Hindawi
Wireless Communications and Mobile Computing
Volume 2021, Article ID 5921181, 16 pages
https://doi.org/10.1155/2021/5921181

Research Article

Efficient Energy Utilization with Device Placement and Scheduling in the Internet of Things

Yanli Zhu,[1,2] Xiaoping Yang,[1] Yi Hong,[3,4] Youfang Leng,[1] and Chuanwen Luo[3,4]

[1]*School of Information, Renmin University of China, Beijing 100872, China*
[2]*School of Information and Engineering, Henan Institute of Science and Technology, Xinxiang 453003, China*
[3]*School of Information Science and Technology, Beijing Forestry University, Beijing 100083, China*
[4]*Engineering Research Center for Forestry-Oriented Intelligent Information Processing of National Forestry and Grassland Administration, Beijing 100083, China*

Correspondence should be addressed to Chuanwen Luo; chuanwenluo@bjfu.edu.cn

Received 30 July 2021; Accepted 19 August 2021; Published 14 September 2021

Academic Editor: Yan Huang

The low-power wide-area network (LPWAN) technologies, such as LoRa, Sigfox, and NB-IoT, bring new renovation to the wireless communication between end devices in the Internet of things (IoT), which can provide larger coverage and support a large number of IoT devices to connect to the Internet with few gateways. Based on these technologies, we can directly deploy IoT devices on the candidate locations to cover targets or the detection area without considering multihop data transmission to the base station like the traditional wireless sensor networks. In this paper, we investigate the problems of the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS). In the problems, we consider both the placement and scheduling of IoT devices to monitor all targets (or the whole detection area) such that all targets (or the whole area) are (or is) continuously observed for a certain period of time. The objectives of the problems are to minimize the total energy consumption of the IoT devices. We first, respectively, propose the mathematical models for the MTPS and MAPS problems and prove that they are NP-hard. Then, we study two subproblems of the MTPS problem, minimum location coverage (MLC), and minimum energy consumption scheduling deployment (MESD) and propose an approximation algorithm for each of them. Based on these two subproblems, we propose an approximation algorithm for the MTPS problem. After that, we investigate the minimum location area coverage (MLAC) problem and propose an algorithm for it. Based on the MLAC and MESD problems, we propose an approximation algorithm to solve the MAPS problem. Finally, extensive simulation results are given to further verify the performance of the proposed algorithms.

1. Introduction

The Internet of things (IoT) is a flourishing paradigm in the scenario of modern wireless telecommunications, which has been provided a wide diversity applications for all walks of life in modern time, such as home automation, transportation, industry, agriculture, mobile device applications [1], and smart systems [2]. IoT applications are required a growing number of technologies to offer low-power operation and low-cost and low-complexity end devices that will be able to communicate wirelessly over long distances. With the development of the Low Power Wide Area Network (LPWAN) technologies, such as SigFox, NB-IoT and LoRa, the low power long-range wide-area communication has become a reality [3]. Since the long range communication of the LPWAN technologies is gradually used in the Internet of things, the IoT devices can only communicate with LPWAN gateways and not directly with each other. Taking the LoRa example, a single gateway can support as many as 10^5 IoT devices and three gateways are enough to cover all devices in the urban area within an approximate 15 km radius [4]. The architecture of the LPWAN-based Internet

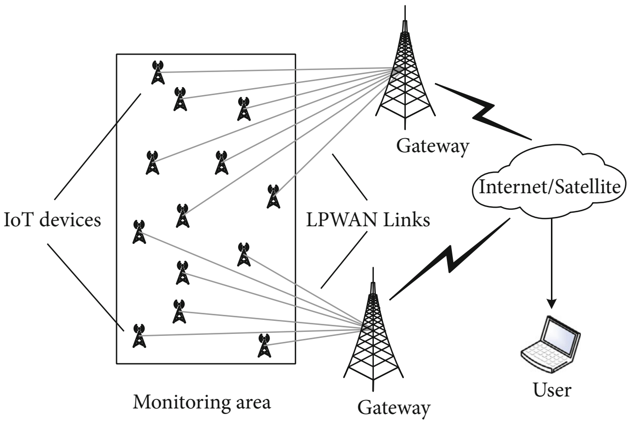


Figure 1: The architecture of the LPWAN-based Internet of things.

of things is shown in Figure 1 [5], in which IoT devices are deployed in the monitoring area to observe targets or the whole monitoring area; the installation of few gateways over the territory allows to gather data from IoT devices that are placed at different miles from the gateways. Then, the received data by the gateways are transmitted to the users through the Internet or satellite for further computational analysis to determine the appropriate response mechanism.

Therefore, we can directly deploy IoT devices to monitor all targets (or the whole monitoring area) in any region with the LPWAN-based network without considering other data transmission methods such as virtual backbone networks [6, 7] and mobile data collectors [8, 9]. Since many IoT devices are battery-powered sensors, for example, in wireless sensor networks (WSNs) and ad hoc networks (ANs), the power usage profile should be carefully designed in order to extend the battery lifetime. How to prolong the network lifetime is a classic problem in WSNs, which is called the coverage problem [10]. Given the m targets (or the entire monitoring area) and n IoT devices in the monitoring area, the coverage problem is to schedule the activity of the IoT devices such that all targets (or the whole monitoring area) are (or is) continuously observed and the network lifetime is maximized. Research on the coverage problem benefits a lot of applications, such as environment monitoring, battlefield surveillance, indoor guarding, smart space, industrial diagnostics, and military facility [11]. Recently, many researchers proposed various problems and corresponding algorithms for the coverage problem. In [12], Cardei et al. studied the target coverage problem with the objective of maximizing the network lifetime of a power constrained WSN deployed for detecting of a set of targets with known locations, in which they did not consider the placement of sensors. In [13], Akhlaghinia et al. studied the heterogeneous point coverage problem in sensor placement to cover a large number of target points with various coverage requirements using a minimum number of heterogeneous sensors. In the problem, they only investigated the placement of sensors without considering network lifetime. In [14], Mini et al. considered both the deployment locations and scheduling of the given IoT devices to maximize the network lifetime with the required coverage level. However, they deployed all available IoT devices to cover targets randomly without considering their candidate sites. In [15], Hanh et al. investigated the problem of maximizing the area coverage in heterogeneous WSNs. The goal of the problem is to find an optimal placement scheme for the given set of sensors so that the coverage area is maximized. In the problem, they only consider the placement of sensors without considering their candidate sites.

In the above literature, they only considered one of the deployment and scheduling of IoT devices or ignored the factor that candidate sites can be placed by IoT devices which has to be considered in some applications, such as a smart city. Actually, to minimize the total energy consumption of IoT devices, we not only need to consider the deployment of IoT devices but also their scheduling. Meanwhile, due to the emergence of wireless charging technologies and natural energy charging methods (e.g., solar charging) for the IoT devices, the current applications of IoTs have shifted from maximizing the network lifetime to working for a certain period of time. In this paper, we study the problems of the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS), where we consider both the placement and scheduling of IoT devices to monitor all targets or the entire monitoring area in a region such that all targets or the whole area are or is continuously observed for a certain period of time and the total energy consumption of all available IoT devices is minimized. The contributions of this paper are shown as follows:

(1) We propose two new practical models of minimizing the total energy consumption of all IoT devices by placing and scheduling them for continuously observing all targets or the entire detection area for a certain period of time. Then, we define the problems as the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS) and prove that they are NP-hard

(2) To solve the MTPS problem, we introduce two other problems, minimum location coverage (MLC) and minimum energy consumption scheduling deployment (MESD). Then, we propose an approximation algorithm for each of them. Afterwards, an approximation algorithm for the MTPS problem is proposed on the basis of the solutions for the MLC and MESD problems

(3) To solve MAPS problem, we introduce another problem, minimum location area coverage (MLAC). Then, we propose an approximation algorithm to solve the problem. Based on the problems MLAC and MESD, we propose an approximation algorithm to solve the MAPS problem

(4) We illustrate the effectiveness of the proposed algorithms by theoretical analysis and simulations

The remainder of this paper is organized as follows. We give the related works in Section 2. Section 3 introduces some models and definitions for the problems MTPS and MAPS. In Section 5, we propose an approximation algorithm to solve the MTPS problem. In Section 6, we propose an approximation algorithm to solve the MAPS problem. Simulations are shown in Section 7. Section 8 concludes this paper.

2. Related Works

In this section, we briefly review the major problems and methods related to the investigated problem in IoTs. As we all know, WSN is a special kind of IoTs. If there is no special explanation, the sensors in WSNs mentioned below represent IoT devices. According to the investigated problem, the related works can be categorized into three categories: IoT device placement problem, target coverage problem, and area coverage problem.

2.1. IoT Device Placement Problem. The IoT device placement problem aims at finding the least number of IoT devices and their locations within all known potential sensing locations for meeting requirements, such as [16–21].

In [16], Altinel et al. investigated the minimum cost point coverage problem with varying sensing quality and price and formulated a binary integer linear programming model for effective sensor placement on a grid-structured area. In [17], Wang introduced the sensor placement optimization problem, where the locations of targets to be covered are known and the candidate locations to place sensors are limited. The objective of the problem is to minimize the number of sensors to cover all targets, and the problem can be solved by the greedy algorithm for solving the set covering problem as shown in [22]. In [18], Gravalos et al. investigated the gateway placement problem for IoTs, which aims at finding the minimum number of gateways along with suitable IoT devices to optimize the overall installation cost without compromising the related QoS requirements. In [19], Jiang et al. proposed a group-greedy method to solve the sensor placement in linear inverse problems, which can find suboptimal solutions with near optimality guarantee using less computational cost compared with convex relaxation methods. In [20], Hasan and Al-Rizzo investigated the sensor deployment to improve the connectivity in IoT by presenting the bioinspired metaheuristics canonical particle multiswarm optimization algorithm. In [21], Jiang et al. studied the optimal sensor placement problem for an IoT-based power grid monitoring system. Then, they proposed a modified binary particle swarm optimization algorithm to determine the optimal number and location of sensors and estimate the ratio of conductor temperature alarms that can be covered by the proposed sensor placement.

2.2. Target Coverage Problem. In general, IoT devices are battery-powered sensors and there are often a lot of redundant sensors randomly placed in a region to cover a certain group of targets. How to schedule deployed sensors to maximize the network lifetime is an important problem in IoTs, which is called the maximum lifetime coverage problem (MLCP) and was proved NP-hard [12]. Currently, many researches devoted themselves to investigating the various problems of the MLCP problem, such as [23–26].

