Movement-efficient Sensor Deployment in Wireless Sensor Networks with Limited Communication Range

Jun Guo, *Student Member, IEEE*, and Hamid Jafarkhani, *Fellow, IEEE*

Abstract

We study a mobile wireless sensor network (MWSN) consisting of multiple mobile sensors or robots. Three key factors in MWSNs, sensing quality, energy consumption, and connectivity, have attracted plenty of attention, but the interaction of these factors is not well studied. To take all the three factors into consideration, we model the sensor deployment problem as a constrained optimization problem. Our goal is to find an optimal sensor deployment (or relocation) to optimize the sensing quality with a limited communication range and a specific network lifetime constraint. We derive necessary conditions for the optimal sensor deployment in both homogeneous and heterogeneous MWSNs. According to our derivation, some sensors are idle in the optimal deployment of heterogeneous MWSNs. Using these necessary conditions, we design both centralized and distributed algorithms to provide a flexible and explicit trade-off between sensing uncertainty and network lifetime. The proposed algorithms are successfully extended to more applications, such as area coverage and target coverage, via properly selected density functions. Simulation results show that our algorithms outperform the existing relocation algorithms.

Index Terms

Sensor deployment, coverage, heterogeneous, mobile wireless sensor networks, network lifetime.

I. Introduction

Mobile wireless sensor networks (MWSNs) are widely employed in agricultural, industrial and military applications. This is due to tremendous technical developments in wireless communications, computation power, data processing and storage. Agriculture gets numerous benefits from MWSNs by successfully assigning mobile sensors to observe the surrounding environments, for example temperature, humidity, and illumination. In industry, MWSNs can be used to monitor material strength, water leak, and toxic gases. Besides, MWSNs are also good tools to detect

The authors are with Center for Pervasive Communications and Computing, University of California, Irvine (e-mail: guoj4@uci.edu; hamidj@uci.edu). This work was supported in part by the NSF Award CCF-1815339. The paper was presented in part at ICC-18 [1].

intruders. According to sensors' functionality/capacity, MWSNs can be classified into homogeneous MWSNs and heterogeneous MWSNs. In homogeneous MWSNs, extensively studied in the literature [2]–[31], sensors share the same capacity, e.g., storage, computation power, sensitiveness, communication radius, coverage radius, moving efficiency, and battery. However, the heterogeneous WSNs consist of sensors with different capacities [32]–[37].

One major challenge in both homogeneous and heterogeneous MWSNs is the deployment of nodes to optimize the performance of the network. To evaluate the sensing quality, the binary coverage model, in which each sensor can only cover a disk with radius R_s, is widely used in MWSNs [2]–[12], [32], [33], [35]. There have been many coverage models and deployment algorithms, for different sensing tasks, in the literature; look at [2]–[4] and the references therein. Four popular coverage categories are (i) area coverage, (ii) target coverage, (iii) barrier coverage, and (iv) sensing uncertainty. A natural sensing task is to maximize the area coverage, which is formulated by the total area covered by sensors. In another popular coverage model, target coverage, the specific target locations are detected and reported by relocated sensors. In this case, sensors or robots are required to collect detailed information from the discrete targets. A full-target-coverage is achieved if and only if every discrete target in the 2-dimensional region is covered by at least one sensor. Sensors in barrier coverage model move along the boundary to detect intruders as they cross the border of a region or domain. To obtain full-barrier-coverage, one should place sensors to cover the whole barrier or boundary. Finally, the minimization of sensing uncertainty requires sensors to form a Centroidal Voronoi Tessellation (CVT). It is mainly used when there is no specific target. The widely used sensing uncertainty model (see more details in Section II) can be presented as a quantizer with the sensing uncertainty as its distortion [2]–[10], [14]–[17], [32]–[34]. In fact, MWSNs should be reconfigurable/flexible to support different sensing tasks. However, to the best of our knowledge, there is no unified framework that models a variety of coverage tasks. In this paper, we will propose a unified relocation model which can be applied to the above four coverage tasks.

Connectivity is another important requirement in MWSNs. In MWSNs, mobile sensor nodes are relocated to collect physical information, such as magnetism, temperature, and voice. The collected data is forwarded to the outside world through access points (APs). Therefore, the collected data is useless if it cannot be forwarded to the AP via single-hop or multiple-hop communications. When sensors are connected by wirelines, the connectivity is guaranteed automatically. But, the connectivity is still a challenge in MWSNs where sensors are communicating

with each other through wireless channels. A common communication model [3], [11]–[13] assumes that each sensor node is able to communicate with sensors in a limited communication range R_c.

