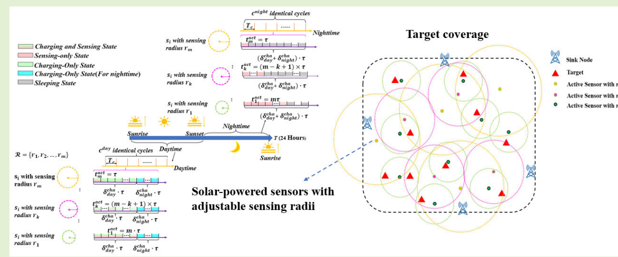
Dynamic Probabilistic Sensing for Enhanced Target Coverage in WRSNs

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

***Abstract*—In wireless sensor networks (WSNs), target coverage is a critical issue. Many existing studies have been proposed for constructing the target coverage, aiming to enhance the surveillance quality while extending the network lifetime. However, they predominantly utilize the Boolean sensing model (BSM), which may lack accuracy. In addition, these studies assumed that sensors are battery-powered with fixed sensing radius, which limits the performance of the target coverage. This article proposes a target coverage algorithm, called maximize surveillance quality while balancing energy consumption and acquisition (MSQBE), which employs probabilistic sensing model (PSM) and solar-powered sensors with adjustable sensing radii, aiming to maximize the surveillance quality while balancing the acquired and consumed energy to perpetuate the network lifetime. The MSQBE initially employs a 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 photovoltaic (PV) power function from that day to estimate the PV power for the next day, aiming to achieve an accurate estimation of the next day's PV power. Then, the MSQBE partitions the time into several identical cycles and time slots. The length of each time slot depends on the evaluation of PV power for the next day. These time slots and targets form several space–time points. Finally, the MSQBE designs the task schedule for each sensor, aiming to maximize the surveillance quality of the bottleneck space–time point. The experimental results show that the proposed MSQBE outperforms the existing method in terms of surveillance quality, surveillance stability, and utilization of solar power.**



***Index Terms*— Adjustable sensing radius, solar-powered sensor, target coverage, wireless sensor networks (WSNs).**

I. INTRODUCTION

WIRELESS sensor networks (WSNs) have been widely applied in many applications, such as environmental monitoring [1], intruder detection [2], and aerospace [24]. 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 [3], [4] and most studies concerned with these two issues. Many functions of WSNs rely on full coverage of specified objects, which can be an area, a barrier, or a set of targets. The area coverage aims to monitor the entire area [5], [6], while barrier coverage aims to detect the intruder [7], [8]. The target coverage aims to monitor a set of given targets [9], [10], [11], which are called points of interest (POIs). The target coverage has been widely applied in many applications such as culture relics, military installations, and smart homes. It is noticed that area and barrier coverage can both be converted into special cases of target coverage, this article aims to address the network lifetime and surveillance quality problems in target coverage scenarios.

The surveillance quality mainly depends on the sensing model and the scheduling algorithm, which schedules all the sensors to monitor all the targets effectively. Most existing studies depicted a sensor detection effect by applying a deterministic Boolean sensing model (BSM) [9], [10], [11]. BSM is a rough approximation of the practical sensing model, which assumes that the sensing ability of each sensor is perfect and the event that occurred in the sensing range of each sensor is definitely detected. Although BSM is easy to analyze, however, it is difficult to reflect the physical features of sensing

Manuscript received 9 May 2024; revised 20 June 2024; accepted 20 June 2024. Date of publication 11 July 2024; date of current version 15 August 2024. This work was supported in part by the Program for Scientific Research Innovation Team in Colleges and Universities of Anhui Province under Grant 2022AH010095, in part by the Program for Key Scientific Research Projects in Colleges and Universities of Anhui Province under Grant 2022AH051781, and in part by the Program for Scientific Research Fund for Talents in Hefei University under Grant 21-22RC16. The associate editor coordinating the review of this article and approving it for publication was Prof. Chen Yang. *(Corresponding author: Chih-Yung Chang.)*

Chi Zhang and Pei Xu are with the School of Artificial Intelligence and Big Data, Hefei University, Hefei 230601, China (e-mail: zhangchi_hfuu@163.com; xupeimk@163.com).

Youxi Li and Chih-Yung Chang are with the Department of Computer Science and Information Engineering, Tamkang University, New Taipei City 25137, Taiwan (e-mail: 808414022@o365.tku.edu.tw; cychang@mail.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.2024.3421656

and the evaluation of the surveillance quality by applying BSM may not be accurate enough. Contrarily, 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. Xu et al. [12] and Si et al. [13] applied PSM aims to enhance the surveillance quality of the construct target coverage. The PSM is more practicable compared to the BSM because it matches real sensing behavior. The PSM is more practicable compared to the BSM because it matches real sensing behavior. However, most of these studies adopt sensors with a fixed sensing radius. It is noted that since sensors are typically randomly deployed, a fixed sensing radius may not be flexible enough to fully utilize the detection capability of each sensor. In recent years, the usage of sensors with adjustable sensing radii has become more prevalent [4], [14], aiming to further improve the utilization of each sensor's detection probability. However, sensors with different sensing radii exhibit different energy consumption rates, leading to increased complexity in energy allocation. Although some computational intelligence methods, such as multiobjective optimization [25], can effectively solve such problems, most devices in WSNs have limited computational capabilities, making them unable to use such good methods.

The other core indicator of the coverage problem is the network lifetime. Most studies discussed prolonging the network lifetime by designing the optimal sleep-wake-up algorithm [15], energy-efficient algorithm [16], and the synchronous wireless sensor and sink placement method based on a multiobjective optimization algorithm [26]. However, the sensors applied in these studies are all battery-powered, and energy remains a limiting factor. It is impractical to replace batteries for WSNs in some large-scale deployment scenarios [17]. The WSNs will be inefficient if too many sensors run out of energy. Therefore, replacing the energy of the sensor has attracted much attention from researchers. The WSNs that can be energy replenished are called wireless rechargeable sensor networks (WRSNs). In literature, many energy-replenishing algorithms have been proposed. These algorithms can be categorized into two main classes, including the mobile charger [17], [18] and environmental energy harvesting [14], [19]. Although mobile chargers can provide stable and sufficient energy. However, the cost of deploying mobile chargers is substantial in large-scale scenarios. Environmental energy harvesting mainly includes wind power and solar power. Compared to solar power, wind power exhibits greater instability, and locations with sufficient wind resources are less common, rendering it less universally applicable. Although solar power offers some advantages, most existing studies assume that photovoltaic (PV) is constant. However, the PV power is variable and dependent on weather conditions. Consequently, the evaluation of PV power under this assumption will be conservative and rough, which limits the performance of WSNs.

As discussed above, the adoption of solar-powered sensors with adjustable sensing radii poses some new challenges.

1) Solar power is only available during the daytime, making the distribution of energy and the scheduling of sensor operations more complex.
2) Solar power energy is a dynamic energy resource and is affected by weather conditions. A conservative scheduling may lead to fully charged batteries and wasted energy during peak solar output, while an aggressive schedule for better surveillance might drain batteries quickly when the solar output is low.
3) Sensors with different sensing radii consume varying amounts of power. Smaller radii allow for longer active time but with reduced detection capability, while larger radii enable more extensive detection at the cost of shorter active time, complicating energy management due to the multiple possible schedules. Despite plenty of pioneering research, the field still lacks a comprehensive algorithm to effectively address these challenges, which in turn enhances surveillance quality while perpetuating network lifetime.

To address these challenges, this article proposes a novel target coverage algorithm, called maximize surveillance quality while balancing energy consumption and acquisition (MSQBE) for WRSNs. Assuming that all sensors are solar-powered with adjustable sensing radii, the proposed MSQBE applies the PSM over the BSM due to its greater practicality. To address the first challenge, as well as the variable and weather-dependent nature of solar energy in the second challenge, the MSQBE employs a 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, aiming to achieve an accurate estimation of the next day's PV power. Consequently, the energy allocated for each sensor, which is consumed during daytime and nighttime, is calculated based on the evaluation of acquired energy in the following day and their respective time lengths, which guarantee the balance of acquired and consumed energy in both daytime and nighttime.

To address the remaining part of the second challenge, which arises from the variability of PV power throughout the day, inappropriate sensor scheduling can lead to problems, including insufficient solar energy utilization and node failure due to excessive power consumption rate. The MSQBE partitions the time into several identical cycles, each consisting of a fixed number of identical time slots. Based on the average PV power and the length of each cycle, the MSQBE allocates a specific amount of energy for each sensor per cycle. The task schedule designed for each sensor must ensure that the consumed energy equals the allocated energy per cycle.

To address the third challenge, the MSQBE meticulously determines the length of each time slot by considering the available energy for each sensor per cycle and its maximum power consumption rate. It then uses the known relationship between power consumption rate and sensing radius to link active time with sensing radius, translating active time into the number of time slots per cycle. Under this scheme, the task schedule for each sensor only focuses on enhancing surveillance quality, as the balance between acquired and consumed energy is already ensured. This approach significantly simplifies the energy allocation and the design of task schedules for each sensor.

The main contributions of the proposed MSQBE are itemized as follows.

1) *Considering the Operations of Solar-Powered Sensors During Nighttime:* It is noticed that the solar-powered sensor cannot be charged during nighttime, and the surveillance quality during daytime and nighttime should be consistent. The proposed MSQBE algorithm designs the task schedule for each sensor and calculates the required energy to support sensor operation during nighttime, ensuring that the surveillance quality during the daytime and nighttime is equivalent and maximized.
2) *Considering the Variation of PV Power:* This article notes that PV power is variable and affected by weather conditions. To enhance the precision of PV power evaluation, the proposed MSQBE algorithm, utilizing weather forecast data, identifies the previous year's day that closely resembles the next day's weather and estimates the subsequent day's PV power using the predetermined PV power function for that day.
3) *Improving the Utilization of Solar Power:* Based on the evaluation of PV power on the next day, the potential energy charged from solar power on the next day can be obtained. Therefore, the proposed MSQBE algorithm can accurately allocate energy for sensor operation during both daytime and nighttime, mitigating concerns about sensors depleting more energy than they acquire, which could lead to sensors being dead. This precision in energy management prevents an overly conservative approach to power usage, thereby enhancing the utilization rate of solar energy.
4) *Enhancing Surveillance Quality Through Sensors With Adjustable Sensing Radius While Perpetuating Network Lifetime:* Given that sensors are often randomly deployed around the monitored target, sensor density varies across different areas. Areas with high deployment density may have redundant detection capabilities. The proposed MSQBE aims to schedule each sensor to select the best sensing radius such that the detection capacity of all the sensors can be fully utilized to maximize the surveillance quality. Furthermore, by evaluating the potential energy obtained from solar power, the MSQBE meticulously schedules the activation and charging of each sensor in sequence. This approach is designed to dynamically balance the energy acquired and consumed by each sensor, thereby perpetuating the network lifetime.

The remainder of this article is organized as follows. Section II reviews the existing related work. Section III presents the network environment and problem statement of the investigated issue. Section IV describes the proposed MSQBE in detail. Section V gives the performance evaluations of MSQBE. Finally, a conclusion and future work of this article is drawn in Section VI.

II. Related Work

In recent years, the target coverage problem in WSNs has received much attention. Plenty of target coverage algorithms have been proposed to enhance the surveillance quality and extend the network lifetime. These studies can be classified into four classes, including WSNs with fixed sensing radius sensors, WSNs with adjustable sensing radius sensors, WSNs with battery-powered sensor, and WSNs with rechargeable sensor. The following reviews these studies and compares them with our work.

A. WSNs With Fixed Sensing Radius Sensors

Because the fixed sensing radius sensor is comparatively easier to analyze, some studies applied the fixed sensing radius sensor to construct the target coverage. Gao et al. [9] addressed the continuous target coverage issue in mobile sensor networks, focusing on reducing sensor movement within a Euclidean surveillance space. It defined this challenge as the k-sink minimum movement target coverage problem and introduced a polynomial-time approximation scheme, the energy effective movement algorithm (EEMA) to solve it. To obtain a better performance of target coverage, Zhou et al. [11] proposed a flexible approach for target k-coverage in WRSNs, permitting temporary energy depletion of some sensors to reduce the cost of mobile chargers. It established a theoretical model to assess this strategy's impact and utilized a distributed algorithm for creating load-balanced sensor clusters. Then, it proposed λ-GTSP charging algorithm optimized sensor charging in clusters to maintain k-coverage, extending its application to include mobile targets. However, the abovementioned studies both adopted the BSM. To obtain a more practicable algorithm, Xu et al. [12] applied PSM and investigated the connected target coverage (CTC) problem by utilizing directional probabilistic sensors with an exponential attenuation model to efficiently monitor targets. It proved that this minimum energy coverage problem is NP-hard and addressed it by converting it to a network flow problem. It proposed that the MWMFA algorithm offers a solution with validated time complexity and approximation ratio. However, directional sensors, such as cameras, represent a specialized category and are not universally prevalent. Si et al. [13] investigated the efficient utilization of mobile sensors with the PSM to enhance coverage. Initially, it defined the concept of a "safe cell" and developed algorithms to identify safe cells and barrier gaps. Subsequently, it devised an algorithm aimed at optimizing barrier coverage using the fewest mobile sensors. In the final step, the Kuhn–Munkres (KM) algorithm was employed to address the problem of minimizing mobile sensor movement. While developing algorithms for sensors with a fixed sensing radius is relatively straightforward, such algorithms lack flexibility due to the fixed sensing radius constraint.