In [23], Berman et al. defined the MLCP problem as a sensor network life problem (SNLP) and proposed an approximation algorithm with a performance ratio of $1 + \ln n$ to solve the problem based on the minimum weigh sensor coverage problem (MWSCP) which aims at finding the minimum total weight of sensors to cover a certain set of targets, where n is the number of deployed sensors. In [24], Ding et al. first improved the algorithm for the MWSCP problem to $4 + \varepsilon$, where $\varepsilon > 0$. Then, they proposed an approximation algorithm with an approximate ratio of $4 + \zeta$ in the light of the MWSCP problem, where $\zeta > 0$. In [25], Lu et al. investigated the maximum lifetime coverage scheduling (MLCS) problem to address the scheduling problem for both target coverage and data collection in WSNs for maximizing the network lifetime. Then, they proposed an approximation algorithm with a constant factor for the problem. In [26], Shi et al. defined a new coverage problem in battery-free WSN, which is not only to maximize coverage quality but also to prolong network lifetime. Then, they proposed two centralized approximate algorithms and a distributed algorithm for solving the problem.

2.3. Area Coverage Problem. In [27], Xing et al. divided the detection area into grids and guaranteed the coverage of vertices of each grid to approximate the area coverage. However, the proposed approach as an approximation solution can not actually ensure the coverage of the whole area. In [28], Yu et al. studied the k-coverage problem, where a minimum subset of sensors among the deployed ones is selected such that each point in the detection area is covered by at least k sensors. In [29], Qin and Chen investigated the area coverage problem to maximize the coverage lifetime of wireless sensor networks for monitoring the area of interest. They proposed the area coverage algorithm based on differential evolution, which considered the balanced cost and minimal energy for sensors.

3. Model and Problem Definitions

In this sections, we introduce some parameters and problem definitions.

Let $\mathscr{A}$ be represented as a two-dimensional plane area; the whole of which can be observed by IoT devices that work together. There are m targets located on $\mathscr{A}$. Let $Q = \{r_1, r_2, \cdots, r_m\}$ represent the set of m targets located on $\mathscr{A}$. Due to the particularity of some detection areas, such as smart cities and farmlands, IoT devices can only be deployed on some fixed locations. We call these fixed locations as *candidate sites* of IoT devices. Let $L = \{L_1, L_2, \cdots, L_N\}$ denote the set of N candidate sites. Suppose that there exists n available IoT devices that can be deployed in the candidate sites to monitor targets in Q or the whole detection area $\mathscr{A}$. We use $S = \{s_1, s_2, \cdots, s_n\}$ to represent the set of n IoT devices, in which each IoT device s_i can be active continuously at most E_i time slots. In this paper, we assume that each IoT device $s_i \in S$ continuously works E_i time slots once it starts working and $E_i \leq T$, where T is the minimum time that the targets (or the detection area) are (or is) continuously observed. For any pair of $s_i \in S$ and $r_j \in Q$ (or $L_k \in L$ and $r_j \in Q$), let d_{s_i,r_j} (or d_{L_k,r_j}) denote the Euclidean distance between s_i and r_j (or L_k and r_j).

In this paper, we aim to find a subset $C \subseteq S$ of IoT devices which are deployed on some candidate sites in L to observe all targets in Q or the whole detection area and to minimize the total energy consumption of all IoT devices by scheduling IoT devices in C such that all targets or the whole detection area can be continuously observed by IoT devices at least T time. More formally, for two different kinds of problems, target coverage and area coverage, we call the research problems as the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and the minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS), respectively, whose detailed definitions are shown in Definitions 1 and 2.

Definition 1 (MTPS). Given a set $Q = \{r_1, r_2, \cdots, r_m\}$ of m targets located on detection area $\mathscr{A}$, a set $S = \{s_1, s_2, \cdots, s_n\}$ of n IoT devices in which all IoT devices have the same coverage range R and each IoT device $s_i \in S$ can work E_i time slots, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites to be put on IoT devices, a positive time T, the minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) problem aims at finding a subset $C \subseteq S$ of IoT devices placed on the candidate sites in L and scheduling the IoT devices in C such that

(1) For any candidate site $L_k \in L$, it can be placed in more than one IoT device from C

(2) For arbitrary target $r_j \in Q$, it is continuously observed by IoT devices in C at least T time

(3) For each IoT device $s_i \in C$, it continuously works in E_i time slots once it starts working

(4) The total energy consumption of IoT devices in C, $M = \sum_{s_i \in C} E_i$, is minimized

Definition 2 (MAPS). Given a detection area $\mathscr{A}$, a set $S = \{s_1, s_2, \cdots, s_n\}$ of n IoT devices in which all IoT devices have the same coverage range R, and each IoT device $s_i \in S$ can work E_i time slots, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites to be put in IoT devices, a positive time T, the minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS) problem aims at finding a subset $C \subseteq S$ of IoT devices placed on the candidate sites in L and scheduling the IoT devices in C such that

(1) For any candidate site $L_k \in L$, it can be placed in more than one IoT devices from C

(2) For arbitrary point $p \in \mathscr{A}$, it is continuously observed by IoT devices in C at least T time

(3) For each IoT device $s_i \in C$, it continuously works in E_i time slots once it starts working

(4) The total energy consumption of IoT devices in C, $M = \sum_{s_i \in C} E_i$, is minimized

4. Mathematical Formulation for the Problems

In this section, we will introduce the mathematical formulations for the problems MTPS and MAPS.

We first introduce some notations as follows:

i is the index of IoT devices, where $1 \leq i \leq n$. j is the index of targets, where $1 \leq j \leq m$. k is the index of candidate sites, where $1 \leq k \leq N$. t is the index of active time slots, where T time is divided into T time slots. We define the binary variables x_{ik}, a_{jk}, and y_{it} as follows:

$$x_{ik} = \begin{cases} 1, & \text{if } s_i \text{ is placed at } L_k, \\ 0, & \text{otherwise,} \end{cases}$$
$$a_{jk} = \begin{cases} 1, & \text{if } d_{L_k,r_j} \leq R, \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$
$$y_{it} = \begin{cases} 1, & \text{if } s_i \text{ is active at } T \text{ time slot,} \\ 0, & \text{otherwise.} \end{cases}$$

4.1. Mathematical Formulation for the MTPS Problem. In this subsection, we will introduce the mathematical formulation for the MTPS problem. The problem can be formulated into an integer programming (IP) problem as follows:

$$\min \sum_{k=1}^{N} \sum_{i=1}^{n} x_{ik} \cdot E_i, \tag{2}$$

s.t.

$$\sum_{k=1}^{N} \sum_{i=1}^{n} a_{jk} \cdot x_{ik} \cdot y_{it} \geq 1, \quad j = 1, 2, \cdots, m, t = 1, 2, \cdots, T, \tag{3}$$

$$\sum_{k=1}^{N} \sum_{t=b_i}^{b_i+E_i-1} x_{ik} \cdot y_{it} = E_i \, \exists b_i \in \{1, 2, \cdots, T\}, \quad i = 1, 2, \cdots, n, \tag{4}$$

$$x_{ik} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, k = 1, 2, \cdots, N, \tag{5}$$

$$a_{jk} \in \{0, 1\}, \quad j = 1, 2, \cdots, m, k = 1, 2, \cdots, N, \tag{6}$$

$$y_{it} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, t = 1, 2, \cdots, T. \tag{7}$$

The function of equation (2) is to minimize the total energy consumption of IoT devices for continuous observing of all targets at least T time. Constraint (3) ensures that for each target $r_j \in Q$, there at least exists an IoT device $s_i \in S$

located at some candidate site $L_k \in L$ to cover r_j at any tth time slot. Constraint (4) guarantees that the IoT device $s_i \in S$ located on $L_k \in L$ will run out of energy as soon as it starts working. Constraints (5)–(7) define the domains of the variables.

4.2. Mathematical Formulation for the MAPS Problem. In this subsection, we will introduce the mathematical formulation for the MAPS problem. Let S_i be the sensing region of sensor s_i. We use $||A||$ to denote the area of $\mathscr{A}$. The problem can be formulated as follows:

$$\min \sum_{k=1}^{N}\sum_{i=1}^{n} x_{ik}\cdot E_i, \tag{8}$$

$$\text{s.t.} \left\| \left(\bigcup_{s_i \in C} S_i \cdot \sum_{k=1}^{N} x_{ik}\cdot y_{it} \right) \bigcap \mathscr{A} \right\| = ||A||, \quad t = 1, 2, \cdots, T, \tag{9}$$

$$\sum_{k=1}^{N}\sum_{t=b_i}^{b_i+E_i-1} x_{ik}\cdot y_{it} = E_i \ \exists b_i \in \{1,2,\cdots,T\}, \quad i = 1, 2, \cdots, n, \tag{10}$$

$$x_{ik} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, k = 1, 2, \cdots, N, \tag{11}$$

$$y_{it} \in \{0, 1\}, \quad i = 1, 2, \cdots, n, t = 1, 2, \cdots, T, \tag{12}$$

The function of equation (8) is to minimize the total energy consumption of IoT devices for continuous observing of the whole detection area $\mathscr{A}$ at least T time. Constraint (9) ensures that for any point $p \in \mathscr{A}$, there at least exists an IoT device $s_i \in S$ located at some candidate site $L_k \in L$ to cover p at any tth time slot. Constraint (10) guarantees that the IoT device $s_i \in S$ located on $L_k \in L$ will run out of energy as soon as it starts working. Constraints (11)–(12) define the domains of the variables.

4.3. NP-Hard Proofness. In the following, we will prove that the problems MTPS and MAPS are NP-hard. We first prove that the MTPS problem is NP-hard. Then, based on the MTPS problem, we prove that the MAPS problem is also NP-hard.

We consider a special case of the MTPS problem where we set $T = 1$, $N = n$, and $E_i = 1$ for any IoT device $s_i \in S$. At this point, the objective of the MTPS problem can be transformed to find a subset $C \subseteq S$ of IoT devices with minimum cardinality such that all targets are covered. Since for any IoT device $s_i \in C$, it needs to be placed at the corresponding candidate site to cover targets, the objective of the MTPS problem changes from finding the minimum subset C to looking for the minimum subset $L' \subseteq L$ of candidate sites, where $|C| = |L'|$. Therefore, the special case of the MTPS problem can be equivalently transformed into the minimum point cover (MPC) problem as shown in Definition 3.

