKASC data," *Appl. Energy*, vol. 375, Dec. 2024, Art. no. 124085.

Pei Xu received the Ph.D. degree in computer science and technology from Anhui University, Hefei, China, in 2020.

He is currently a Lecturer with the School of Artificial Intelligence and Big Data, Hefei University, Hefei. His current research interests include smart sensing technology, the Internet of Things, and deep learning.

Xiaofeng Wang received the Ph.D. degree in pattern recognition and intelligent systems from the University of Science and Technology of China, Hefei, China, in 2009.

He is currently a Professor with the School of Artificial Intelligence and Big Data, Hefei University, Hefei. He has been serving as a Candidate for academic and technical leaders in Anhui, China, a High-Level Mentor in Anhui, a Top Professional and a Technical Talent in Hefei, and a Standing Director of Anhui Computer Society. His current research interests include artificial intelligence and computer vision technologies.

Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan City, Taiwan, in 1995.

He is currently a Distinguished Professor with the Department of Computer Science and Information Engineering and the Department of Artificial Intelligence, Tamkang University, New Taipei, Taiwan. His current research interests include artificial intelligence, deep learning and machine learning, the Internet of Things, and wireless sensor networks.

Dr. Chang is the Co-Chair of ACM SIGMOBILE, Taiwan. He has been serving as an Associate Guest Editor for several SCI-indexed journals, including *International Journal of Ad Hoc and Ubiquitous Computing* since 2011, *Journal of Applied Science and Engineering* since 2018, *International Journal of Distributed Sensor Networks* from 2012 to 2014, *IET Communications* in 2011, *Telecommunication Systems* in 2010, *Journal of Information Science and Engineering* in 2008, and *Journal of Internet Technology* from 2004 to 2008.

Youxi Li received the Ph.D. degree in computer science and information engineering from Tamkang University, New Taipei, Taiwan, in 2024.

His current research interests include the Internet of Things and artificial intelligence.

Diptendu Sinha Roy received the Ph.D. degree in engineering from the Birla Institute of Technology, Mesra, India, in 2010.

In 2016, he joined the Department of Computer Science and Engineering, National Institute of Technology (NIT) Meghalaya, Shillong, India, as an Associate Professor, where he has been working as the Chair of the Department of Computer Science and Engineering since January 2017. Prior to his stint at NIT Meghalaya, he worked with the Department of Computer Science and Engineering, National Institute of Science and Technology, Berhampur, India. His current research interests include software reliability, distributed and cloud computing, and the Internet of Things (IoT), specifically applications of artificial intelligence/machine learning for smart integrated systems.

Dr. Roy is a member of the IEEE Computer Society.