B. WSNs With Adjustable Sensing Radius Sensors

Considering the differences in sensor distribution density, some studies applied sensors with adjustable sensing radii to further improve the performance of constructed target coverage by enhancing the detection capability utilization of each sensor. Chen et al. [10] addressed the target coverage problem by considering the range-adjustable sensors. Based on the BSM, this study proposed a neighborhood-based estimation of distribution algorithm (NEDA) to optimize network lifetime by evolving coverage schemes where sensors are selectively activated. A linear programming (LP) model assigned activation times to these schemes, maximizing network lifetime based on individual fitness derived from LP. NEDA iteratively refines coverage schemes and solves the LP model, incorporating neighborhood sampling for diversity and a heuristic repair strategy for enhancing search efficiency. However, the target coverage algorithm proposed in this study still applies the BSM. Based on PSM, Fan et al. [4] investigated the barrier 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 barriers, determining their optimal working directions and sensing ranges. This scheme facilitates the formation of additional barriers, thereby extending the service lifetime of the network. However, directional sensors are a specific type of sensor that can only detect a fan-shaped area in a certain direction. Considering more common types of sensors, Zhang et al. [14] proposed a target coverage mechanism, called target coverage mechanism for sensors with adjustable sensing range (TCSAR), designed to optimize the surveillance quality of POIs while ensuring perpetual network life. TCSAR, employing the PSM, allows adjustable sensing radii for each sensor. It operates in two phases: a scheduling phase, where sensors are allocated based on their surveillance contribution, particularly focusing on bottleneck POIs, and a space–time transformation phase, which enhances POI surveillance by dynamically adjusting sensor radii across spatial and temporal dimensions.

C. WSNs With Battery-Powered Sensors

For the WSNs with batter-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. Keshmiri and Bakhshi [15] presented a two-phase optimization approach to address energy-efficient CTC. Initially, sensors were grouped into disjoint cover sets (CSs) via a multiobjective ILP model, with the remaining nodes serving as potential cluster heads (CHs). Then, CSs were activated in sequence to relay target data using a refined MILP model, optimized with the branch-and-bound method. Our method outperformed two key existing strategies, as validated by extensive experiments. Nguyen et al. [16] addressed the unique challenge of maximizing network lifetime while ensuring full target coverage and bounded relay hop connectivity. We introduced an exact LP formulation to find the optimal solution, and to mitigate time complexity, we developed a $(1+\varepsilon)^2$-approximation algorithm leveraging partitioning and shifting techniques. In addition, we proposed a scalable approximate LP formulation, where the variable size depends polynomially on the number of targets and sensors and its performance ratio is defined by M, and the maximum number of targets a sensor can cover. With a sufficient number of sensors, our algorithms could extend network lifetime by up to 3.68 times compared to existing methods and significantly reduce time complexity, maintaining it under 20% of that of benchmark approaches. However, these studies adopted the battery-powered sensor and the energy constraint still limits the performance of WSNs.

D. WSNs With Rechargeable Sensors

To overcome the energy limitation of WSNs, some studies applied the recharged sensor. Shang et al. [17] introduced a reinforcement learning method named reinforcement learning recharging (RLR) for a mobile charger in WSNs to optimize energy usage, recharging costs, and coverage benefits. RLR comprised three modules: sensor energy management (SEM), charger location update (CLP), and charger reinforcement learning (CRL). SEM allows distributed energy management by sensors, CLP ensures effective charger-sensor communication, and CRL uses rewards and penalties to guide charger actions for better coverage. Through learning from experiences, the charger improved its decision-making on when and where to charge, enhancing WSN coverage efficiency. Xue et al. [18] transformed the coverage of target maximization on-demand charging scheduling problem into a multiobjective optimization problem, aiming to simultaneously enhance the average coverage and energy efficiency. This study proved that the problem is NP-hard by reformulating it as a multiple traveling salesman problem with deadline. Then, it proposed the MaxCov scheme to optimize charger scheduling to enhance coverage, while the MaxCov-RG scheme, which groups requests, balances performance and computational complexity. However, the large-scale deployment of mobile chargers leads to a very high cost, and the charging path scheduling of mobile chargers is complex. Zhang et al. [14] applied the solar-powered sensors with adjustable sensing radius to construct the target coverage algorithm, called TCSAR, which scheduled the sensor to be activated in a specific sequence. The TCSAR involved two phases: the scheduling phase selected sensors based on their surveillance contribution to critical POIs and the space–time transformation phase enhanced the surveillance quality of POIs by reducing the sensing radii of some sensors. Li et al. [19] proposed solutions that maximize the data uploaded by radio frequency energy harvesting devices that are tasked with monitoring targets. It proposed a mixed integer linear program to size the charging time of hybrid access points and the sampling and data upload times of devices. It also proposed two heuristic algorithms to generate set covers. The results showed that the amount of data collected increases when there are more devices or deployed hybrid access points. Moreover, the distance-based algorithm achieved almost 97% of the optimal result.

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

III. Assumption and Problem Statement

This section initially introduces the network environment and assumptions of the given WSNs. Then, the problem statements and constraints are described.

TABLE I
COMPARISONS OF THE MAIN CHARACTERISTICS OF THE PROPOSED MSQBE WITH THE EXISTING RELATED WORK

Related Work	PSM	Adjustable Sensing Radius Sensor	Rechargeable Sensor	Considering the impact of meteorological scenarios on *PV* power	Maximizing the surveillance quality
[4]	○	○	×	×	×
[9]	×	×	×	×	×
[10]	○	○	×	×	×
[11]	×	×	○	×	×
[12]	○	×	×	×	×
[13]	○	×	×	×	×
[14]	○	○	○	×	○
[15]	×	×	×	×	×
[16]	×	×	×	×	×
[17]	×	×	○	×	×
[18]	×	×	○	×	×
[19]	×	×	○	×	○
MSQBE	○	○	○	○	○

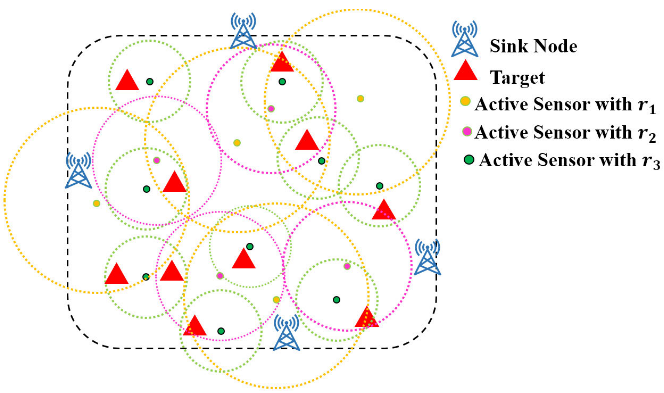


Fig. 1. Example of considered WRSNs.

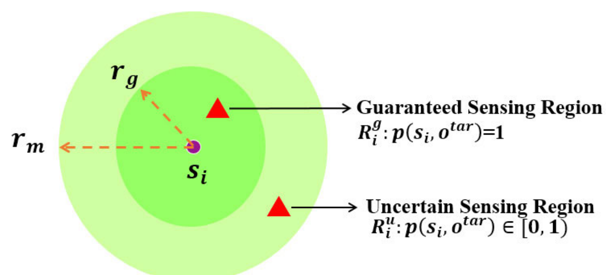


Fig. 2. Logic view of PSM.

A. Network Environment

This article considers that there are n monitored targets, also known as POIs that are randomly distributed in the 2-D region R, denoted by $O^{\text{tar}} = \{o_1^{\text{tar}}, o_2^{\text{tar}}, \ldots, o_m^{\text{tar}}\}$. There is a set of solar-powered sensors $S = \{s_1, s_2, \ldots, s_n\}$ randomly deployed in R, and each of them has a set of available sensing radii, which is a finite set of discrete values defined as $\mathcal{R} = \{r_1, r_2, \ldots, r_m\}$; without loss of generality, satisfy $r_i < r_{i+1} (1 \le i < m)$, and Fig. 1 gives an example of considered solar-powered WSNs. As shown in Fig. 1, each red triangle represents a POI that is the target that needs to be monitored. Each node and the corresponding circle represent an active sensor, where the green one, the purple one, and the yellow one indicate the active sensor with a sensing radius r_1–r_3, respectively. The signal tower represents the sink node.

B. Sensing Model

In this article, the PSM is applied. In general, the target will be detected by the sensor with a higher probability if it is closer to the sensor. Fig. 2 illustrates the main features of PSM.

As shown in Fig. 2, the sensing range of s_i consists of two subregions, the inner one, which is the guaranteed sensing area region represented by R_i^u, and the outer one, which is the uncertain sensing region represented by R_i^g. The sensing radii of R_i^u and R_i^g are r_g and r_m, respectively. The target within R_i^g is considered to be guaranteed detected by the sensor s_i with the detection probability of s_i for this target is 1. The target within R_i^u is considered to be uncertainly detected by the sensor s_i with the detection probability of s_i for this target ranging from 0 to 1, and the detection probability of s_i for this target decreases with increasing distance from the sensor s_i.

Let $p(s_i, o^{\text{tar}})$ denote the detection probability of s_i to the target o^{tar}. Let $d(s_i, o^{\text{tar}})$ denote the distance between s_i and o^{tar}. The value of $p(s_i, o^{\text{tar}})$ can be calculated by the following expression:

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

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

C. Charging and Discharging Model

In the given WSNs, assuming that each sensor is solar-powered and mainly has four possible states. There are two possible during daytime: charging and sensing state, and charging-only state; there are two possible during nighttime:

sensing-only state and sleeping state. A sensor consumes energy if it stays in charging and sensing or sensing-only state, which is considered an active sensor. A sensor can be charged if it stays in charging and sensing or charging-only state. It is noticed that a sensor performs sensing and charging operations simultaneously if it stays in the sensing and charging state. A sensor does not perform any operation if it stays in a sleeping state.

Let E denote the battery capacity, which represents the maximum available energy of each sensor. Let e_k^{con} denote the power consumption rate of each sensor when it selects the sensing radius $r_k \in \mathcal{R}$. The power consumption rate corresponding to each sensing radius can be calculated according to the quadratic model [20] as

$$e_k^{\text{con}} = e_m^{\text{con}} \times (r_k/r_m)^2 \tag{2}$$

where e_m^{con} is the power consumption rate of the maximum sensing radius r_m, which is given in advance.

Recall that the proposed MSQBE partitions time into several identical time slots, enabling it to not only evaluate and enhance the surveillance quality in the spatial dimension but also in the temporal dimension. Then, a fixed number of time slots constitute a cycle, denoted by T_c. According to (2), the power consumption rate of each sensor is maximum when this sensor selects the maximum sensing radius r_m. As the energy allocated to each cycle is identical, the active time of each sensor is minimized when it selects the sensing radius r_m. Here, the term "active time" refers to the longest duration within a cycle that the allocated energy can support the sensor in performing sensing operations. It is assumed that the energy allocated for each cycle is only sufficient to sustain the sensor's activation with the maximum sensing radius r_m for one time slot per cycle. This time slot, denoted by τ, serves as the basic time unit in each cycle for task scheduling. The discussion regarding the lengths of one time slot and one cycle will be presented 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 expression:

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

Let t_k^{act} denote the active time when each sensor selects the sensing radius r_k, which can be calculated according to the following expression:

$$t_k^{\text{act}} = \frac{E_{\text{day}}^{\text{con}}}{e_k^{\text{con}}} = \frac{E_{\text{day}}^{\text{con}}}{e_m^{\text{con}} \times (r_k/r_m)^2}$$

where $E_{\text{day}}^{\text{con}}$ represents the energy allocated for each sensor consumed during the daytime. Since sensors can be continuously charged during the daytime, the acquired and consumed energy can remain balanced in the daytime if this equilibrium can be achieved within every cycle. Therefore, the energy allocated for each sensor consumed in each cycle is $E_{\text{day}}^{\text{con}}$.