Definition 3 (MPC). Given a set $Q = \{r_1, r_2,\cdots,r_m\}$ of m targets, a set $L = \{L_1, L_2,\cdots,L_N\}$ of N candidate sites to be put in IoT devices, and all IoT devices have the same coverage range R, the minimum point cover (MPC) problem is to find a subset $L' \subseteq L$ of candidate sites such that all targets are covered by IoT devices located on the candidate sites in L' and the number of candidate sites $|L'|$ is minimized.

In the MPC problem, for each candidate site $L_k \in L$, we use U_k to denote the set of targets covered by L_k where for each target r_j, $r_j \in U_k$ if and only if $d_{L_k,r_j} \le R$. Let $F = \{U_1, U_2,\cdots,U_N\}$. Then, the MPC problem can be equivalently transformed into the set cover (SC) problem, as shown in Definition 4.

Definition 4 (SC). Given a set $Q = \{r_1, r_2,\cdots,r_m\}$ of m targets and a collection $F = \{U_1, U_2,\cdots,U_N\}$ of N sets, where each $U_k \in F$ is a subset of Q, the set cover (SC) problem is to find a subset $F' \subset F$ such that $\bigcup_{U_k \in F}{}' U_k = Q$ and $|F'|$ is minimum.

Theorem 5. *The MPC problem is NP-hard.*

Proof. According to the SC and MPC problems, we can obtain that the decision version of the MPC problem has a YES answer if and only if the decision version of the SC problem has a YES answer and $|F'| = |L'|$. Since the SC problem was proved NP-hard [30], the MPC problem is NP-hard. □

Theorem 6. *The MTPS problem is NP-hard.*

Proof. According to Theorem 5, we can verify that the MPC is NP-hard. Since the MPC problem is a special case of the MTPS problem, the MTPS problem is also NP-hard. □

Theorem 7. *The MAPS problem is NP-hard.*

Proof. Since the continuous region $\mathscr{A}$ is made up of an infinite set of points, we can take any set of discrete points in $\mathscr{A}$ as a special case of the MAPS where all the discrete points can be seen as targets for being observed by IoT devices. In such a case, the MAPS problem can be transformed into the MTPS problem. According to Theorem 6, we can verify that the MTPS is NP-hard. Therefore, the MAPS problem is also NP-hard. □

5. Algorithm for the MTPS Problem

According to the definition of the MTPS problem, we can obtain that the total energy consumption of the IoT devices depends on the number of IoT devices with their corresponding initial energy. Therefore, we need to deploy IoT devices as few as possible to cover targets and schedule the placement of IoT devices to minimize the total energy consumption such that every target in Q is continuously observed at T time slots by IoT devices. Based on these considerations, we can find that the MTPS problem consists of two subproblems, minimum location coverage (MLC) and minimum energy consumption scheduling deployment

(MESD), as shown in Definitions 8 and 9. In this section, we first propose an approximation algorithm for each of the problems MLC and MESD. Then, based on the problems MLC and MESD, we propose an approximation algorithm to solve the MTPS problem.

Definition 8 (MLC). Given a set $Q = \{r_1, r_2, \cdots, r_m\}$ of m targets, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites to be put in IoT devices with coverage range R, the minimum location coverage (MLC) problem is to find minimum subset $L' \subseteq L$ of candidate sites such that all targets are within the coverage range of candidate sites in L'.

Definition 9 (MESD). Given a set $S = \{s_1, s_2, \cdots, s_n\}$ of n IoT devices in which each IoT device $s_i \in S$ has active time E_i, a set $P = \{L_1, L_2, \cdots, L_K\}$ of K sites to be put IoT devices, a positive time T, the Minimum Energy consumption Scheduling Deployment (MESD) problem is to find a subset $C \subseteq S$ of IoT devices placed at the sites in P and to schedule the IoT devices in C such that

(1) for any site $L_k \in P$, it can be placed more than one IoT device from C

(2) for arbitrary site $L_k \in P$, the IoT devices located at L_k can cumulatively work at least T time

(3) for each IoT device $s_i \in C$, it continuous works E_i time once it starts working, and

(4) the total energy consumption of IoT devices in C, $M = \sum_{s_i \in C} E_i$, is minimized

5.1. Algorithm for the MLC Problem. In this subsection, we propose a greedy algorithm, called MLCA, to solve the MLC problem. Let U_k denote the set of targets within the coverage range of L_k. The MLCA algorithm consists of two steps. Firstly, for arbitrary $L_k \in L$, we compute its coverage set U_k. For any $r_j \in Q$, if $d_{L_k, r_j} \le R$, then, $U_k = U_k \cup \{r_j\}$. Secondly, we repeat the following steps until one of the conditions $L = \varnothing$ and $Q = \varnothing$ is satisfied.

(i) Select L_k with the maximum U_k from L

(ii) Execute the operations $L' = L' \cup \{L_k\}$, $L = L/\{L_k\}$ and $Q = Q/U_k$

(iii) For arbitrary $L_i \in L$, update its coverage set by deleting targets in $U_k \cap U_i$ from U_i

After executing the above algorithm, we can obtain a set $L' \subseteq L$ of candidate sites, which can cover all targets in Q. The pseudocode of the algorithm is shown in Algorithm 1. Then, we will analyze the performance of the MLCA algorithm.

Theorem 10. *Suppose that L^* is an optimal solution for the MLC problem. If there exists a solution for the MLC problem, then, we can verify that the approximation ratio of the MLCA algorithm is* $\ln m + 1$*, where m is the number of targets.*

```
Input: $Q = \{r_1, r_2, \cdots, r_m\}$, $L = \{L_1, L_2, \cdots, L_N\}$, $R$;
Output: $L'$;
Sets of $L' = \varnothing$, $U_k = \varnothing$ for any $L_k \in L$;
for arbitrary $L_k \in L$do
    for any $r_j \in Q$do
        if $d_{L_k, r_j} \le R$ then
            $U_k = U_k \cup \{r_j\}$;
        end
    end
end
while $L \neq \varnothing \&\& Q \neq \varnothing$ do
    Pick $L_k = \arg\ \max_{L_k \in L} U_k$;
    $L' = L' \cup \{L_k\}$, $L = L \setminus \{L_k\}$, $Q = Q \setminus U_k$;
    for any $L_i \in L$do
        $U_i = U_i \setminus U_k$;
    end
end
if $Q = \varnothing$ then
There is no solution for the MLC problem;
end
```

Algorithm 1: MLCA.

Proof. According to the MLCA algorithm, we can observe that the while loop terminates after at most m steps, since in each iteration of the while loop, there is at least one target that is covered by the candidate site L_k. Let Q_k denote the number of targets that are still not covered at iteration k of the while loop. In each iteration k, we can use all $|L^*|$ candidate sites in the optimal solution to cover all targets in Q. Therefore, there must exist a candidate site in L^* that covers at least $Q_k/|L^*|$ targets, which means at least $Q_k/|L^*|$ targets are covered in every iteration. In other words, we can obtain after iteration k; there are left at most $Q_k - Q_k/|L^*|$ targets that have not been covered by candidate sites, that is,

$$\begin{aligned} Q_{k+1} &\le \left(1 - \frac{1}{|L^*|}\right) \cdot Q_k \le \left(1 - \frac{1}{|L^*|}\right)^2 \cdot Q_{k-1} \le, \cdots, \\ &\le \left(1 - \frac{1}{|L^*|}\right)^{k+1} \cdot Q_0 = \left(1 - \frac{1}{|L^*|}\right)^{k+1} \cdot m, \end{aligned} \tag{13}$$

where the last equality depends on the fact that $Q_0 = m$, since all m tragets are not covered by candidate sites before the first iteration of the while loop. Notice that there exists $1 \le i \le m$ such that after executing i iterations of the while loop, $Q_i \le 1$. Based on the fact that $1 + x \le e^x$ for any $x \in (-\infty, +\infty)$, we have

$$\left(1 - \frac{1}{|L^*|}\right)^i = \left(\left(1 - \frac{1}{|L^*|}\right)^{|L^*|}\right)^{i/|L^*|} \le e^{-i/|L^*|}. \tag{14}$$

Based on inequations (13) and (14), we can obtain

$$m \cdot e^{-i/|L^*|} \le 1 \Leftrightarrow i \ge |L^*| \cdot \ln m. \tag{15}$$

Therefore, we can obtain that after $i = |L^*| \cdot \ln m$ iterations, the remaining number of targets in Q_i is smaller or equal to 1. Thus, the algorithm will terminate after at most $|L^*| \cdot \ln m + 1$ iterations, which can obtain $|L'| \leq |L^*| \cdot \ln m + 1 \leq (\ln m + 1) \cdot |L^*|$, since $|L^*| \geq 1$ and only one candidate site is added into L' in each iteration based on the algorithm MLCA. □

Theorem 11. *The time complexity of the MLCA algorithm is $O(mN)$, where m and N are the number of targets and the number of candidate sites, respectively.*

Proof. The MLCA algorithm consists of two phases. Firstly, the algorithm needs N iterations to compute the corresponding coverage sets for all candidate sites in L, as shown in the first while loop. In each iteration, since the number of targets is less than or equal to m, at most m steps are needed to determine which coverage set targets belong to. Therefore, we need at most mN steps to compute the coverage sets for all candidate sites in L. Secondly, at most $\min\{N, m\}$ iterations are needed in the while loop. In each iteration, we need to pick the candidate site L_k with the maximum U_k for executing N steps since $1 \leq k \leq N$. Then, we update all coverage sets of candidate sites with N steps. Therefore, we need at most $N \cdot \min\{N, m\}$ steps in the second while loop.

Consequently, the time complexity of the MLCA algorithm is $O(mN) = O(mN + N \cdot \min\{N, m\})$. □

5.2. Algorithm for the MESD Problem. In this subsection, we propose an approximation algorithm to solve the MESD problem, called MESDA. Before describing the algorithm, we introduce some notations. For any $1 \leq k \leq K$, we use C_k to denote the set of IoT devices placed at location L_k and let $\Phi = \{C_1, C_2, \cdots, C_K\}$. Let M_k represent the total energy consumption of the IoT devices in C_k.

The MESDA consists of two phases. The first phase is to find a subset $C_k \subseteq S$ of IoT devices from S for any $1 \leq k \leq K$ such that $\sum_{s_i \in C_k} E_i \geq T$. The second phase is to optimize M_k by replacing the high-energy-consuming IoT devices in C_k with low-energy-consuming ones from the remaining IoT devices in S for any $1 \leq k \leq K$. Afterwards, we compute $\Phi = \{C_1, C_2, \cdots, C_K\}$, $C = \bigcup_{1 \leq k \leq K} C_k$, and $M = \sum_{C_k \in \Phi} M_k$. The detailed description of the algorithm is shown as follows.