As mentioned above, the energy for each cycle is only sufficient to sustain the sensor's activation with the maximum sensing radius r_m for one time slot per cycle, that is, $t_m^{\text{act}} = \tau$.

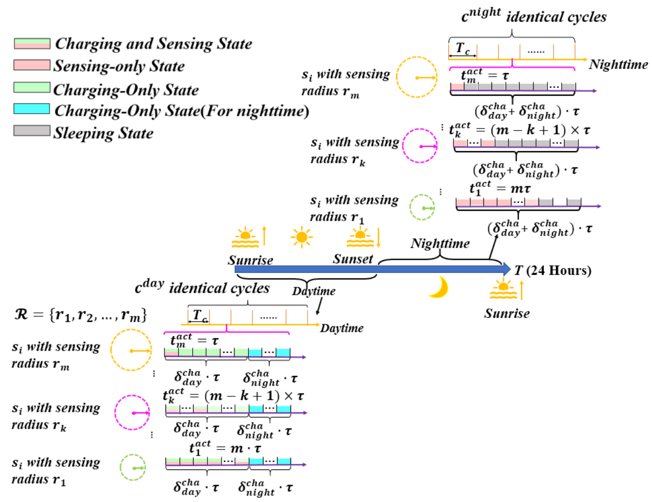


Fig. 3. Example of charging and discharging model.

Therefore, we have

$$t_k^{\text{act}} = (m-k+1) \times \tau. \tag{4}$$

According to (4), each sensor with a sensing radius r_k can be activated $(m-k+1)$ time slots.

Fig. 3 illustrates the relationship between active time and sensing radius. The active time t_1^{act} of s_i with sensing radius r_1 is $m \cdot \tau$. The active time t_m^{act} of s_i with sensing radius r_m is τ. The active time t_k^{act} of s_i with any random sensing radius r_k is $(m-k+1) \times \tau$, where $r_k \in \mathcal{R}$.

D. Problem Statement

This article addresses the target coverage problem in solar-powered WSNs that adopt sensors with adjustable sensing radii. The proposed MSQBE aims to construct a target coverage to maximize the surveillance quality while balancing energy consumption and acquisition, thereby perpetuating the network lifetime. The objective function and some constraints are presented as follows.

Let $\mathbb{s}$ denote a possible scheduling algorithm to form a target coverage algorithm. The set of all the possible scheduling algorithms is denoted by $\mathbb{S}$. The following will discuss the surveillance quality under the scheduling algorithm $\mathbb{s}$. Let $S_{v,h}^{\text{act}}(\mathbb{s})$ denote the set of active sensors at time point t_h that, by applying algorithm s, can cover the target $o_v^{\text{tar}} \in O^{\text{tar}}$. Consider a space–time point (o_v^{tar}, t_h), where $o_v^{\text{tar}} \in O^{\text{tar}}$ and $t_h \in T_c$. Let $p_{v,h}^{\mathbb{S}}$ denote the cooperative detection probability of all the active sensors in $S_{v,h}^{\text{act}}(\mathbb{s})$ to space–time point (o_v^{tar}, t_h), which can be calculated by applying the following expression:

$$p_{v,h}^{\mathbb{S}} = 1 - \prod_{s_i \in S_{v,h}^{\text{act}}(\mathbb{S})} \left(1 - p\left(s_i, o_v^{\text{tar}}\right)\right) \tag{5}$$

where $p(s_i, o_v^{\text{tar}})$ is the detection probability of s_i to o_v^{tar}, which can be obtained by applying (1).

Let $q(\mathbb{s}, o_v^{\text{tar}})$ denote the surveillance quality for o_v^{tar} achieved by applying algorithm $\mathbb{s}$, which is represented by the weakest cooperative detection probability of all active sensors to o_v^{tar}

at any time point, that is,

$$q(\mathbb{s}, o_v^{\text{tar}}) = \min_{t_h \in T_c} p_{v,h}^{\mathbb{S}}.$$

Let $Q(\mathbb{s})$ denote the surveillance quality for O^{tar} achieved by algorithm $\mathbb{s}$. Because there are multiple monitored targets in O^{tar}, to guarantee that the surveillance quality can be maximized to any target at any time point, in this article, $Q(\mathbb{s})$ is represented by the weakest surveillance quality among all active sensors for any target at any time point, that is,

$$Q(\mathbb{s}) = \min_{o_v^{tar} \in O^{tar}} q\left(\mathbb{s}, o_v^{\text{tar}}\right).$$

However, as the number of time points is infinite. This article divides time into several identical time slots, with a fixed number of time slots constituting one cycle. The task schedule of each sensor within one time slot remains identical, and therefore, the surveillance quality is considered uniform with each time slot. The detail of this approach is presented in Section IV.

The proposed MSQBE aims to construct a target coverage to maximize the surveillance quality while balancing energy consumption and acquisition. The object function of this article is given in (6).

Objective Function:

$$\max(Q(\mathbb{s})). \tag{6}$$

Equation (6) should be achieved under the condition that some constraints should be further satisfied. The following presents these constraints.

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

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

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

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

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

First, the state constraint ensures that any sensor s_i can only stay in one of the four previously mentioned states at any time.

1) State Constraint:

$$\lambda_{i,h}^{\text{cha-sen}} + \lambda_{i,h}^{\text{cha-only}} + \lambda_{i,h}^{\text{sen-only}} + \lambda_{i,h}^{\text{slp}} = 1. \tag{7}$$

Second, the following cycle-working constraint guarantees that each sensor will be activated at least one time slot in each cycle. This constraint aims to ensure that all the deployed sensors will participate in the monitoring task.

2) Cycle-Working Constraint:

$$\sum_{h=1}^{\left\lceil \frac{T_c}{\tau} \right\rceil} \left(\lambda_{i,h}^{\text{cha-sen}} + \lambda_{i,h}^{\text{sen-only}}\right) \geq 1 \quad \forall s_i. \tag{8}$$

Let $e^{\text{PV}}(t)$ denote the function of PV power function. The energy that a sensor can obtain from solar power in one day is $\int_{t_s}^{t_e} e^{\text{PV}}(t)dt$, where t_s and t_e are the start time and end time of daytime, respectively. Each sensor can be charged during the daytime. Finally, the energy balance constraint ensures that the energy acquired from solar power balances the energy consumed for sensing operations by each sensor over the entire day, thereby perpetuating the network lifetime.

3) Energy Balance Constraint:

$$\int_{t_s}^{t_e} e^{\text{PV}}(t)dt = \sum_{j=1}^{\left\lceil \frac{T^{\text{day}}}{T_c} \right\rceil} \left(e_{k,j}^{\text{con}} \cdot t_{k,j}^{\text{act}}\right) \tag{9}$$

where $e_{k,j}^{\text{con}}$ represents the power consumption rate of the sensor s_i, $t_{k,j}^{\text{act}}$ represents the active time of s_i in the jth cycle, and T^{day} represents the length of daytime.

Section IV presents the proposed MSQBE algorithm, which aims to achieve the objective function shown in (6) while satisfying the constraints given in (7)–(9).

IV. Proposed MSQBE Algorithm

This article proposed a novel target coverage algorithm, called MSQBE for WRSNs. The MSQBE adopts solar-powered sensors with adjustable sensing radius. Its objective is to efficiently schedule the sensing radius and active time of each sensor such that the surveillance quality for all the targets is O^{tar} while balancing energy consumption and acquisition.

The proposed MSQBE mainly consists of three phases, including the *Time dividing phase*, *detection probability calculation phase*, and *sensor scheduling phase*. The first phase initially employs 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, which is already known, aiming to achieve an estimation of the next day's PV power. Subsequently, the MSQBE divides the continuous time into several equal-length time slots, and a fixed number of time slots further constitutes one cycle, where the length of each time slot and the cycle is based on the estimation of the next day's PV power, which aims to balance the energy consumption and acquisition. These time slots and targets constitute a series of space–time points, allowing for convenient evaluation of surveillance quality for each target at each time slot. The second phase is to calculate the detection probability of each sensor to each target, which is considered as the surveillance contribution of each sensor. The last phase initially identifies the bottleneck space–time point that has the weakest detection probability, and then, the MSQBE designs the task schedule with the largest contribution to the bottleneck space–time point for each sensor in a distributed manner, aiming to maximize the surveillance quality for O^{tar}.

A. Time Dividing Phase

This phase aims to divide continuous time into several time slots of equal length, organized into cycles of equal length. The sensor scheduling scheme operates on a cycle-based basis, with the task schedule of each sensor within each time slot being identical. Consequently, the surveillance quality within each time slot is also uniform. The length of each time slot and cycle is determined by evaluating the potential for energy acquisition and the rate of energy consumption, where the potential for energy acquisition can be calculated by the PV power of that day. 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 involves calculating the length of one time slot and one cycle for both daytime and nighttime, grounded in an assessment of the PV power and the energy consumption rate of each sensor, aiming to achieve a balance between acquired energy and consumed energy.

1) Step 1 (PV Power Evaluation Based on Similarity Calculation): It is noticed that the total amount of energy obtained from solar power over a certain period of time is the integral of the PV power during that time period [21]. The length of each time slot and cycle should be determined in advance, as the algorithm requires future PV power data. While many existing PV power forecasting methods are based on deep learning algorithms, these require high-performance computing power. It is impractical for WSNs, as the majority of devices within WSNs are edge devices with limited computing power. Even if the PV power forecasting operation is executed at the base station, there is a significant challenge in transmitting the forecasting results to all sensor nodes. The PV power forecasting method based on deep learning algorithms is alternative good approach. Nonetheless, it necessitates frequent data transmissions to the sensors to ensure the dissemination of the most recent prediction results. The exact transmission frequency and data volume are contingent upon the desired level of accuracy.

To estimate the PV power within an acceptable margin of error without consuming excessive resources, the MSQBE algorithm first stores the historical daily PV power data and corresponding meteorological data of the previous year in the base station and then obtains the meteorological data of the next day through the network. The main meteorological factors affecting PV power generation include horizontal radiation, land surface temperature, surface wind speed, relative humidity, and total cloud cover [22]. To store both historical and forecast meteorological data, some matrices are constructed. These matrices serve as a structured framework, allowing for the organized storage and efficient retrieval of data points corresponding to various meteorological parameters over different time intervals. Assume that there are n observation time points each day to record the meteorological data at those times. Let m_1, m_2, m_3, m_4, and m_5 be the labels of horizontal radiation, land surface temperature, surface wind speed, relative humidity, and total cloud cover, respectively. Let $\mathcal{M}^{\text{met-hist}}_{5\times n,i}$ denote the historical daily meteorological data matrix for the ith day of the year, that is,

$$\mathcal{M}^{\text{met-hist}}_{5\times n,i} = \begin{bmatrix} \text{data}^{i\text{-hist}}_{m_1,f_1} & \cdots & \text{data}^{i\text{-hist}}_{m_1,f_n} \\ \vdots & \ddots & \vdots \\ \text{data}^{i\text{-hist}}_{m_5,f_1} & \cdots & \text{data}^{i\text{-hist}}_{m_5,f_n} \end{bmatrix} \tag{10}$$

where each element $\text{data}^{i\text{-hist}}_{m_l,f_k}$ in $\mathcal{M}^{\text{met-hist}}_{5\times n,i}$ is the corresponding meteorological data recorded at the time f_k in ith day, where $1 \le k \le n$ and $1 \le i \le 365$.

Let $\mathcal{M}^{\text{met-fore}}_{5\times n,j}$ denote the forecast meteorological data matrix for the following day, which corresponds to the jth day, that is,

$$\mathcal{M}^{\text{met-fore}}_{5\times n,j} = \begin{bmatrix} \text{data}^{j\text{-fore}}_{m_1,f_1} & \cdots & \text{data}^{j\text{-fore}}_{m_1,f_n} \\ \vdots & \ddots & \vdots \\ \text{data}^{j\text{-fore}}_{m_5,f_1} & \cdots & \text{data}^{j\text{-fore}}_{m_5,f_n} \end{bmatrix}. \tag{11}$$

It is noticed that there are 365 historical daily meteorological data matrices. Let $\mathbb{M}^{\text{met-hist}}$ denote the set of the historical daily meteorological data matrix, that is,

$$\mathbb{M}^{\text{met-hist}} = \left\{\mathcal{M}^{\text{met-hist}}_{5\times n,1}, \mathcal{M}^{\text{met-hist}}_{5\times n,2}, \ldots, \mathcal{M}^{\text{met-hist}}_{5\times n,365}\right\}.$$

Then, the algorithm aims to identify the matrix from $\mathbb{M}^{\text{met-hist}}$ that has the smallest degree of difference from $\mathcal{M}^{\text{met-fore}}_{5\times n,j}$. This identified matrix is referred to as the template meteorological data matrix of the following day. To achieve this, the Frobenius norm, or F-norm in short, is employed to evaluate the similarity between matrices [23]. For any matrix $\mathcal{M}^{\text{met-hist}}_{5\times n,i} \in \mathbb{M}^{\text{met-hist}}$, the algorithm initially calculates the difference matrix between $\mathcal{M}^{\text{met-hist}}_{5\times n,i}$ and $\mathcal{M}^{\text{met-fore}}_{5\times n,j}$, denoted by $\mathcal{M}^{\text{diff}}_{5\times n}(i, j)$, according to the following expression:

$$\begin{aligned} &\mathcal{M}^{\text{diff}}_{5\times n}(i, j) \\ &\quad = \mathcal{M}^{\text{met-hist}}_{5\times n,i} - \mathcal{M}^{\text{met-fore}}_{5\times n,j} \\ &\quad = \begin{bmatrix} \text{data}^{i\text{-hist}}_{m_1,f_1} - \text{data}^{j\text{-fore}}_{m_1,f_1} & \cdots & \text{data}^{i\text{-hist}}_{m_1,f_n} - \text{data}^{j\text{-fore}}_{m_1,f_n} \\ \vdots & \ddots & \vdots \\ \text{data}^{i\text{-hist}}_{m_5,f_1} - \text{data}^{j\text{-fore}}_{m_5,f_1} & \cdots & \text{data}^{i\text{-hist}}_{m_5,f_n} - \text{data}^{j\text{-fore}}_{m_5,f_n} \end{bmatrix}. \end{aligned} \tag{12}$$

Subsequently, the F-norm of each $\mathcal{M}^{\text{diff}}_{5\times n}(i, j)$ is

$$\left\|\mathcal{M}^{\text{diff}}_{5\times n}(i, j)\right\|_F = \sqrt{\sum_{l=1}^{5}\sum_{k=1}^{n}\left|\text{data}^{i\text{-hist}}_{m_l,f_k} - \text{data}^{j\text{-fore}}_{m_l,f_k}\right|^2}. \tag{13}$$

Let $\mathcal{M}^{\text{tem}}_{5\times n}$ denote the identified template meteorological data matrix from $\mathbb{M}^{\text{met-hist}}$ for the following day, which is:

$$\mathcal{M}^{\text{tem}}_{5\times n} = \arg \min_{\mathcal{M}^{\text{met-hist}}_{5\times n,i} \in \mathbb{M}^{\text{met-hist}}} \left\|\mathcal{M}^{\text{diff}}_{5\times n}(i, j)\right\|_F.$$

Therefore, the PV power data for the following day are represented by $\mathcal{M}^{\text{tem}}_{5\times n}$, which is given in advance. Based on the $\mathcal{M}^{\text{tem}}_{5\times n}$, the template PV power function on the following day, corresponding to the jth day and denoted by $e^{\text{PV}}_{j\text{-tem}}$, can be obtained by applying the least-squares method. The least-squares method is well known and can be omitted here.

2) Step 2 (Calculating the Length of One Time Slot and Cycle): After obtaining the estimated PV power function of the following day, this step, while taking into account the sensor's operations during both day and night, calculates the length of each time slot and cycle to maintain a balance between the energy consumption and acquisition of each sensor.

Let t_s^{day} and t_e^{day} denote the starting time and the ending time of the daytime, respectively. The total amount of energy that might be obtained on the following day, denoted by $E_{\text{cha}}^{\text{total}}$, can be evaluated by the following expression:

$$E_{\text{cha}}^{\text{total}} = \int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}(t)dt. \tag{14}$$

Since sensors cannot be charged via solar energy during nighttime, it is necessary to reserve sufficient power in advance for the sensors to consume during the nighttime. Let $E_{\text{day}}^{\text{con}}$ and $E_{\text{night}}^{\text{con}}$ denote the energy allocated for each sensor consumed during daytime and nighttime, respectively. To satisfy the energy balance constraint, which is shown in (9), we have

$$E_{\text{day}}^{\text{con}} + E_{\text{night}}^{\text{con}} = \int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}(t)dt. \tag{15}$$

Recall the charging and discharging model; the energy allocated for each sensor consumed in each cycle is $E_{\text{day}}^{\text{con}}$; and the length of one time slot, denoted by τ, corresponds to the active time of the sensor with the maximum sensing radius r_m in one cycle. Therefore, the length of one time slot is

$$\tau = \frac{E_{\text{day}}^{\text{con}}}{e_m^{\text{con}}} \tag{16}$$

where e_m^{con} is the power consumption rate corresponding to the maximum sensing radius r_m.

Next, the length of each cycle is discussed in detail. As shown in Fig. 3, $\delta_{\text{day}}^{\text{cha}}$ and $\delta_{\text{night}}^{\text{cha}}$ are the numbers of time slots that are used for charging the energy consumed during the daytime and nighttime, respectively. Each cycle consists of $(\delta_{\text{day}}^{\text{cha}} + \delta_{\text{night}}^{\text{cha}})$ time slots. Therefore, determining the length of each cycle is to determine the value of $\delta_{\text{day}}^{\text{cha}}$ and $\delta_{\text{night}}^{\text{cha}}$. To satisfy the energy balance constraint, we have

$$\left(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}dt\right) \cdot \left(t_e^{\text{day}} - t_s^{\text{day}}\right)^{-1} \delta_{\text{day}}^{\text{cha}} \cdot \tau = e_m^{\text{con}} \cdot \tau \tag{17}$$

where the term $(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}(t)dt)(t_e^{\text{day}} - t_s^{\text{day}})^{-1}$ is the average charging rate of PV power. Equation (17) guarantees that the acquired and consumed energy is balanced in each cycle. According to (17), the value of $\delta_{\text{day}}^{\text{cha}}$ can be calculated, that is,

$$\delta_{\text{day}}^{\text{cha}} = \left[e_m^{\text{con}}\left(t_e^{\text{day}} - t_s^{\text{day}}\right)\right] \cdot \left(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}(t)dt\right)^{-1} \tag{18}$$

Let $n_{\text{day}}^{\text{cyc}}$ denote the number of cycles in the daytime, that is,

$$n_{\text{day}}^{\text{cyc}} = \left(t_e^{\text{day}} - t_s^{\text{day}}\right) \cdot \left[\left(\delta_{\text{day}}^{\text{cha}} + \delta_{\text{nignt}}^{\text{cha}}\right) \cdot \tau\right]^{-1}.$$

Therefore, the total energy charged during the daytime for the sensor consumed in the nighttime is

$$E_{\text{night}}^{\text{con}} = n_{\text{day}}^{\text{cyc}} \cdot \delta_{\text{nignt}}^{\text{cha}} \cdot \tau \cdot \left(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}(t)dt\right)\left(t_e^{\text{day}} - t_s^{\text{day}}\right)^{-1}.$$

Let $n_{\text{night}}^{\text{cyc}}$ denote the number of cycles in the nighttime, that is,

$$n_{\text{night}}^{\text{cyc}} = \left(T - t_e^{\text{day}} + t_s^{\text{day}}\right) \cdot \left[\left(\delta_{\text{day}}^{\text{cha}} + \delta_{\text{nignt}}^{\text{cha}}\right) \cdot \tau\right]^{-1} \tag{19}$$

where T is the length of one day.

The total consumed energy of each sensor in the nighttime is $n_{\text{night}}^{\text{cyc}} \cdot e_m^{\text{con}} \cdot \tau$. To satisfy the energy balance constraint, we have

$$\begin{aligned} &n_{\text{night}}^{\text{cyc}} \cdot e_m^{\text{con}} \cdot \tau \\ &\quad = n_{\text{day}}^{\text{cyc}} \cdot \delta_{\text{nignt}}^{\text{cha}} \cdot \tau \cdot \left(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}(t)dt\right)\left(t_e^{\text{day}} - t_s^{\text{day}}\right)^{-1}. \end{aligned} \tag{20}$$

According to (20), the value of $\delta_{\text{night}}^{\text{cha}}$ can be obtained, that is,

$$\delta_{\text{night}}^{\text{cha}} = e_m^{\text{con}}\left[T - t_e^{\text{day}} + t_s^{\text{day}}\right] \cdot \left(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}dt\right)^{-1}. \tag{21}$$

Until now, the length of one cycle can be obtained, that is,

$$T_c = \left(T \cdot \tau \cdot e_m^{\text{con}}\right)\left(\int_{t_s^{\text{day}}}^{t_e^{\text{day}}} e_{j\text{-tem}}^{\text{PV}}dt\right)^{-1}. \tag{22}$$

The next phase further calculates the detection probability of each sensor to each target.

B. Detection Probability Calculation Phase

This phase aims to calculate the detection probability of each sensor to each target such that the surveillance quality can be obtained.

Recall the sensing model discussed in Section III, the detection probability of the sensor to the target depends on the distance between the sensor and the target. For any target $o_v^{\text{tar}} \in O^{\text{tar}}$, if the target o_v^{tar} is not covered by s_i, the detection probability of s_i to o_v^{tar} is zero.

If o_v^{tar} is covered by the sensor s_i, that is,

$$d\left(s_i, o_v^{\text{tar}}\right) \leq r_k$$

where $d(s_i, o_v^{\text{tar}})$ is the distance between s_i and o_v^{tar}, and r_k is the sensing radius of s_i, the detection probability of s_i to o_v^{tar}, denoted by $p(s_i, o_v^{\text{tar}})$, can be calculated by applying (1).

If there are more than one sensor can cover the target o_v^{tar}, let $S_v^{\text{cov}} = \{s_{v,1}^{\text{cov}}, s_{v,2}^{\text{cov}}, \ldots, s_{v,n}^{\text{cov}}\}$ denote the set of active sensors that can cover o_v^{tar}. Let $p^{\text{co-det}}(S_v^{\text{cov}}, o_v^{\text{tar}})$ denote the cooperative detection probability of all the sensors in S_v^{cov} to the target o_v^{tar}, which can be calculated by applying the following expression:

$$p^{\text{co-det}}\left(S_v^{\text{cov}}, o_v^{\text{tar}}\right) = 1 - \prod_{i=1}^{n}\left[1 - p\left(s_{v,i}^{\text{cov}}, o_v^{\text{tar}}\right)\right]. \tag{23}$$

Fig. 4 gives an example to illustrate how to calculate the detection probability of each sensor to different targets. As shown in Fig. 4, the target o_1^{tar} is not covered by any

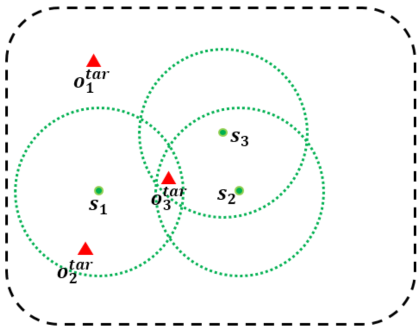


Fig. 4. Example of calculating the detection probability.

sensor, and therefore, we have $p(s_1, o_1^{\text{tar}}) = p(s_2, o_1^{\text{tar}}) = p(s_3, o_1^{\text{tar}}) = 0$.

The target o_2^{tar} is covered by the sensor s_1, and the detection probability of s_1 to o_2^{tar} depends on the distance between s_1 and o_2^{tar}. By applying (1), we have

$$p\left(s_1, o_2^{\text{tar}}\right) = \begin{cases} 1, & d\left(s_1, o_2^{\text{tar}}\right) \le r_g \\ e^{-\lambda\left(d\left(s_1, o_2^{\text{tar}}\right) - r_g\right)^{\gamma}}, & r_g < d\left(s_1, o_2^{\text{tar}}\right) \le r_k. \end{cases}$$

Furthermore, the target o_3^{tar} is covered by s_1, s_2, and s_3 simultaneously. The cooperative detection probability of s_1, s_2, and s_3 to o_3^{tar} is

$$p^{\text{co-det}}\left(S_3^{\text{cov}}, o_3^{\text{tar}}\right) = 1 - \prod_{i=1}^{3}\left[1 - p\left(s_{3,i}^{\text{cov}}, o_3^{\text{tar}}\right)\right]$$

where $S_3^{\text{cov}} = \{s_1, s_2, s_3\}$.