Initially, we set $\Phi = \varnothing$, $C = \varnothing$, $M = 0$, $C_k = \varnothing$, and $M_k = 0$ for each $1 \leq k \leq K$. The first phase of the MESD algorithm repeats the following four steps until the conditions $P = \varnothing$ and $S = \varnothing$ are satisfied.

(i) Select s_i with the maximum E_i from S, where if there exists two IoT devices $s_i, s_j \in S$ such that $E_i = E_j$, then, their maximum ID is selected

(ii) Pick L_k with the minimum M_k for any $1 \leq k \leq K$, where if there exist M_k and M_l such that $M_k = M_l$, then, their minimum ID is selected

(iii) Add s_i into C_k, and $M_k = M_k + E_i$. Then, $S = S/\{s_i\}$

(iv) Compare M_k with T. If $M_k \geq T$, then, $P = P/\{L_k\}$

After executing the first phase of the algorithm, we can obtain a set C_k of IoT devices for any $L_k \in P$, where the total working time M_k of IoT devices is greater than or equal to T. In the following, for any $1 \leq k \leq K$, we optimize M_k by replacing the high-energy-consuming IoT devices in C_k with low-energy-consuming ones in S.

The second phase of the algorithm repeats the following steps until $S = \varnothing$.

(i) Select s_i with the maximum E_i from S

(ii) Compute a tuple $< s_j, k \geq \arg \min_{s_j \in C_k, 1 \leq k \leq K} (M_k - E_j + E_i)$ such that $M_k - E_j + E_i \geq T$

(iii) Compare E_i with E_j. If $E_i \geq E_j$, then, s_i is deleted from S, otherwise, $C_k = C_k \cup \{s_j\}/\{s_i\}$, $M_k = M_k - E_i + E_j$, and s_i is removed from S

After executing the second phase of the algorithm, we can obtain a set C_k of IoT devices located on each $L_k \in P$ and the total energy consumption $M_k = \sum_{s_i \in C_k} E_i$ of IoT devices in C_k such that all targets covered by site L_k are continuously observed at least T time. Finally, we can obtain $\Phi = \{C_1, C_2, \cdots, C_K\}$, $M = \sum_{C_k \in \Phi} M_k$, and $C = \bigcup_{1 \leq k \leq K} C_k$. The pseudocode of the algorithm is shown in Algorithm 2.

We use C^* to represent the optimal set of IoT devices placed at sites in P for the MESD problem. Let M^* denote the total energy consumption of IoT devices in C^*. Without loss of generality, we use C_k^* to be the optimal set of IoT devices placed at $L_k \in P$ when C^* has been confirmed. Let M_k^* represent the total energy consumption of IoT devices in C_k^*.

Theorem 12. *If the MESDA algorithm has the feasible solution, then, we can verify that the approximation ratio of the algorithm is 2.*

Proof. According to the definition of the MESD problem, we have $M_k^* \geq T$ for any $1 \leq k \leq K$ and $M^* = \sum_{1 \leq k \leq K} M_k^* \geq K \cdot T$.

For any $1 \leq k \leq K$, we let $< s_i, E_i \geq \arg \max_{s_j \in C_k} E_j$, where C_k is obtained by the MESDA algorithm. We analyze the performance of the algorithm in the light of the following two cases.

(1) $0 < E_i < (T/2)$. Then, we can obtain that for any $s_j \in C_k$, $E_j < (T/2)$. Based on the algorithm, the last IoT device added into C_k makes M_k be greater than or equal to T, which means $M_k \leq T + E_j < (3T/2) \leq 1.5 M_k^*$

(2) $(T/2) \leq E_i < T$. If there exists $E_j \in C_k$ such that $(T/2) < E_j \leq E_i$, then based on the algorithm, we can obtain $C_k = \{s_i, s_j\}$ and $M_k = E_i + E_j < 2T < 2 M_k^*$. Otherwise, we can derive $M_k \leq T + E_j \leq 1.5 M_k^*$

```
Input: $S = \{s_1, s_2, \cdots, s_n\}$, $E_i$ for each $s_i \in S$, $P = \{L_1, L_2, \cdots, L_K\}$, $T$;
Output: $C$, $M$, $\Phi$;
Sets of $C = \emptyset$, $M = 0$, $C_k = \emptyset$ and $M_k = 0$ for each $1 \le k \le K$;
while $P \neq \emptyset \&\& S \neq \emptyset$ do
    Pick $s_i = \arg\ \max_{s_i \in S} E_i$;
    Pick $L_k = \arg\ \min_{L_k \in P} M_k$;
    $C_k = C_k \cup \{s_i\}$, $M_k = M_k + E_i$, $S = S \setminus \{s_i\}$;
    if $M_k \ge T$ then
        $P = P \setminus \{L_k\}$;
    end
end
while $S \neq \emptyset$ do
    Pick $s_i = \arg\ \max_{s_i \in S} E_i$;
    Select $<s_j, k> = \arg \min_{s_j \in C_k, 1 \le k \le K} (M_k - E_j + E_i)$ such that $M_k - E_j + E_i \ge T$;
    if $E_i \ge E_j$ then
        $S = S \setminus \{s_i\}$;
    else
        $C_k = C_k \cup \{s_j\} \setminus \{s_i\}$, $M_k = M_k - E_j + E_i$, $S = S \setminus \{s_i\}$;
    end
end
$\Phi = \bigcup_{1 \le k \le K} \{C_k\}$, $M = \sum_{C_k \in \Phi} M_k$, $C = \bigcup_{C_k \in \Phi} C_k$;
```

Algorithm 2: MESDA.

From what have been discussed, we can obtain $M_k < 2 M_k^*$. Therefore, we have $M = \sum_{1 \le k \le K} M_k \le \sum_{1 \le k \le K} 2 M_k^* \le 2 M^*$, which means that the approximation ratio of the MESDA algorithm is 2. □

Theorem 13. *The time complexity of the MESDA algorithm is $O(n^3 + nK)$, where n and K are the number of IoT devices and the number of sites, respectively.*

Proof. According to the MESDA algorithm, we can verify that the algorithm contains two while loops running and the other operations with constant running time. The first while loop consists of at most n iterations, since $|S| \le n$. In each iteration, at most n steps are needed to select the IoT device s_i with the maximum energy and K steps are required to compute the site L_k with the minimum total energy consumption of IoT devices located on L_k. The other operations in the iteration can be executed in constant time. Therefore, the first while loop runs at most $O(n^2 + nK)$ time. The second while loop runs $|S|$ iterations. In each iteration, the algorithm runs at most $|S|$ steps to select the IoT device s_i with the maximum energy. Then, it calculates the total energy consumption with at most $O(n \cdot |S|)$ running time by replacing each IoT device that has been deployed on site in P with s_i and selects s_j with the minimum $M_k - E_j + E_i$ among all IoT devices except IoT devices in S. The other operations in the iteration can be executed in constant time. Therefore, the total running time of the second while loop is at most $O(n^3) = O(n \cdot |S|^2 + |S|^2)$.

Consequently, we can obtain that the time complexity of the MESDA algorithm is $O(n^3 + nK) = O(n^2 + nK) + O(n^3)$. □

5.3. Algorithm for the MTPS Problem. In this subsection, we propose an approximation algorithm, called MTPSA, to solve the MTPS problem based on the MLC and MESD problems. The algorithm consists of two steps corresponding to Algorithms 1 and 2. The detailed illustration of the algorithm is shown in Algorithm 3.

Suppose that C_{opt}^* is an optimal subset of S for the MTPS problem. Let M_{opt}^* be the total energy consumption of IoT devices in C_{opt}^*. Since each $r_j \in Q$ needs to be continuously observed at least T time, we divide T into equal T time slots. We use c_t^* to represent the minimum energy consumption of all active IoT devices for the MTPS problem such that all targets are covered at the tth time slot, and let L_t^* denote the set of candidate sites placed in the active IoT devices in the tth time slot.

Lemma 14. *For any $1 \le t \le T$, we have $c_t^* \ge |L^*|$, where L^* is the minimum set of candidate sites for the MLC problem.*

Proof. In the MTPS problem, all targets need to be covered by IoT devices placed at candidate sites in L for any $1 \le t \le T$, which means that there exists a subset $L_t \subseteq L$ that can cover all targets for the arbitrary tth time slot. Thus, based on the definitions of MTPS problem and MLC problem, we can obtain that for any such subset L_t, it is a feasible solution for the MLC problem. Therefore, we have $|L_t^*| \ge |L^*|$,

```
Input: $Q = \{r_1, r_2, \cdots, r_m\} S = \{s_1, s_2, \cdots, s_n\}$, $E_i$ for each $s_i \in S$, $L = \{L_1, L_2, \cdots, L_N\}$, $R$, $T$;
Output: $C$, $M$, $\Phi$;
Step 1: Compute the set $L'$ of candidate sites to cover all targets in $Q$ by executing Algorithm 1;
Step 2: Compute $\Phi$, $C$, $M$ based on $L'$ by executing Algorithm 2,
where $P = L'$;
```

Algorithm 3: MTPSA.

since L_t^* is a feasible solution for the MLC problem. Therefore, we can derive $c_t^* \geq |L^*|$ since $c_t^* = |L_t^*|$. □

Lemma 15. *We can obtain $M_{opt}^* \geq |L^*| \cdot T$.*

Proof. According to Lemma 14, we can obtain $c_t^* \geq |L^*|$ for arbitrary $1 \leq t \leq T$. Based on the definition of the MTPS problem, we have $M_{\text{opt}}^* \geq \sum_{1 \leq t \leq T} c_t^* \geq |L^*| \cdot T$. □

Theorem 16. *The performance ratio of the MTPSA algorithm is 2 $\ln m + 2$, where m is the number of targets.*

Proof. According to the MTPSA algorithm, we have $M = \sum_{L_k \in L}{}' M_k$, where M is the total energy consumption of IoT devices obtained by the MTPSA algorithm and M_k is the energy consumption of IoT devices located on L_k. On the basis of Theorems 10 and 12, we can derive $|L'| \leq \ln m \cdot |L^*| + 1$ and $M_k < 2T$ for any $1 \leq k \leq |L'|$. Based on Lemma 15, we can obtain

$$\begin{aligned} M &= \sum_{L_k \in L'} M_k \leq \left|L'\right| \cdot \max\left\{M_k \mid L_k \in L'\right\} \\ &\leq (\ln m \cdot |L^*| + 1) \cdot 2T \leq 2(\ln m + 1) \cdot M_{\text{opt}}^*. \end{aligned} \tag{16}$$

The theorem has been proved. □

Theorem 17. *The time complexity of the MTPSA algorithm is $O(n^3 + mN + nN)$, where n is the number of available IoT devices in S, m denotes the number of targets in Q, and N represents the number of candidate sites in L.*

Proof. Based on Theorems 11 and 13, we can obtain that the time complexity of the MTPSA algorithm is $O(n^3 + mN + nN) = O(mN + n^3 + nK)$, since $K = |L'| \leq N$. □

6. Algorithm for the MAPS Problem

In this section, we propose an algorithm to solve the MAPS problem.