This phase calculates the detection probability of each sensor to each target, which is considered as the surveillance contribution of each sensor to each target. If the detection probability of a sensor to a target is greater than zero, then the sensor is considered to have a surveillance contribution to that target; otherwise, it is considered to have no surveillance contribution. The next phase aims to first identify the bottleneck space–time point that has the weakest detection probability and then schedule the sensor with the largest contribution to the bottleneck space–time point, aiming to maximize the surveillance quality for O^{tar}.

C. Sensor Scheduling Phase

This phase aims to schedule all the sensors to be activated at different sensing radii and active times for monitoring all the targets, with the goal of maximizing surveillance quality. To achieve this, this phase mainly consists of two steps. The first step is to identify the bottleneck space–time point that has the weakest cooperative detection probability based on the current schedule of each sensor. The second step is to schedule the sensor with the maximum surveillance contribution to monitor the bottleneck space–time point, aiming to maximize the surveillance quality of the target coverage.

1) Step 1 (Identifying the Bottleneck Space–Time Point): Recall that the continuous time is divided into several time slots of equal length, and the task schedule of each sensor is identical. Each time slot and target can form multiple space–time points. Each space–time point, represented by (o_v^{tar}, t_h), where $o_v^{\text{tar}} \in O^{\text{tar}}$ and $t_h \in T_c$, is a 2-D data structure. Based on the concept of space–time point, the detection probability of all the active sensors to target o_v^{tar} at time slot t_h can be easily represented by the detection probability to space–time point (o_v^{tar}, t_h).

This step aims to identify the bottleneck space–time point that has the weakest detection probability. Because the proposed MSQBE schedules all the sensors in a distributed manner, each sensor only needs to focus on the local monitoring region (LMR) within its sensing range. Let LMR_i denote the LMR of s_i, which can be expressed by the following expression:

$$\text{LMR}_i = \left\{o_v^{\text{tar}} \,\middle|\, d\left(s_i, o_v^{\text{tar}}\right) \le r_m\right\}.$$

Let LMR_i^{st} denote the local monitoring space–time region of s_i, which expands the LMR_i^{st} into two dimensions. The LMR_i^{st} only considers the target within LMR_i in the spatial dimension and the time slot in one cycle in the temporal dimension, that is,

$$\text{LMR}_i^{\text{st}} = \left\{\left(o_v^{\text{tar}}, t_h\right) \middle| o_v^{\text{tar}} \in \text{LMR}_i, t_h \in T_c\right\}.$$

Recall that $p_{v,h}$ denotes the cooperative detection probability of all the active sensors in $S_{v,h}^{\text{act}}$ to space–time point (o_v^{tar}, t_h), where $S_{v,h}^{\text{act}}$ is the set of sensors that can cover o_v^{tar} and is scheduled to be active at the time slot t_h. The value of $p_{v,h}$ can be obtained according to the following expression:

$$p_{v,h} = 1 - \sum_{s_j \in S_{v,h}^{\text{act}}} \left[1 - p\left(s_j, o_v^{\text{tar}}\right)\right]. \tag{24}$$

Let $\text{o}_{i,\text{loc}}^{\text{weak}}$ denote the local bottleneck space–time point in LMR_i^{st}, and $\text{o}_{i,\text{loc}}^{\text{weak}}$ can be identified according to the following expression:

$$\text{o}_{i,\text{loc}}^{\text{weak}} = \arg \min_{(o_v^{\text{var}}) \in \text{LMR}_i^{\text{st}}} p_{v,h}. \tag{25}$$

To calculate $p_{v,h}$ and $\text{o}_{i,\text{loc}}^{\text{weak}}$, two matrices include the local surveillance quality matrix $\mathcal{M}_{i,loc}^{\text{QoS}}$ and local current schedule matrix $\mathcal{M}_{i,\text{loc}}^{\text{act}}$, where $\mathcal{M}_{i,\text{loc}}^{\text{QoS}}$ stores the cooperative detection probability of active sensors to the space–time points in LMR_i^{st}, while $\mathcal{M}_{i,\text{loc}}^{\text{act}}$ stores the local current scheduling result of 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 the communication radius of each sensor, which is twice the maximum sensing radius. Let $\mathbb{N}(s_i)$ denote the set of neighboring sensors of s_i, and we have

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

Both $\mathcal{M}_{i,\text{loc}}^{\text{QoS}}$ and $\mathcal{M}_{i,\text{loc}}^{\text{act}}$ are maintained by s_i. Assume that there are m targets in LMR_i and each cycle consists of n time slots. The structure of $\mathcal{M}_{i,\text{loc}}^{\text{QoS}}$ is

$$\mathcal{M}_{i,\text{loc}}^{\text{QoS}} = \begin{pmatrix} p_{1,1} & \cdots & p_{1,n} \\ \vdots & \ddots & \vdots \\ p_{m,1} & \cdots & p_{m,n} \end{pmatrix}. \tag{26}$$

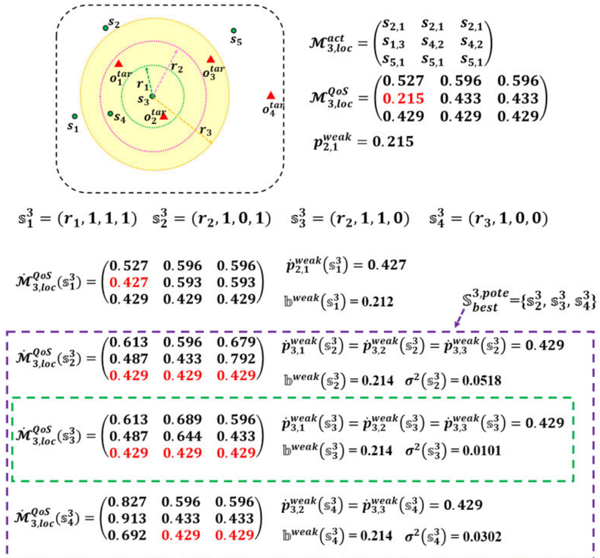


Fig. 5. Example of identifying the bottleneck space–time point and designing the task schedule.

The structure of $\mathcal{M}^{\text{act}}_{i,\text{loc}}$ is

$$\mathcal{M}^{\text{act}}_{i,\text{loc}} = \begin{pmatrix} S^{\text{act}}_{1,1} & \cdots & S^{\text{act}}_{1,n} \\ \vdots & \ddots & \vdots \\ S^{\text{act}}_{m,1} & \cdots & S^{\text{act}}_{m,n} \end{pmatrix}. \tag{27}$$

Each sensor will broadcast its scheduling result. By receiving the scheduling results from the sensors in $\mathbb{N}(s_i)$, $\mathcal{M}^{\text{act}}_{i,\text{loc}}$ can be initially obtained. According to $\mathcal{M}^{\text{act}}_{i,\text{loc}}$, the value of each element in $\mathcal{M}^{\text{QoS}}_{i,\text{loc}}$ can be calculated by applying (24), and the local bottleneck space–time point $\mathbb{o}^{\text{weak}}_{i,\text{loc}}$ can be identified according to (25). If there exists more than one space–time point that satisfies (25), let $\mathbb{O}^{\text{weak}}_{i,\text{loc}}$ denote the set of local bottleneck space–time points in LMR^{st}_i. The next step will schedule the sensor with maximum surveillance contribution to $\mathbb{o}^{\text{weak}}_{i,\text{loc}}$ in a distributed manner, aiming to maximize the surveillance quality for all targets.

Fig. 5 depicts an example to illustrate the operation of identifying the bottleneck space–time point. As shown in Fig. 5, a set of targets $O^{\text{tar}} = \{o^{\text{tar}}_1, o^{\text{tar}}_2, \ldots, o^{\text{tar}}_4\}$ are distributed in a monitoring region, a set of sensor $S = \{s_1, s_2, \ldots, s_5\}$ are randomly deployed around the targets. For an unscheduled sensor s_3, the set of available sensing radii is $\mathcal{R} = \{r_1, r_2, r_3\}$. The local monitoring space–time region of s_3 is

$$\text{LMR}^{\text{st}}_3 = \left\{\left(o^{\text{tar}}_v, t_h\right) \mid d\left(s_3, o^{\text{tar}}_v\right) \le r_3, t_h \in T_c\right\}.$$

The set of neighboring sensors of s_3 is $\mathbb{N}(s_3) = \{s_1, s_2, s_4, s_5\}$. Assume that each cycle T_c has three time slots, and the local current schedule matrix of s_3 is

$$\mathcal{M}^{\text{act}}_{3,\text{loc}} = \begin{pmatrix} s_{2,1} & s_{2,1} & s_{2,1} \\ s_{1,3} & s_{4,2} & s_{4,2} \\ s_{5,1} & s_{5,1} & s_{5,1} \end{pmatrix}$$

where each element $s_{i,k}$ in $\mathcal{M}^{\text{act}}_{3,\text{loc}}$ represents the sensor s_i with sensing radius r_k.

Based on $\mathcal{M}^{\text{act}}_{3,\text{loc}}$, the local surveillance quality matrix of s_3 can be obtained, that is,

$$\mathcal{M}^{\text{QoS}}_{3,\text{loc}} = \begin{pmatrix} 0.527 & 0.596 & 0.596 \\ 0.215 & 0.433 & 0.433 \\ 0.429 & 0.429 & 0.429 \end{pmatrix}.$$

According to (25), the local bottleneck space–time point $\mathbb{o}^{\text{weak}}_{3,\text{loc}}$ in LMR^{st}_3 is (o^{tar}_2, t_1), which is marked in red ink in $\mathcal{M}^{\text{QoS}}_{3,\text{loc}}$.

2) Step 2 (Designing the Task Schedule With the Maximum Contribution to Monitor the Bottleneck Space–Time Point): Recall that the detection probability of each sensor to each target is considered as the surveillance contribution to each target. If a sensor s_i can cover the target o^{tar}_v and is activated at the time slot t_h, then it is considered to have the surveillance quality to space–time point (o^{tar}_v, t_h). This step involves scheduling the sensor with maximum surveillance contribution to the local bottleneck space–time point, which is referred to as the best task schedule, in a distributed manner, aiming to maximize the surveillance quality for all the targets.

The task schedule of each sensor includes the sensing radius and the active time. Let $\mathbb{s}^i_l$ denote the lth possible task schedule for the sensor s_i, that is,

$$\mathbb{s}^i_l = \left(r_l, t^l_1, t^l_2, \ldots, t^l_n\right)$$

where any t^l_h in $\mathbb{s}^i_l$ is the Boolean variable that represents whether s_i is activated at the time slot t_h, that is,

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

The active time of each sensor in one cycle should satisfy (4), and therefore, any legal task schedule for the sensor s_i should satisfy the following expression:

$$\sum_{h=1}^{n} t^l_h = m - n + 1$$

where m is the number of available sensing radii.

Let $\mathbb{S}^i_{\text{weak}}$ denote the set of legal task schedules for s_i that have surveillance contributions to $\mathbb{o}^{\text{weak}}_{i,\text{loc}}$. If $\mathbb{s}_i$ is scheduled by applying the task schedule $\mathbb{s}^i_l \in \mathbb{S}^i$, the local surveillance quality matrix $\mathcal{M}^{\text{QoS}}_{i,\text{loc}}$ should be updated according to $\mathbb{s}^i_l$. Let $\dot{\mathcal{M}}^{\text{QoS}}_{i,\text{loc}}(\mathbb{s}^i_l)$ denote the updated local surveillance quality matrix, that is,

$$\dot{\mathcal{M}}^{\text{QoS}}_{i,\text{loc}}\left(\mathbb{s}^i_l\right) = \begin{pmatrix} \dot{p}_{1,1}\left(\mathbb{s}^i_l\right) & \cdots & \dot{p}_{1,n}\left(\mathbb{s}^i_l\right) \\ \vdots & \ddots & \vdots \\ \dot{p}_{m,1}\left(\mathbb{s}^i_l\right) & \cdots & \dot{p}_{m,n}\left(\mathbb{s}^i_l\right) \end{pmatrix} \tag{28}$$

where each element $\dot{p}_{v,h}(\mathbb{s}^i_l)$ in $\dot{\mathcal{M}}^{\text{QoS}}_{i,\text{loc}}(\mathbb{s}^i_l)$ is the updated cooperative detection probability to space–time point (o^{tar}_v, t_h) when $\mathbb{s}_i$ is scheduled by applying $\mathbb{s}^i_l$. $\dot{p}_{v,h}(\mathbb{s}^i_l)$ can be calculated as follows:

$$\dot{p}_{v,h}\left(\mathbb{s}^i_l\right) = 1 - \left(1 - p_{v,h}\right)\left(1 - p\left(s_i, o^{\text{tar}}_v\right) \times t^l_h\right).$$

Then, the new local bottleneck space–time point, denoted by $\dot{\mathbb{o}}^{\text{weak}}_{i,\text{loc}}(\mathbb{s}^i_l)$, in LMR^{st}_i can be obtained according to (25), that is,

$$\dot{\mathbb{o}}^{\text{weak}}_{i,\text{loc}}\left(\mathbb{s}^i_l\right) = \arg \min_{(o^{\text{tar}}_v, t_h) \in \text{LMR}^{\text{st}}_i} \dot{p}_{v,h}\left(\mathbb{s}^i_l\right). \tag{29}$$

This step aims to schedule the sensor with the maximum surveillance contribution to enhance the cooperative detection probability to the local bottleneck space–time point. Let $\mathbb{b}^{\text{weak}}(\mathbb{s}_l^i)$ denote the benefit of s_i selecting $\mathbb{s}_l^i \in \mathbb{S}_{\text{weak}}^i$, which can be calculated as follows:

$$\mathbb{b}^{\text{weak}}\left(\mathbb{s}_l^i\right) = \dot{p}_{v,h}^{\text{weak}}\left(\mathbb{s}_l^i\right) - p_{v,h}^{\text{weak}} \tag{30}$$

where $\dot{p}_{v,h}^{\text{weak}}(\mathbb{s}_l^i)$ and $p_{v,h}^{\text{weak}}$ are the cooperative detection probability to $\dot{\mathbb{o}}_{i,\text{loc}}^{\text{weak}}(\mathbb{s}_l^i)$ and $\mathbb{o}_{i,\text{loc}}^{\text{weak}}$, respectively. Equation (30) reflects the improved surveillance quality if s_i is scheduled by applying $\mathbb{s}_l^i$.

Let $\mathbb{s}_{\text{best}}^i$ denote the best task schedule for s_i, which can be derived by

$$\mathbb{s}_{\text{best}}^i = \arg \max_{\mathbb{s}_l^i \in \mathbb{S}_{\text{weak}}^i} \left[\mathbb{b}^{\text{weak}}\left(\mathbb{s}_l^i\right)\right]. \tag{31}$$

However, there may exist more than one task schedule that satisfies (31). Let $\mathbb{S}_{\text{best}}^{i,\text{pote}}$ denote the set of task schedules that satisfy (31), each of which has the potential to be selected as $\mathbb{s}_{\text{best}}^i$. To finally determine the task schedule for s_i, the MSQBE further checks other advantages of each task schedule in $\mathbb{S}_{\text{best}}^{i,\text{pote}}$. To enhance the surveillance quality at $\mathbb{o}_{i,\text{loc}}^{\text{weak}}$ while also distributing surveillance capability as evenly as possible among all space–time points in LMR_i^{st}, the task schedule that can minimize the differences in cooperative detection probability across all space–time points in LMR_i^{st} will be finally selected as $\mathbb{s}_{\text{best}}^i$. It is noticed that the differences in cooperative detection probability across all space–time points in LMR_i^{st} can be evaluated by the variance of all the elements in the local surveillance quality matrix. Let $\sigma^2(\mathbb{s}_l^i)$ denote the variance of all the elements in the updated local surveillance quality matrix $\dot{\mathcal{M}}_{i,\text{loc}}^{\text{QoS}}(\mathbb{s}_l^i)$ if s_i is scheduled by applying $\mathbb{s}_l^i$, and the value of $\sigma^2(\mathbb{s}_l^i)$ can be calculated as follows:

$$\sigma^2(\mathbb{s}_l^i) = \frac{1}{m \times n} \sum_{j=1}^{m} \sum_{k=1}^{n} [\dot{p}_{j,k}(\mathbb{s}_l^i) - \mu]^2 \tag{32}$$

where μ is the average of all the elements in $\dot{\mathcal{M}}_{i,\text{loc}}^{\text{QoS}}(\mathbb{s}_l^i)$.

The best task schedule for s_i can be finally determined by applying the following expression:

$$\mathbb{s}_{\text{best}}^i = \arg \min_{\mathbb{s}_l^i \in \mathbb{S}_{\text{best}}^{i,\text{pote}}} \left[\sigma^2\left(\mathbb{s}_l^i\right)\right]. \tag{33}$$

By applying the operations in this step, the best task schedule for each sensor can be designed in a distributed manner, which aims to maximize the surveillance quality for all the targets.

The following gives an example that continues the example given in Fig. 5. Given that the current local bottleneck space–time point $\mathbb{o}_{3,\text{loc}}^{\text{weak}}$ in LMR_3^{st} is (o_2^{tar}, t_1), $\mathbb{S}_{\text{weak}}^3 = \{\mathbb{s}_1^3, \mathbb{s}_2^3, \mathbb{s}_3^3, \mathbb{s}_4^3\}$, where $\mathbb{s}_1^3 = (r_1, 1, 1, 1)$, $\mathbb{s}_2^3 = (r_2, 1, 0, 1)$, $\mathbb{s}_3^3 = (r_2, 1, 1, 0)$, and $\mathbb{s}_4^3 = (r_3, 1, 0, 0)$. For each task schedule $\mathbb{s}_l^3 \in \mathbb{S}_{\text{weak}}^3$, the updated local surveillance quality matrix $\dot{\mathcal{M}}_{3,\text{loc}}^{\text{QoS}}(\mathbb{s}_l^3)$ can be obtained by applying (28), which is given in Fig. 5. The new local bottleneck space–time points under the task schedule $\mathbb{s}_l^3$ can be obtained by applying (29), which are marked in red ink the updated local surveillance quality matrices shown in Fig. 5. Each benefit of s_3 selecting $\mathbb{s}_l^3 \in \mathbb{S}_{\text{weak}}^3$ can be calculated according to (30) and is shown in Fig. 5. It is noticed that task schedules $\mathbb{s}_2^3$, $\mathbb{s}_3^3$, and $\mathbb{s}_4^3$ satisfy (31). Consequently, $\mathbb{S}_{\text{best}}^{3,\text{pote}} = \{\mathbb{s}_2^3, \mathbb{s}_3^3, \mathbb{s}_4^3\}$, which is marked by a purple dotted box in Fig. 5. Therefore, the variance of all the elements in corresponding updated local surveillance quality matrices is further calculated and displayed in Fig. 5. By applying (33), the best task schedule $\mathbb{s}_{\text{best}}^3$ is $\mathbb{s}_3^3$, which is marked by a green dotted box in Fig. 5.

Table II summarizes the main process of the MSQBE algorithm. Steps 1 and 2 summarize the time dividing phase, step 3 summarizes the detection probability calculation phase, and steps 4 and 5 finally summarize the sensor scheduling phase.

V. Performance Evaluations

This section studies the performance improvement of the proposed MSQBE against the existing NEDA [10] and TCSAR [14]. The NEDA adopts the battery-powered sensor with a fixed sensing radius and the BSM as its sensing model. It selectively activates sensors to monitor all targets, aiming to find a set of coverage schemes and determine the activation time of each coverage scheme so that the sensor's energy is optimally allocated to maximize network lifetime while guaranteeing monitoring quality. The TCSAR adopts solar-powered sensors with adjustable sensing radius and PSM as its sensing model. It assumes that the PV power is constant, initially schedules all the sensors with the maximum sensing radius, and then selectively reduces the sensing radius of some sensors, aiming to further enhance the surveillance quality.

A. Simulation Environment

Table III lists the simulation parameters of the experimental study. The MATLAB R2022b is used as the simulation tool. The size of the considered area is 400 × 400 m. The number of randomly deployed sensors ranges from 200 to 1000, while the number of targets ranges from 20 to 100. The minimum sensing radius is 10 m, while the maximum sensing radius is 40 m. The communication radius is 80 m. The radius of the guaranteed sensing region is 5 m. The battery capacity of each sensor is 1080 J. The maximum power consumption rate is 60 J/h. The historical daily PV power data and the corresponding meteorological data are from the dataset of Desert Knowledge Australia (DKA) solar center.

Fig. 6 depicts a scenario of the deployed WSNs. There are 800 sensors and 60 targets randomly deployed in a 400 × 400 m monitoring region.

B. Simulation Results

As is widely recognized, typical weather conditions are primarily categorized into three types: clear, cloudy, and rainy days. This classification is also supported by the analysis of the dataset. Fig. 7 compares the surveillance quality of three compared algorithms under clear, cloudy, and rainy conditions by varying the number of deployed sensors and available sensing radii. The number of deployed sensors and the available sensing radii vary from 200 to 1000 and from 3 to 7, respectively. The number of targets is 60.

TABLE II
MSQBE ALGORITHM

Inputs:
1. A set of solar power sensors with adjustable sensing radius $S = \{s_1, s_2, \dots, s_n\}$.
2. The set of monitored targets $O^{tar} = \{o_1^{tar}, o_2^{tar}, \dots o_m^{tar}\}$
3. The set of historical daily meteorological data matrix $\{\mathcal{M}_{5\times n,1}^{met-hist}, \mathcal{M}_{5\times n,2}^{met-hist}, \dots, \mathcal{M}_{5\times n,365}^{met-hist}\}$

Outputs: The scheduled sensors form a target coverage

Step 1. /* ***PV Power Evaluation*** */
For *j*=1:365 {
Establish the forecasted meteorological data matrix $\mathcal{M}_{5\times n}^{met-fore}$
On the *j*th day
Calculates the difference matrix $\mathcal{M}_{5\times n}^{diff}(i,j)$ according to Exp. (13)
}
The template meteorological data matrix $\mathcal{M}_{5\times n}^{tem}$ for the following day is $\mathcal{M}_{5\times n}^{tem} = arg \min_{\mathcal{M}_{5\times n,i}^{met-hist} \in \mathbb{M}^{met-hist}} ||\mathcal{M}_{5\times n}^{diff}(i,j)||_F$

Step 2. /****Calculating the Length of One Time Slot and Cycle****/
Calculates the length of one time slot and cycle according to Exps. (16) and (22), respectively

Step 3. /* ***Detection Probability Calculation*** */
If (target o_v^{tar} is covered by the sensor s_i)
Calculates $p(s_i, o_v^{tar})$ according to Exp. (1)
else if (target o_v^{tar} covered by a set of sensors S_v^{cov}
Calculates $p^{co-det}(S_v^{cov}, o_v^{tar})$ according to Exp. (23)

Step 4. /* ***Identifying the Bottleneck Space-Time Point*** " */
For(each unscheduled sensor s_i) {
Establishes the local surveillance quality matrix $\mathcal{M}_{i,loc}^{QoS}$ according to Exp. (26)
Establishes local current schedule matrix $\mathcal{M}_{i,loc}^{act}$ according to Exp. (27)
Identifies the local bottleneck space-time point $\mathbb{o}_{i,loc}^{weak}$ according to Exp. (25)
}

Step 5. /* ***Designing the Task Schedule with the Maximum Contribution to Monitor the Bottleneck Space-Time Point*** */
For (each possible task schedule $\mathbb{s}_l^i$ for s_j {
Calculates the updated local surveillance quality matrix $\dot{\mathcal{M}}_{i,loc}^{QoS}(\mathbb{s}_l^i)$ according to Exp. (28)
Identifies the new local bottleneck spacetime point, $\dot{\mathbb{o}}_{i,loc}^{weak}(\mathbb{s}_l^i)$, according to Exp. (29)
Calculates the benefit $\mathbb{b}^{weak}(\mathbb{s}_l^i)$ according to Exp. (30)
If (only one task schedule $\mathbb{s}_l^i$ satisfies Exp. (31))
$\mathbb{s}_l^i$ is the $\mathbb{s}_{best}^i$
else
For each task schedule $\mathbb{s}_l^i$ satisfies Exp. (31) {
Calculates the variance of all the elements in the updated local surveillance quality matrix $\sigma^2(\mathbb{s}_l^i)$
Determines the best task schedule according to Exp. (33)
}
}
Repeat executes the operations from Step 4 to step 5
Until all the sensors are scheduled

In general, the surveillance qualities of the three compared algorithms have a similar trend in that they are increased with the number of deployed sensors. This occurs because more sensors can be activated in each time slot, leading to a high surveillance quality. The other trend is that the surveillance qualities of MSQBE and TCSAR are increased with the number of available sensing radii. The reason is that sensors with a larger number of available sensing radii enable