Similar to the definition of the MTPS problem, the total energy consumption of the IoT devices depends on the number of IoT devices with their corresponding initial energy. Therefore, we also need to deploy IoT devices as few as possible to cover the whole detection area and schedule the placement of IoT devices to minimize the total energy consumption such that the whole area is continuously observed T time slots by IoT devices. Therefore, we can find that the MAPS problem is also composed of two subproblems, minimum location area coverage (MLAC) and MESD, as shown in Definitions 18 and 9. The MESD problem has been solved by Algorithm 2. Thus, in this section, we first propose an approximation algorithm to solve the MLAC problem. Then, based on the problems MLAC and MESD, we propose an approximation algorithm to solve the MAPS problem.

We use $A(L_k)$ to denote the coverage region of L_k when the coverage range of IoT devices put on L_k is R, that is, for any point $p \in A(L_k)$, $d_{L_k,p} \leq R$. Let $D(L_k)$ be the border of $A(L_k)$.

Definition 18 (MLAC). Given a detection area $\mathscr{A}$, a set $L = \{L_1, L_2, \cdots, L_N\}$ of N candidate sites, $\Delta = \{A(L_1), A(L_2), \cdots, A(L_N)\}$, the minimum location area coverage (MLAC) problem is to find a minimum subset $L' \subseteq L$ of candidate sites such that $||\bigcup_{L_k \in L'} A(L_k) \bigcap \mathscr{A}|| = ||A||$.

In the following, we first introduce the definition of the Voronoi diagram of candidate sites on L, which is used to solve the MLAC problem.

6.1. Voronoi Diagram of Candidate Sites. The definition of the Voronoi diagram of candidate sites in L can be defined as the subdivision of the detection area $\mathscr{A}$ into N cells as shown in [31]. Any point $p \in \mathscr{A}$, in the cell corresponding to a candidate site L_k, is closer to L_k than to any other candidate site in L. Formally, the Voronoi cell corresponding to L_k can be defined as

$$\text{Cell}(L_k) = \bigcap_{k=1, j \neq k}^{n} \left\{p \mid d_{L_k,p} \leq d_{L_j,p}\right\}. \tag{17}$$

In equation (17), two Voronoi cells meet along a Voronoi edge and three cells meet at a Voronoi vertex. For simplicity, we use $G(VV, VE)$ to denote the Voronoi diagram of candidate sites in L, where VV represents the set of Voronoi vertices and VE denotes the set of Voronoi edges.

6.2. Algorithm for the MLAC Problem. In this subsection, we propose an approximation algorithm to solve the MLAC problem, which is called the MLAC Algorithm (MLACA).

Before describing the algorithm, we introduce some definitions and parameters as follows:

```
Input: The dimensions of $\mathscr{A}$, the coverage range $R$, $L = \{L_1, L_2, \cdots, L_N\}$;
Output: $L'$;
Sets of $L' = L$, $VV = \varnothing$, $VE = \varnothing$, $NV(L_k) = \varnothing$ for any $L_k \in L$;
Compute the Voronoi diagram $G(VV, VE)$ and $Reg(L')$ of candidate sites on $L'$ by using algorithm in [32];
for any $L_k \in L'$ do
    Compute $NV(L_k)$ based on $G(VV, VE)$ and $Reg(L')$;
    Construct $G(VV(L_k), NE(L_k))$ on $NV(L_k)$ by using algorithm in [32];
    Compute $NI(L_k)$ based on $G(VV(L_k), NE(L_k))$;
    if $VV(L_k) \cup NI(L_k) \subset \bigcup_{L_j \in NV(L_k)} A(L_j)$ then
        $L' = L' \setminus \{L_k\}$;
        Compute the Voronoi diagram $G(VV, VE)$ of candidate sites in $L'$ by using algorithm in [32];
    end
end for
```

Algorithm 4: MLACA.

```
Input: $S = \{s_1, s_2, \cdots, s_n\}$, $E_i$ for each $s_i \in S$, $L = \{L_1, L_2, \cdots, L_N\}$, $R$, $T$;
Output: $C$, $M$;
Step 1: Compute the set $L'$ of candidate sites to cover the whole area by executing Algorithm 4;
Step 2: Compute $C$, $M$ based on $L'$ by executing Algorithm 2,
where $P = L'$;
```

Algorithm 5: MAPSA.

Definition 19 (neighbor Voronoi diagram). The Neighbor Voronoi diagram of a candidate site L_k is the Voronoi diagram of the Voronoi neighbors of L_k when L_k is excluded, where we call a site L_j a Voronoi neighbor of L_k if their cells share an edge.

Definition 20 (redundant candidate site). A candidate site L_k is said to be a redundant candidate site if $||\bigcup_{L_j \in L/L_k} A(L_j) \bigcap \mathscr{A}|| = ||A||$.

Definition 21 (neighbor Voronoi vertices). The neighbor Voronoi vertices of a candidate site L_k are the Voronoi vertices of the neighbor Voronoi diagram of Voronoi neighbors of L_k.

Definition 22 (neighbor Voronoi intersection vertices). Neighbor Voronoi intersection vertices of L_k are the intersections between edges of the neighbor Voronoi diagram and the circumcircle $D(L_k)$ of $A(L_k)$.

We use $\text{reg}(L) = \{\text{cell}(L_1), \text{cell}(L_2), \cdots, \text{cell}(L_N)\}$ to represent the set of all cells for $G(VV, VE)$. Let $NV(L_k)$ be the set of Voronoi neighbors of L_k. We use $G(VV(L_k), NE(L_k))$ to represent the neighbor Voronoi diagram of Voronoi neighbors of L_k, where $VV(L_k)$ denotes the set of the neighbor Voronoi vertices and $NE(L_k)$ represents the set of the Voronoi edge of $G(VV(L_k), NE(L_k))$. We use $NI(L_k)$ to be the set of the neighbor Voronoi intersection vertices of L_k.

Based on Theorem 10 in the work [31], we can obtain the following lemma.

Lemma 23. *A candidate site L_k is a redundant candidate site if and only if all the neighbor Voronoi vertices and neighbor Voronoi intersection vertices of L_k are covered by the Voronoi neighbors of L_k.*

In the following, we will introduce the detailed description of the MLACA algorithm.

Initially, we set $L' = L$, $VV = \varnothing$, $VE = \varnothing$, and $NV(L_k) = \varnothing$ for any $L_k \in L$. The execution of the algorithm consists of three steps as follows.

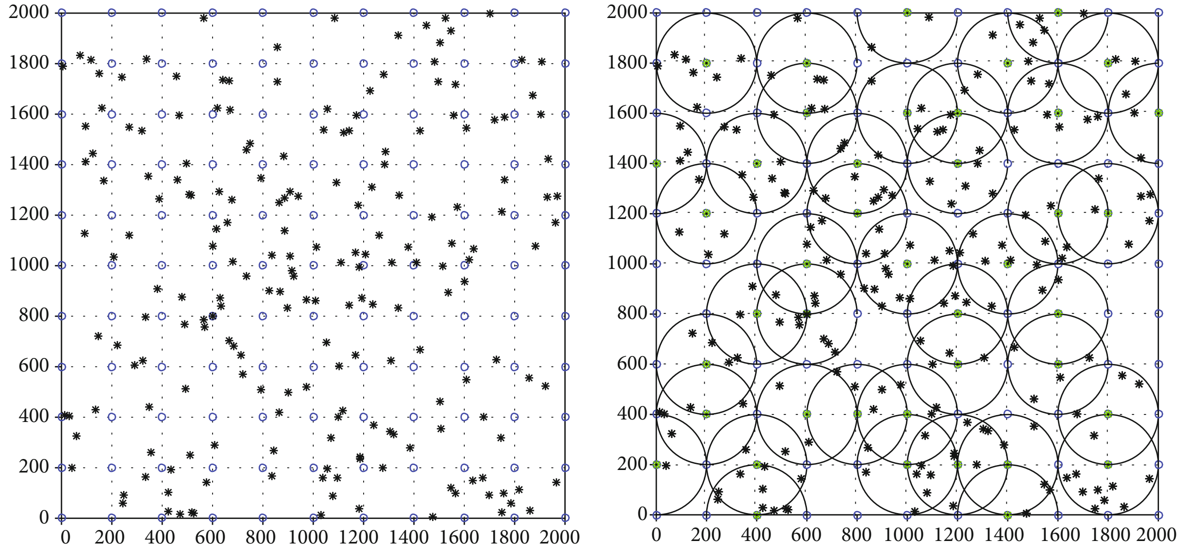
Firstly, we compute the Voronoi diagram $G(VV, VE)$ and $\text{reg}(L')$ of candidate sites in L' by using the algorithm in [32].

Secondly, we delete all redundant candidate sites from L'. For any $L_k \in L'$, we compute the set $NV(L_k)$ based on $G(VV, VE)$ and $\text{Reg}(L')$. Then, we construct the Voronoi diagram $G(VV(L_k), NE(L_k))$ on $NV(L_k)$ by using the algorithm in [32]. Next, we compute the set $NI(L_k)$ based on $G(VV(L_k), NE(L_k))$. Afterwards, we judge whether or not all points in $VV(L_k)$ and $NI(L_k)$ are covered by coverage regions of all Voronoi neighbors. If yes, we delete L_k from L' and update the Voronoi diagram $G(VV, VE)$ of candidate sites in new L' by using the algorithm in [32].

Finally, the algorithm returns L'.

The pseudocode of the algorithm is shown in Algorithm 4.