TABLE III
SIMULATION PARAMETERS

Parameters	Value
Simulation Tool	MATLAB R2022b
Monitoring Region	400m×400m
Number of Sensors	200-1000
Number of Targets	20-100
Minimum Sensing Radius	10m
Maximum Sensing Radius	40m
Communication Radius	80m
Battery Capacity of Each sensor	1080 Joule
The Radius of Guaranteed Sensing Region	5m
Maximum Power Consumption Rate	60 Joule/hour
PV Power and Meteorological Data	Dataset from DKA Solar Center
Deployment Strategy	Randomly

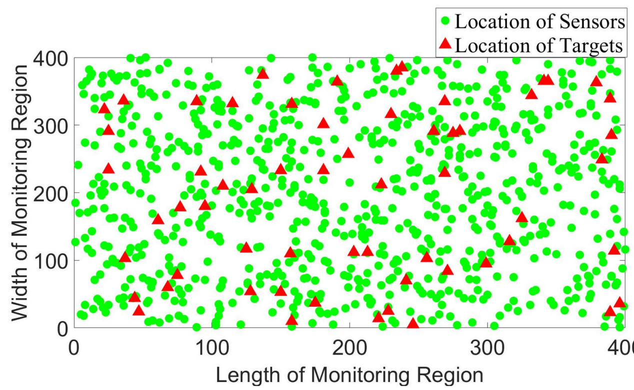
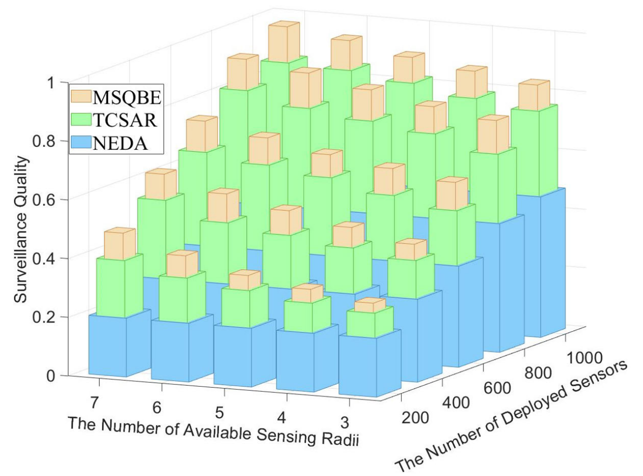
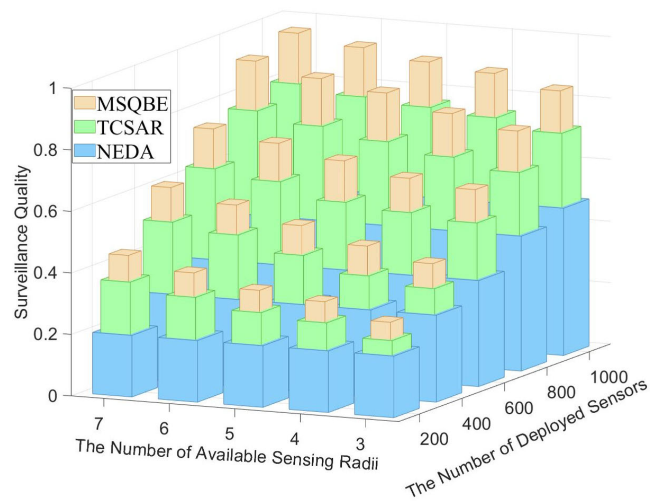


Fig. 6. Scenario that 800 sensors and 60 targets are randomly deployed in a monitoring region with the size 400 × 400 m.

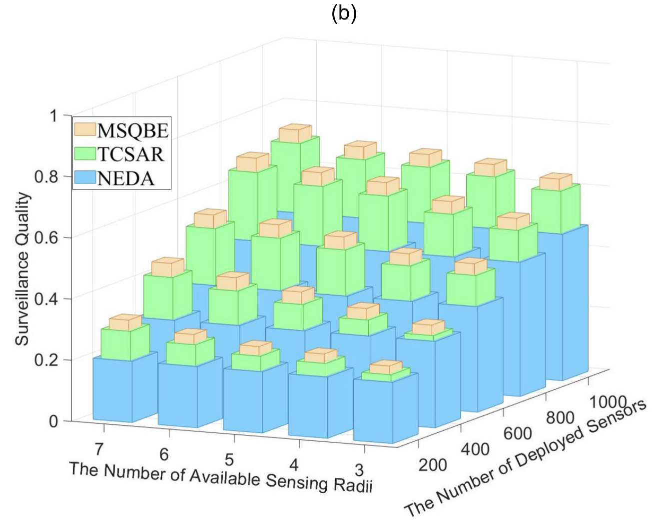
a more refined partitioning of detection capabilities, thereby allowing for a more precise allocation of detection capabilities to bottleneck space–time points. This improves the utilization of each sensor's detection capability, resulting in improved surveillance quality. Furthermore, the surveillance qualities of MSQBE and TCSAR are better on clear day compared to cloudy day and rainy day. This is because the solar-powered sensor can harvest more energy on a clear day. Consequently, each sensor can be activated in more time slots per cycle, which improves the surveillance quality. In comparison, the proposed MSQBE algorithm outperforms the TCSAR and NEDA in terms of surveillance quality in all cases. This occurs because the MSQBE algorithm considers the variable and weather-dependent nature of solar energy. It employs the similarity-based calculation to evaluate the PV power for the following day such that the available energy from solar power in the following day can be evaluated precisely. In contrast, TCSAR assumes that the PV power is constant. Under this assumption, the evaluation of PV power might be overly conservative to prevent some sensors from becoming nonoperational due to power depletion. Meanwhile, NEDA



(a)



(b)



(c)

Fig. 7. Comparison of MSQBE, TCSAR, and NEDA in terms of surveillance quality with varying numbers of deployed sensors and selectable sensing radius in different weather conditions. (a) On a clear day. (b) On a cloudy day. (c) On a rainy day.

adopts sensors with fixed sensing radii, which reduces the utilization of detection capability. As a result, the surveillance quality drops greatly compared to that of MSQBE.

Fig. 8 investigates the surveillance quality of the three compared algorithms under clear, cloudy, and rainy conditions by varying the number of targets and the deployed sensors. The number of deployed sensors varies from 200 to 1000, and the number of targets varies from 20 to 100. As shown in Fig. 8, the surveillance qualities of three compared algorithms generally decrease with the number of targets. It occurs because, when the number of targets is increased, more sensors are required to monitor each target. Similar to the results obtained In Fig. 7, both MSQBE and TCSAR obtain better results on clear day compared to cloudy and rainy days. The proposed MSQBE consistently outperformed TCSAR and NEDA in all cases. This occurs because, in contrast to TCSAR, which initially schedules all sensors with the maximum sensing radius and subsequently reduces the sensing radius of some sensors to enhance surveillance quality, MSQBE adjusts the sensing radius for each sensor during the task schedule design to maximize surveillance quality. This scheme allows MSQBE to exploit more opportunities for enhancing surveillance quality to space–time points. In addition, unlike NEDA, which adopts the BSM, the MSQBE algorithm, based on the PSM, can accurately identify the space–time points with the weakest surveillance quality and schedule the sensors with the maximum surveillance contribution to those points, resulting in a higher surveillance quality.

Fig. 9 further investigates the surveillance stability of three compared algorithms in both spatial and temporal dimensions. The surveillance stability can be quantified by the fairness index, which indicates the degree of deviation in surveillance quality over a given number of n time slots or across specified $m$targets. Recall that $p_{v,h}$ represents the cooperative detection probability to space–time point (o_v^{tar}, t_h). Let ζ_v and ζ_h denote the fairness indices of vth selected target across n time slots and hth time slot on $m$targets, respectively. ζ_j and ζ_i are defined as follows:

$$\zeta_v = \frac{\left(\sum_{v=1}^{n} p_{v,h}\right)^2}{n \sum_{v=1}^{n} \left(p_{v,h}\right)^2} \tag{34}$$

$$\zeta_h = \frac{\left(\sum_{h=1}^{m} p_{v,h}\right)^2}{m \sum_{h=1}^{m} \left(p_{v,h}\right)^2}. \tag{35}$$

In Fig. 9, each cycle comprises six time slots, and the number of randomly selected targets is 9. Fig. 9(a) first compares the fairness index of six time slots on nine randomly selected targets. The value of the fairness index approaching 1 indicates that the surveillance qualities on six time slots are stable. The proposed MSQBE outperforms the TCSAR and NEDA in terms of the fairness index of six time slots on nine randomly selected targets. This occurs because the MSQBE adjusts the sensing radius of each sensor by designing the best task schedule that aims to maximize the surveillance quality. In addition, MSQBE distributes the surveillance capability of each sensor as evenly as possible among all space–time points when multiple task schedules all have the maximum surveillance contribution to the space–time points. Fig. 9(b) further compares the fairness index of nine randomly selected targets on six time slots in one cycle. Results align with those presented in Fig. 9(a). The proposed MSQBE outperforms the

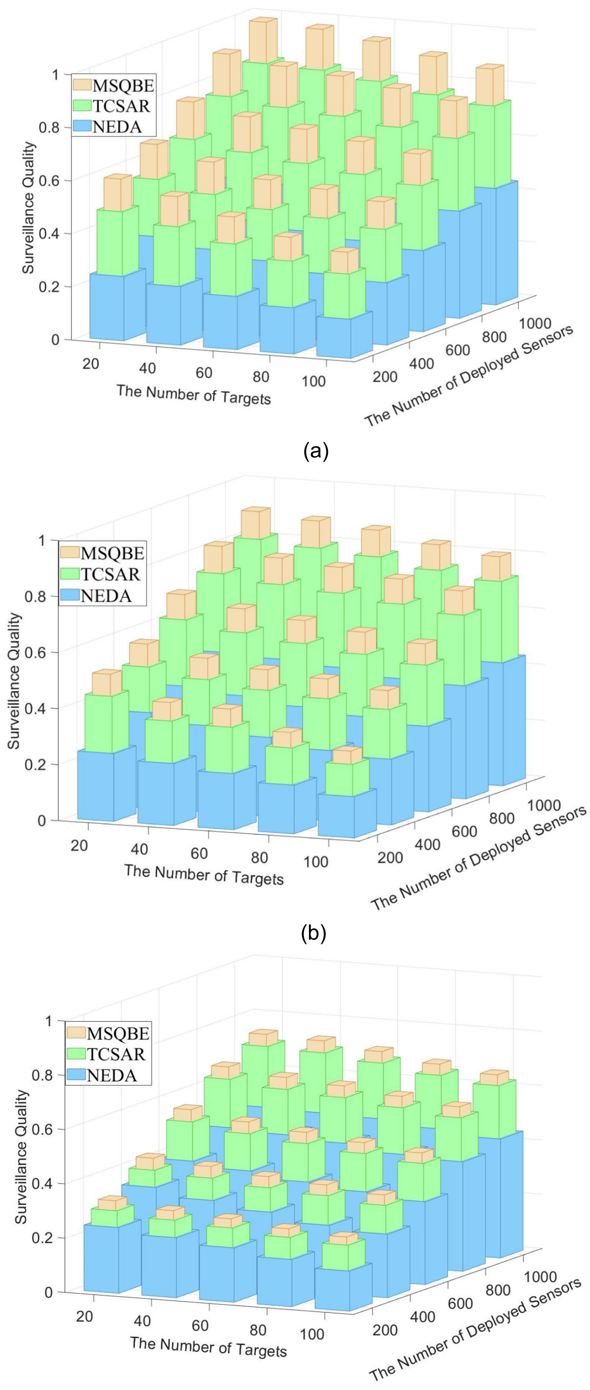


Fig. 8. Comparison of MSQBE, TCSAR, and NEDA in terms of surveillance quality with varying numbers of targets and the number of deployed sensors. (a) On a clear day. (b) On a cloudy day. (c) On a rainy day.

TCSAR and NEDA in terms of nine randomly selected targets on six time slots in one cycle.