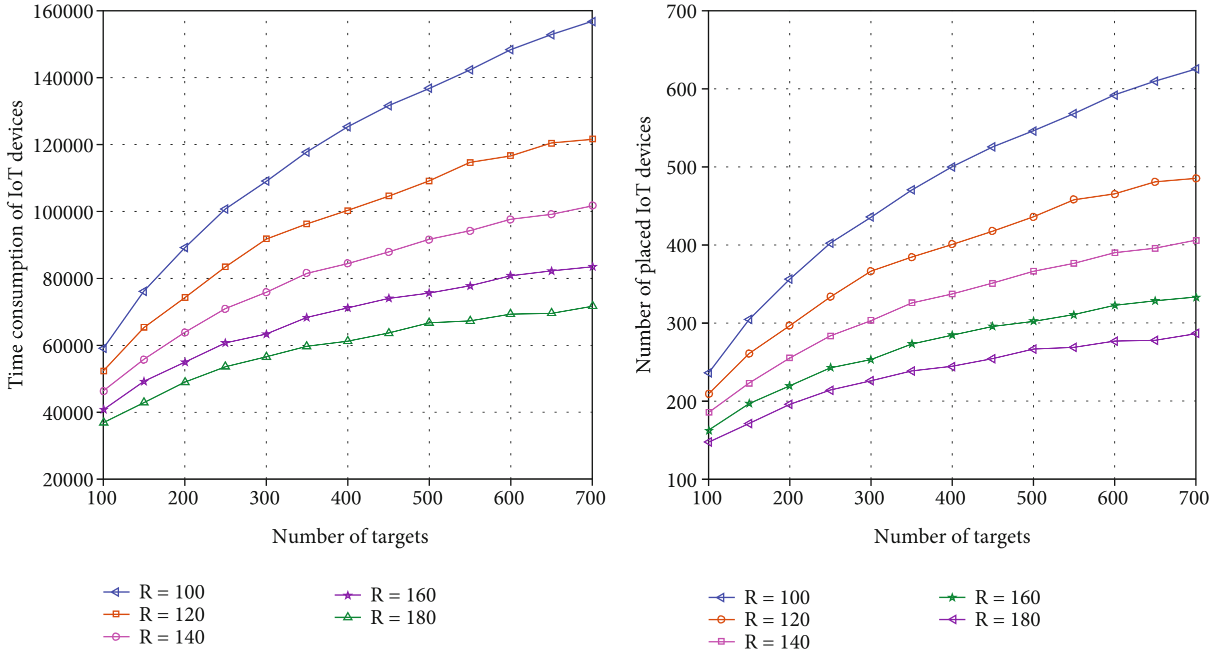
Theorem 24. *The time complexity of the MLACA algorithm is $O(N^2 \log N)$, where N represents the number of candidate sites in L.*



(a) The grid points as candidate sites

(b) The green points as placed sites

Figure 2: A coverage solution for an instance of the MTPS problem, where targets are deployed on the 2000 m × 2000 m detection area.



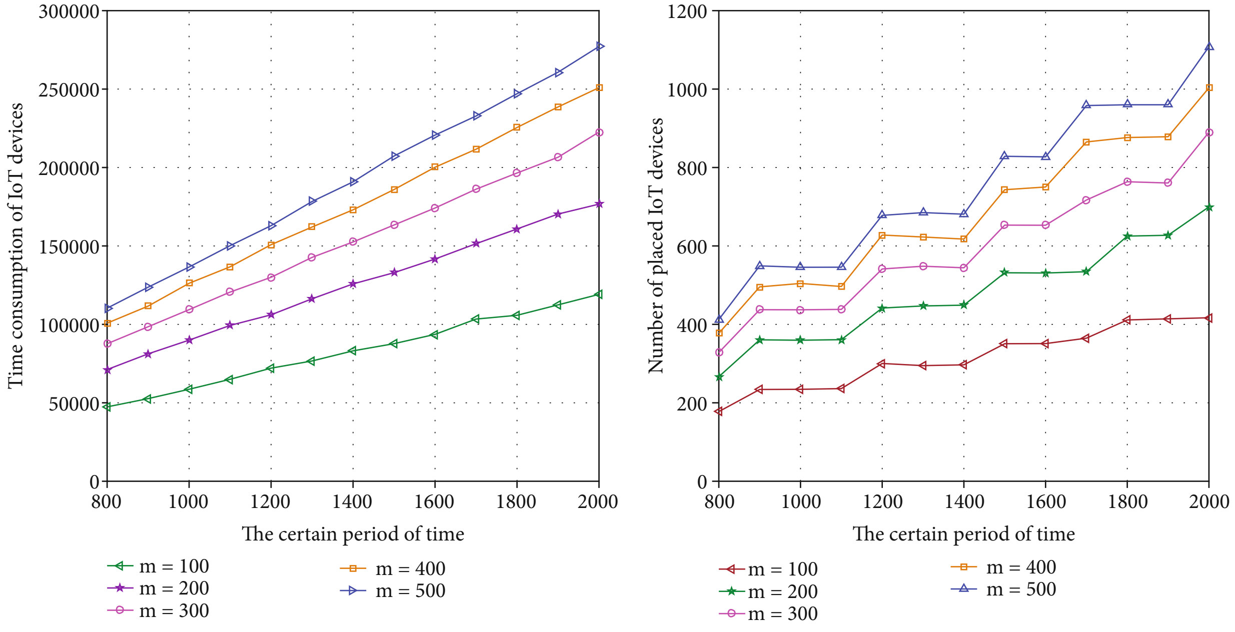
(a) Performance of MTPSA

(b) Effects of m and R

Figure 3: Simulations by varying m from 100 to 700 under different R.

Proof. According to the MLACA algorithm, we can find that the algorithm consists of three steps. Firstly, we compute the Voronoi diagram of candidate sites in L with $O(N \log N)$ time by using the algorithm in [32]. Secondly, we need at most N iterations in the for loop since $|L| = N$. In each iteration, we first need at most N times to compute the set $NV(L_k)$ of Voronoi neighbors of L_k. Then, we need at most $O(|NV(L_k)| \log |NV(L_k)|)$ time to compute the neighbor Voronoi diagram on $NV(L_k)$. Next, we need at most N times to compute $NI(L_k)$. We need at most another $O(N \log N)$ time to update the Voronoi diagram by using the algorithm in [32] when L_k is a redundant candidate site. Therefore, we need at most $O(N^2 \log N) = O(N * (N + |NV(L_k)| \log |NV(L_k)| + N \log N))$ time for the for loop since $|NV(L_k)| \leq N$.

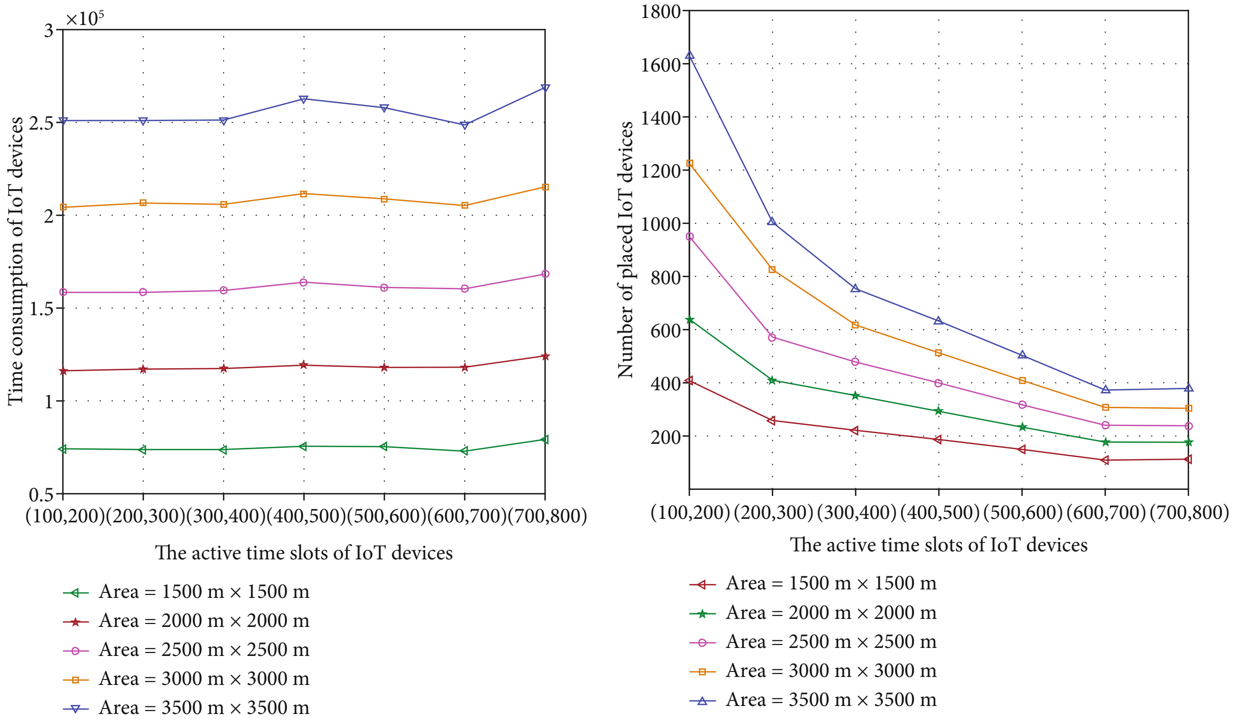
From what has been discussed, we can verify that the time complexity of the MLACA algorithm is $O(N^2 \log N)$. □



(a) Performance of MTPSA

(b) Effects of T and m

Figure 4: Results by varying T from 800 to 2000 under different m.