Fig. 10 depicts the comparison of MSQBE and TCSAR in terms of the utilization of solar power under clear, cloudy, and rainy conditions. Since the NEDA algorithm does not use solar-powered sensors, it was not included in the comparison. Fig. 10(a)–(c) details the actual available energy (AAE) under clear, cloudy, and rainy conditions, respectively. In addition,

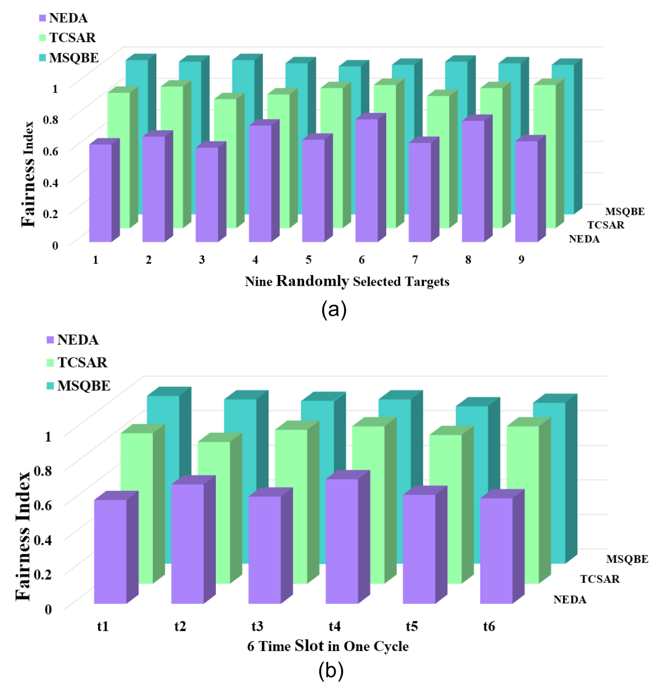


Fig. 9. Comparison of MSQBE, TCSAR, and NEDA in terms of surveillance stability. Fairness index of different (a) time slots and (b) targets.

these figures compare the actual consumed energy (ACE) and actual utilization of solar power (AUSP) of the two compared algorithms under the same conditions, with each of Fig. 10(a)–(c) comparing ten days selected from respective weather conditions. Because the TCSAR assumes that PV power is constant and schedules sensors based on the constraint that charger energy equals consumed energy, it does not consider the variation of PV power and the effects of weather conditions. Therefore, the TCSAR adopts the average PV power of each month as the daily charging rate of the sensors for that same month. The value of AUSP can be calculated by applying the following expression:

$$\text{AUSP} = \frac{\text{ACE}}{\text{AAE}}. \tag{36}$$

It should be noted that AUSP reaches 100% when ACE is equal to the AAE, which is the ideal utilization of solar power (IUSP). In Fig. 10, the red point indicates that the AUSP 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 AUSP is larger 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 AUSP is to the IUSP, the better the performance in terms of solar power utilization.

As shown in Fig. 10, the solar power utilization of MSQBE achieves the best performance under rainy conditions. This occurs because the PV power under rainy conditions is relatively low and exhibits less variation. The solar power utilization of MSQBE achieves the worst performance under

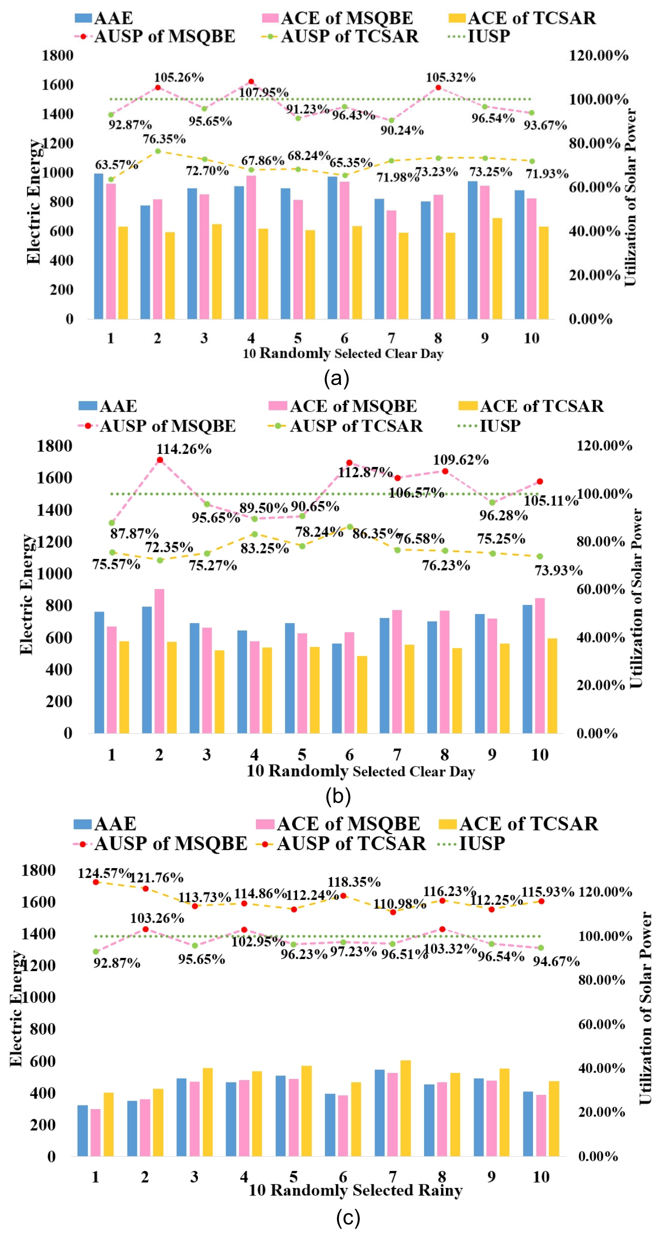


Fig. 10. Comparison of MSQBE and TCSAR in terms of utilization of solar power. (a) On a clear day. (b) On a cloudy day. (c) On a rainy day.

cloudy conditions. This occurs because PV power under cloudy conditions is influenced by random cloud cover and is relatively more unstable compared to rainy and cloudy conditions, leading to higher discrepancies between the actual and the estimated solar energy. However, overall, the performance of MSQBE is better than that of TCSAR in all cases. This occurs because the proposed MSQBE employs 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 following day, aiming to achieve an accurate estimation of the following day's PV power. The TCSAR assumes that the PV power is constant throughout each day, which will lead a significant discrepancies between actual solar power and the estimated solar power, that is, the discrepancies between AAE and ACE of TCSAR. Therefore, the target coverage formed by the TCSAR poses a significant risk of sensor failure due to battery depletion on rainy days and results in a substantial waste of solar power on clear and cloudy days.

VI. Conclusion

This article proposes a target coverage algorithm for solar-powered WSNs, called MSQBE, which applies adjustable sensing radius sensors with the PSM, aiming to maximize the surveillance quality of the target coverage while balancing the acquired and consumed energy of each sensor. To achieve this, the MSQBE 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 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. Then, MSQBE calculates the surveillance quality of each space–time point, which consists of each target and time slot, identifies the "bottleneck space–time point," and schedules the sensors with the maximum surveillance contribution to monitor it. Extensive experiments demonstrate that the proposed MSQBE outperforms the existing TCSAR and NEDA in terms of surveillance quality, surveillance stability, and solar power utilization. However, the method based on similarity calculation used in this article to estimate the PV power for the next day is still not precise enough. In addition, compared to variable radius sensors, mobile sensors have a greater capability to enhance monitoring quality. Therefore, our future work will further utilize deep learning techniques to predict PV power and employ mobile sensors to further improve the network's monitoring quality.

References

[1] M. A. Jamshed, K. Ali, Q. H. Abbasi, M. A. Imran, and M. Ur-Rehman, "Challenges, applications, and future of wireless sensors in Internet of Things: A review," *IEEE Sensors J.*, vol. 22, no. 6, pp. 5482–5494, Mar. 2022.

[2] L. Abualigah, A. Diabat, P. Sumari, and A. H. Gandomi, "Applications, deployments, and integration of Internet of Drones (IoD): A review," *IEEE Sensors J.*, vol. 21, no. 22, pp. 25532–25546, Nov. 2021.

[3] 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.

[4] 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.

[5] Y. Chen, S. Huang, and R. Fitch, "Active SLAM for mobile robots with area coverage and obstacle avoidance," *IEEE/ASME Trans. Mechatronics*, vol. 25, no. 3, pp. 1182–1192, Jun. 2020.

[6] 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.

[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] 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.

[10] 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.

[11] 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.

[12] X. Xu, Z. Dai, A. Shan, and T. Gu, "Connected target ϵ-probability coverage in WSNs with directional probabilistic sensors," *IEEE Syst. J.*, vol. 14, no. 3, pp. 3399–3409, Sep. 2020.

[13] 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.

[14] Q. Zhang, C.-Y. Chang, Z. Dong, and D. S. Roy, "TCSAR: Target coverage mechanism for sensors with adjustable sensing range in WRSNs," *IEEE Sensors J.*, vol. 22, no. 4, pp. 3756–3765, Feb. 2022.

[15] H. Keshmiri and H. Bakhshi, "A new 2-phase optimization-based guaranteed connected target coverage for wireless sensor networks," *IEEE Sensors J.*, vol. 20, no. 13, pp. 7472–7486, Jul. 2020.

[16] P. Le Nguyen, Y. Ji, M. K. Pham, H. Le, and T. H. Nguyen, "$(1+\epsilon)^2$-and polynomial-time approximation algorithms for network lifetime maximization with relay hop bounded connected target coverage in WSNs," *IEEE Sensors J.*, vol. 21, no. 7, pp. 9577–9599, Apr. 2021.

[17] 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.

[18] H. Xue, H. Chen, Z. Ni, X. Liu, and F. Xia, "Towards maximizing coverage of targets for WRSNs by multiple chargers scheduling," *IEEE Trans. Mobile Comput.*, early access, Feb. 23, 2024, doi: 10.1109/TMC.2024.3369054.

[19] C. Li, K.-W. Chin, and C. Yang, "Complete target coverage in radio frequency and solar-powered sensor networks," *IEEE Syst. J.*, vol. 15, no. 3, pp. 3609–3619, Sep. 2021.

[20] A. Rossi, A. Singh, and M. Sevaux, "An exact approach for maximizing the lifetime of sensor networks with adjustable sensing ranges," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3166–3176, 2012.

[21] G. K. Singh, "Solar power generation by PV (photovoltaic) technology: A review," *Energy*, vol. 53, pp. 1–13, May 2013.

[22] 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.

[23] C. Shen, C. Fan, Y. Wang, and W. Xue, "Limited memory BFGS algorithm for the matrix approximation problem in Frobenius norm," *Comput. Appl. Math.*, vol. 39, no. 2, pp. 1–25, Feb. 2020.

[24] C. Yang, Z. Fan, W. Lu, and H. Gao, "Uncertain iterative optimal attitude control method for periodic satellite with reliability constraint," *IEEE Trans. Aerosp. Electron. Syst.*, early access, May 28, 2024, doi: 10.1109/TAES.2024.3404915.

[25] Q. Yu, C. Yang, G. Dai, L. Peng, and J. Li, "A novel penalty function-based interval constrained multi-objective optimization algorithm for uncertain problems," *Swarm Evol. Comput.*, vol. 88, Jul. 2024, Art. no. 101584.

[26] Q. Yu, C. Yang, G. Dai, L. Peng, and X. Chen, "Synchronous wireless sensor and sink placement method using dual-population coevolutionary constrained multi-objective optimization algorithm," *IEEE Trans. Ind. Informat.*, vol. 19, no. 6, pp. 7561–7571, Jun. 2023.

Chi Zhang received the B.S. degree in educational technology from Huaibei Normal University, Huaibei, China, in 2012, and the M.S. degree in computer applications technology from Shenyang Aerospace University, Shenyang, China, in 2015.

He is currently a Laboratory Technician with the School of Artificial Intelligence and Big Data, Hefei University, Hefei, China. His current research interests include deep learning and industrial digitalization.

Youxi Li received the M.S. degree in communication engineering from China University of Geosciences, Wuhan, China, in 2009. He is currently pursuing the Ph.D. degree in computer science and information engineering with Tamkang University, New Taipei City, Taiwan.

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

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.

Chih-Yung Chang (Member, IEEE) received the Ph.D. degree in computer science and information engineering from National Central University, Taoyuan, 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 City, 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 served as an Associate Guest Editor for several SCI-indexed journals, including *Journal of Internet Technology* (JIT) from 2004 to 2008, *Journal of Information Science and Engineering* (JISE) in 2008, *Telecommunication Systems* (TS) in 2010, *IET Communications* in 2011, and *International Journal of Distributed Sensor Networks* (IJDSN) from 2012 to 2014. He has been serving as an Associate Guest Editor for several SCI-indexed journals, including *International Journal of Ad Hoc and Ubiquitous Computing* (IJAHUC) since 2011 and *Journal of Applied Science and Engineering* (JASE) since 2018.

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