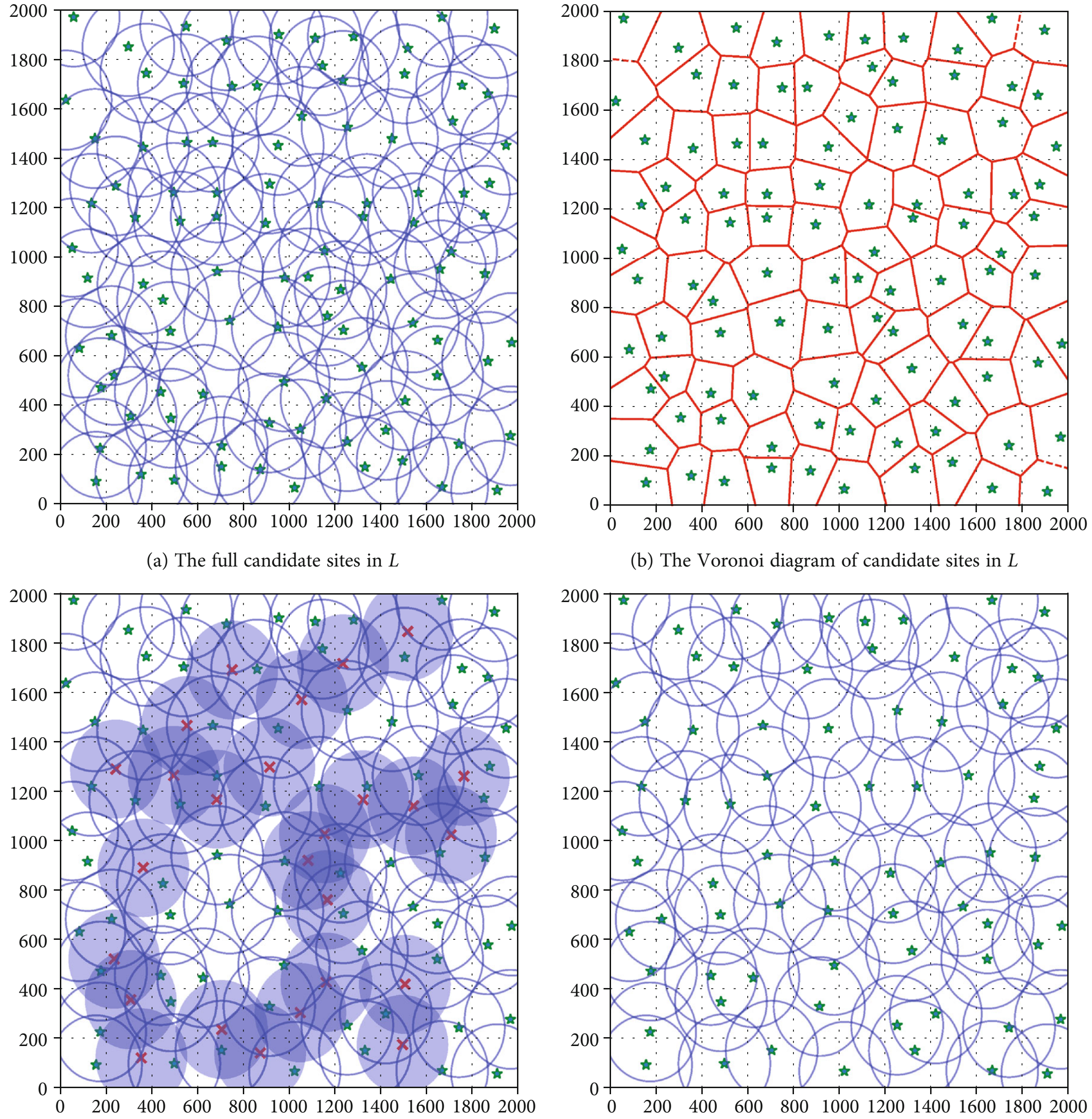


(a) Performance of MTPSA

(b) Effects of the size of the detection area and active time slots E_i

Figure 5: Simulations by varying E_i from [100, 200] to [700, 800] under different monitoring areas.

6.3. Algorithm for the MAPS Problem. In this subsection, we propose an approximation algorithm to solve the MAPS problem, which is called MAPSA. Similar to the MTPSA algorithm, the MAPSA algorithm consists of two phases. The first phase is to find a subset $L' \subseteq L$ of candidate sites by using Algorithm 4 such that the whole area $\mathscr{A}$ is covered by the coverage region of candidate sites in L'. The second phase is to find a subset $C \subseteq S$ of IoT devices placed at the sites in L' by executing Algorithm 2 such that for any point $p \in \mathscr{A}$, it is observed by IoT devices in C at least T time. The



(a) The full candidate sites in L

(b) The Voronoi diagram of candidate sites in L

(c) The redundant candidate sites identified by MLACA

(d) The placed sites of IoT devices obtained by MLACA

Figure 6: The executing process of the MLACA.

detailed illustration of the algorithm is shown in Algorithm 5.

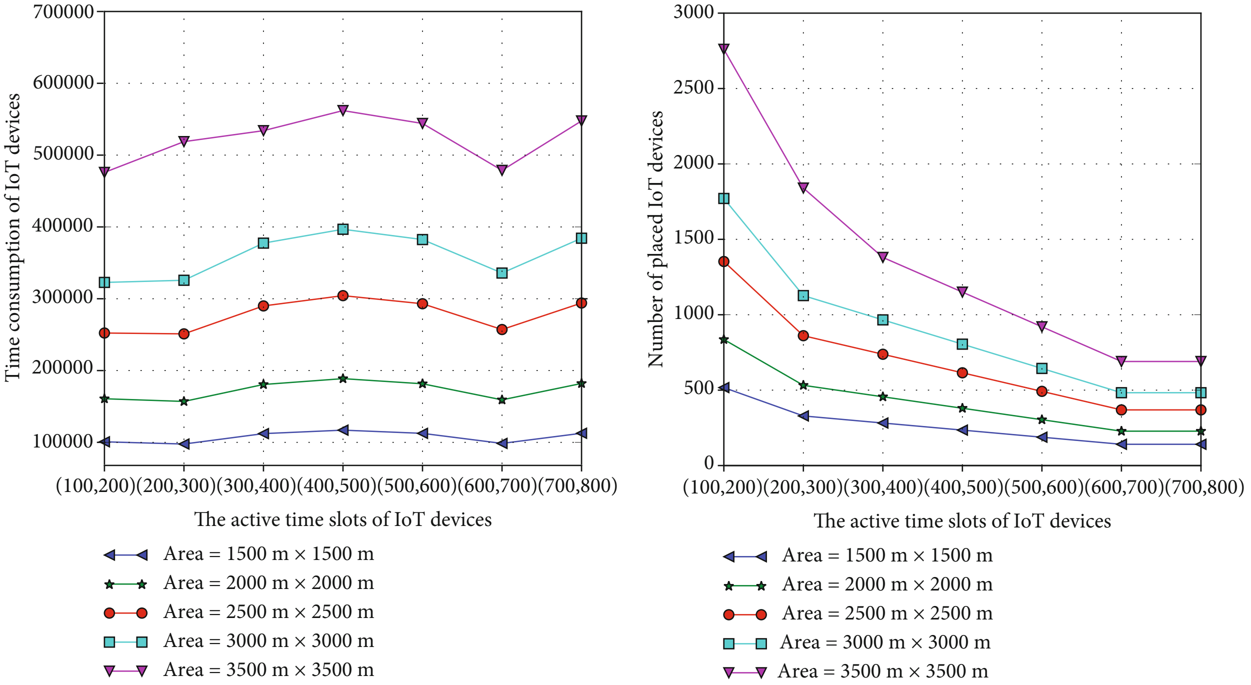
Theorem 25. *The time complexity of the MAPSA algorithm is* $O(n^3 + nN + N^2 \log N)$*, where n is the number of available IoT devices in S, and N represents the number of candidate sites in L.*

Proof. Based on Theorems 13 and 24, we can verify that the time complexity of the MAPSA algorithm is $O(n^3 + nN + N^2 \log N)$. □

7. Simulations

In this section, we evaluate the average performance of the approximation algorithms MTPSA and MAPSA depending on simulations with several critical performance metrics under different configurations. The code of the algorithms is implemented using MATLAB 2016a. For every group of parameter settings, we create 100 instances, execute the simulations, and obtain the average results.

7.1. Simulations for the MTPSA Algorithm. In order to ensure that all targets can be covered by candidate sites, we use a grid to cover the detection area where the distance



(a) Performance of MLACA

(b) Effects of active time slots E_i

FIGURE 7: Performance of MAPSA under different monitoring areas.

between neighbor grid points is R except for all grids in the boundary of the area. As an instance shown in Figure 2(a), all targets are randomly deployed on a $2000\,\text{m} \times 2000\,\text{m}$ detection area and the grid points as the candidate sites are located on the area, where we set $T = 1000$, $n = 2000$, and $R = 200$ m; pick the active time slots E_i of each IoT devices $s_i \in S$ from range [100, 200]. After executing the algorithm MTPSA for the instance, we can verify that the number of the placed sites of IoT devices is 42, as the green nodes shown in Figure 2(b); the number of IoT devices placed at the sites is 252 and the total time consumption of all used IoT devices is $4.2072e + 04$.

In the following, we will evaluate how the network settings, such as the number of targets m, the certain period of time T, the active time slots E_i for each IoT device $s_i \in S$, the coverage range R and the size of the detection area, affect the performance of the MTPSA algorithm.

Firstly, we evaluate how the number of targets m and the coverage range R affect the performance of the MTPSA algorithm when we set $n = 1000$ and $T = 1000$; use the interval [200, 300] to pick a uniformly distributed random active time slots E_i for each IoT device $s_i \in S$; change R to 100, 120, 140, 160, and 180 m; and vary m from 100 to 700 in the $2000\,\text{m} \times 2000\,\text{m}$ detection area, as shown in Figure 3. It is observed that the total time consumption of IoT devices increases with the increasing of m, as shown in Figure 3(a). This is because the number of IoT devices that are placed at candidate sites increases as m grows and the active time slots of IoT devices are fixed, as shown in Figure 3(b). Figure 3(a) also shows that the total time consumption of the used IoT devices decreases with increasing R since the less candidate sites are used to place IoT devices as R grows, which leads to a decline in the number of IoT devices being used, as shown in Figure 3(b).

Figure 4 illustrates the impact of the certain period of time T and the number of targets m on the performance of the MTPSA algorithm when we set $n = 2000$; pick the active time slots E_i of each IoT device $s_i \in S$ from [200, 300], $R = 100$ m, m=100, 200, 300, 400, and 500; and change T from 800 to 2000 in the $2000\,\text{m} \times 2000\,\text{m}$ detection area. Figure 4(a) shows that the total time consumption of used IoT devices is becoming larger with increasing T, since more IoT devices are needed to be placed on candidate sites for continuously observing targets as T increases when the active time slots of IoT devices are fixed, as shown in Figure 4(b). It is also observed that the performance gap is becoming smaller with the increasing of m. This is because the increase of the number of sites placing IoT devices is becoming smaller as more and more targets are randomly deployed on the detection area, which results in a reduction of the number of IoT devices placed at the sites, as shown in Figure 4(b). We also find that the number of used IoT devices increases as T grows since each IoT device $s_i \in S$ continuously works E_i time slots once it starts working.

Figure 5 evaluates the impact of the active time slots of IoT devices and the size of the detection area on the performance of the MTPSA algorithm when we set $n = 2000$, $R = 200$ m, $m = 600$, and $T = 2000$; deploy targets in the monitoring area $1500\,\text{m} \times 1500\,\text{m}$, $2000\,\text{m} \times 2000\,\text{m}$, $2500\,\text{m} \times 2500\,\text{m}$, $3000\,\text{m} \times 3000\,\text{m}$, and $3500\,\text{m} \times 3500\,\text{m}$; and assign the active time slots of each IoT device $s_i \in S$ from [100, 200], [200, 300], [300, 400], [400, 500], [500, 600], [600, 700], and [700, 800], respectively. Figure 5(a) shows that the total time consumption of used IoT devices levels off when $E_i < [700,800]$ with increasing of the active time slots of IoT devices and increases as the size of the monitoring area increases. This is because when $E_i < [700,800]$, at least three IoT devices should be placed at each placement site to continuously observe target T time, which is obviously greater than T.

7.2. Simulations for the MAPSA Algorithm. In order to ensure that the whole area can be covered by candidate sites, we select candidate sites evenly in the monitoring area. As an instance shown in Figure 6(a), given a 2000 m × 2000 m monitoring area, 100 candidate sites are located on the area and we set $R = 200$. The process of the MLACA algorithm is shown in Figures 6(b)–6(d). Firstly, we construct the Voronoi diagram of the candidate sites in L, as shown in Figure 6(b). Secondly, we find all redundant candidate sites from L such that all remaining candidate sites can cover the whole area, as the red points shown in Figure 6(c). Finally, we can obtain the placed sites of IoT devices, as shown in Figure 6(d).

In the following, we will evaluate how the network settings, the active time slots, and the size of the detection area affect the performance of the MAPSA.

Figure 7 evaluates the impact of the active time slots of IoT devices and the size of the detection area on the performance of the MAPSA algorithm when we set $n = 3000$, $R = 200$ m, and $T = 2000$; deploy targets in the monitoring area 1500 m × 1500 m, 2000 m × 2000 m, 2500 m × 2500 m, 3000 m × 3000 m, and 3500 m × 3500 m; and assign the active time slots of each IoT device $s_i \in S$ from [100, 200], [200, 300], [300, 400], [400, 500], [500, 600], [600, 700], and [700, 800], respectively. Figure 7(a) shows that the total time consumption of used IoT devices increases with the increase of the active time slots of IoT devices when $E_i < [400,500]$. This is because when $E_i < [400,500]$, at least five IoT devices should be placed at each placement site to continuously observe the area T time and their total energy consumption increases with E_i increasing. When $E_i > [400,500]$, the total time consumption decreases with the increasing of E_i, since as E_i grows, the less devices are needed for working a certain time. Figure 7(b) shows that the number of used devices decreases with the increasing of E_i for each device. We also find that the number of placed devices decreases as the size of area decreases.

8. Conclusion

In this paper, we investigate the problems minimum energy consumption of IoT devices for target coverage through placement and scheduling (MTPS) and minimum energy consumption of IoT devices for area coverage through placement and scheduling (MAPS), which focuses on finding the placement locations of IoT devices from candidate sites and scheduling them to cover all targets or the whole monitoring area such that all targets or the entire area are (or is) continuously observed for a certain period of time and the total energy consumption of the placed IoT devices is minimized. We first propose the mathematical models for the proposed problems and prove that they are NP-hard. Then, we propose an approximation algorithm for each of them. Finally, extensive simulation results are shown to further verify the performance of the proposed algorithm.

Data Availability

The data used to support the findings of this study are included within the article.

Disclosure

A preliminary version of this paper appeared in the WASA 2021.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported in part by the Fundamental Research Funds for the Central Universities under grants 2021ZY88 and BLX201921 and also supported by the National Natural Science Foundation of China under Grant 62002022.

References

[1] T. Zhu, T. Shi, J. Li, Z. Cai, and X. Zhou, “Task scheduling in deadline-aware mobile edge computing systems,” *IEEE Internet of Things Journal*, vol. 6, no. 3, pp. 4854–4866, 2018.

[2] Z. Cai and X. Zheng, “A private and efficient mechanism for data uploading in smart cyber-physical systems,” *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 2, pp. 766–775, 2018.

[3] D. Ismail, M. Rahman, and A. Saifullah, “Low-power wide-area networks: opportunities, challenges, and directions,” in *Proceedings of the Workshop Program of the 19th International Conference on Distributed Computing and Networking*, New York, NY, USA, 2018.

[4] A. M. Yousuf, E. M. Rochester, B. Ousat, and M. Ghaderi, “Throughput, coverage and scalability of lora lpwan for Internet of things,” in *2018 IEEE/ACM 26th International Symposium on Quality of Service (IWQoS)*, Banff, AB, Canada, 2018.

[5] B. Buurman, J. Kamruzzaman, G. Karmakar, and S. Islam, “Low-power wide-area networks: design goals, architecture, suitability to use cases and research challenges,” *IEEE Access*, vol. 8, pp. 17179–17220, 2020.

[6] C. Luo, J. Yu, D. Li, H. Chen, Y. Hong, and L. Ni, “A novel distributed algorithm for constructing virtual backbones in wireless sensor networks,” *Computer Networks*, vol. 146, no. DEC.9, pp. 104–114, 2018.

[7] Q. Chen, Z. Cai, L. Cheng, and H. Gao, “Structure-free broadcast scheduling for duty-cycled multihop wireless sensor networks,” *IEEE Transactions on Mobile Computing*, p. 1, 2021.

[8] C. Luo, M. N. Satpute, D. Li, Y. Wang, W. Chen, and W. Wu, “Fine-grained trajectory optimization of multiple uavs for efficient data gathering from wsns,” *IEEE/ACM Transactions on Networking*, vol. 29, no. 1, pp. 162–175, 2021.

[9] Z. Cai and Z. He, “Trading private range counting over big Iot data,” in *2019 IEEE 39th International Conference on Distributed Computing Systems (ICDCS)*, pp. 144–153, Dallas, TX, USA, 2019.

[10] C. Luo, Y. Hong, D. Li, Y. Wang, W. Chen, and Q. Hu, “Maximizing network lifetime using coverage sets scheduling in wireless sensor networks,” *Ad Hoc Networks*, vol. 98, p. 102037, 2020.

[11] Y. Wang, S. Wu, Z. Chen, X. Gao, and G. Chen, “Coverage problem with uncertain properties in wireless sensor networks: a survey,” *Computer Networks*, vol. 123, no. aug. 4, pp. 200–232, 2017.

[12] M. Cardei, M. T. Thai, Y. Li, and W. Wu, “Energy-efficient target coverage in wireless sensor networks,” in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, Miami, FL, USA, 2005.

[13] R. Akhlaghinia, S. Hashemi, and B. Shadgar, “Sensor Placement for Heterogenous Point Coverage,” in *2010 Second International Conference on Computer and Network Technology*, Bangkok, Thailand, 2010.

[14] S. Mini, S. K. Udgata, and S. L. Sabat, “Sensor deployment and scheduling for target coverage problem in wireless sensor networks,” *IEEE Sensors Journal*, vol. 14, no. 3, pp. 636–644, 2014.

[15] N. T. Hanh, H. T. T. Binh, N. X. Hoai, and M. S. Palaniswami, “An efficient genetic algorithm for maximizing area coverage in wireless sensor networks,” *Information Sciences*, vol. 488, pp. 58–75, 2019.

[16] İ. K. Altınel, N. Aras, E. Güney, and C. Ersoy, “Binary integer programming formulation and heuristics for differentiated coverage in heterogeneous sensor networks,” *Computer Networks*, vol. 52, no. 12, pp. 2419–2431, 2008.

[17] B. Wang, “Coverage problems in sensor networks,” *ACM Computing Surveys*, vol. 43, no. 4, pp. 1–53, 2011.

[18] I. Gravalos, P. Makris, K. Christodoulopoulos, and E. A. Varvarigos, “Efficient gateways placement for Internet of things with qos constraints,” in *2016 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, Washington, DC, USA, 2016.

[19] C. Jiang, Z. Chen, R. Su, and Y. C. Soh, “Group greedy method for sensor placement,” *IEEE Transactions on Signal Processing*, vol. 67, no. 9, pp. 2249–2262, 2019.

[20] M. Z. Hasan and H. Al-Rizzo, “Optimization of sensor deployment for industrial Internet of things using a multiswarm algorithm,” *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10344–10362, 2019.

[21] J. A. Jiang, J. C. Wang, H. S. Wu et al., “A novel sensor placement strategy for an iot-based power grid monitoring system,” *IEEE Internet of Things Journal*, vol. 7, no. 8, pp. 7773–7782, 2020.

[22] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, MIT Press, 2nd edition, 2001.

[23] P. Berman, G. Calinescu, C. Shah, and A. Zelikovsky, “Efficient energy management in sensor networks,” *Ad Hoc and Sensor Networks, Wireless Networks and Mobile Computing*, vol. 2, pp. 71–90, 2005.

[24] L. Ding, W. Wu, J. Willson, L. Wu, and W. Lee, “Constant-approximation for target coverage problem in wireless sensor networks,” in *Proceedings IEEE Infocom*, pp. 1584–1592, Orlando, FL, USA, 2012.

[25] Z. Lu, W. W. Li, and M. Pan, “Maximum lifetime scheduling for target coverage and data collection in wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 714–727, 2015.

[26] T. Shi, J. Li, H. Gao, and Z. Cai, “A novel framework for the coverage problem in battery-free wireless sensor networks,” *IEEE Transactions on Mobile Computing*, p. 1, 2020.

[27] G. Xing, C. Lu, R. Pless, and J. A. O’Sullivan, “Co-grid: an efficient coverage maintenance protocol for distributed sensor networks,” in *Proceedings of the 3rd international symposium on Information processing in sensor networks*, pp. 414–423, New York, NY, USA, 2004.

[28] J. Yu, S. Wan, X. Cheng, and D. Yu, “Coverage contribution area based k-coverage for wireless sensor networks,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 9, pp. 8510–8523, 2017.

[29] N.-N. Qin and J.-L. Chen, “An area coverage algorithm for wireless sensor networks based on differential evolution,” *International Journal of Distributed Sensor Networks*, vol. 14, no. 8, 2018.

[30] U. Feige, “A threshold of lnnfor approximating set cover,” *Journal of the ACM*, vol. 45, no. 4, pp. 634–652, 1998.

[31] A. Boukerche and X. Fei, “A voronoi approach for coverage protocols in wireless sensor networks,” in *IEEE GLOBECOM 2007- IEEE Global Telecommunications Conference*, pp. 5190–5194, Washington, DC, USA, 2007.

[32] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, vol. 501, John Wiley & Sons, 2009.