Energy efficiency is another key issue in MWSNs, as most sensors have limited battery energy, and it is inconvenient or even infeasible to replenish the batteries of numerous densely deployed sensors [18]. In general, the energy consumption of a device includes communication energy, data processing energy [38], sensing energy, and movement energy. In fact, sensor movement has a much higher energy consumption compared to other types of energy [19], [20], and then dominates the energy consumption. Guiling et al. [39] study the optimal angular velocity and the optimal acceleration to minimize the energy consumption for motion. Simulation results in [39] show that the energy consumption for one-step motion with the optimal angular velocity setting is approximately linear to the movement distance. In fact, the linear movement energy consumption is a popular assumption and widely adopted in the literature [21]–[23], [36], [40]. Total energy consumption and network lifetime are two common energy-related measures. But, network lifetime maximization, which balances the energy consumption among sensors, is a more common and challenging problem because it takes into account the available battery power of sensors.

A. Related Work

A huge body of literature exists on energy-efficient sensor relocation. However, most of the papers in the literature consider one or two key metrics rather than all the three (sensing quality, connectivity, and energy efficiency) together. Moreover, there is no unified framework that can support different coverage models.

References [21]–[23], [35], [36] study the energy saving with a full-area-coverage guarantee. Hungarian Algorithm is applied to minimize the total energy consumption after the full-area-coverage is achieved by Genetic Algorithm [21]. Note that the above method put sensing quality as the first priority, and the energy efficiency is merely the secondary objective. To provide a flexible and fair trade-off between area coverage and energy consumption, virtual force based algorithms, HEAL [22], VFA [23], [35], and DSSA [36], are proposed. Besides, a variant of VFA is designed in [35] to maximize the area coverage in a heterogeneous MWSN. However, connectivity is not considered in [22], [23], [35], [36].

Sensor relocation for target coverage and barrier coverage is also well studied. Rout et al. [24] design a virtual-force based algorithm, OATIDA, to obtain both full-target-coverage and

full-connectivity on a region with obstacles, while energy consumption is ignored. Chen et al. [25] propose a two-phase algorithm to achieve full-target-coverage with minimum total energy consumption, but connectivity is not taken in to account. Similarly, Njoya et al. [26] design an evolutionary-based framework to make the trade-off between target coverage and network lifetime while the connectivity issue is ignored. Liao et al. [27] investigate how to deploy mobile sensors with minimum total energy consumption to form a MWSN that provides both full-target-coverage and full-connectivity. Although all three factors are considered in [27], full-target-coverage and full-connectivity are implemented sequentially, which requires redundant sensors. On the other hand, the existing literature on barrier coverage also seeks the perfect sensing quality, i.e., full-barrier-coverage. Chen et al. [28] focus on 1-dimensional barriers, and then provide an energy-efficient relocation plan to obtain full-barrier-coverage. In [29], a greedy algorithm with binary search is applied to achieve maximum network lifetime and 2-dimensional full-barrier-coverage simultaneously. A faster algorithm which achieves the same purpose as [29] is provided in [30]. Still, the above sensor relocation algorithms designed for barrier coverage ignore the connectivity requirement.

Sensor relocation for sensing uncertainty (or CVT model) has also been investigated in the literature. Li et al. [9] explore directional sensors whose sensing uncertainty varies among different directions, and then design two iterative algorithms to optimize the sensor deployment. But, energy consumption is not taken into their objective function. Taking both connectivity and sensing uncertainty into account, our previous work [33] proposes the necessary conditions for the optimal sensor relocation in heterogeneous MWSNs. Unfortunately, another key metrics, energy consumption, is not taken into consideration. In [10], the authors propose two algorithms, Lloyd-α and DEED, to minimize sensing uncertainty with a movement related penalty function. Note that one has to manually adjust the parameter α (or δ) for Lloyd-α (or DEED) to satisfy a specific energy constraint. To overcome this weakness, two Lloyd-like algorithms without any intermediate parameter are proposed in the conference version of this paper [1]. These two algorithms can be employed to minimize sensing uncertainty with a total energy constraint or a network lifetime constraint. However, our previous work [1] does not consider the connectivity requirement.

B. Our Contributions

In this paper, we study the sensor relocation problem in MWSNs and make the following contributions: (1) A unified optimization framework for different coverage models is provided that

takes three key metrics, sensing quality, connectivity, and energy consumption, into consideration. (2) By providing analytical necessary conditions, we design centralized and distributed Lloyd-like algorithms to optimize sensor relocation with (i) network lifetime constraints and (ii) limited communication ranges.

The rest of the paper is organized as follows. We first introduce the system model and formulate the problems of sensing, energy consumption, and connectivity in Section II. In Section III, we discuss centralized sensor deployments for MWSNs considering the network lifetime and communication range. In Section IV, we propose a distributed algorithm to relocate sensors such that the required network lifetime and full-connectivity are fulfilled during the relocation. The algorithm complexity and communication overhead are analyzed in Section V. After that, we extend the proposed algorithms to other self-deployment scenarios in Section VI. Finally, we present numerical simulations in Section VII and conclude our work in Section VIII.

II. System model

Let $\Omega \in \Re^2$ be a convex target region including its interior. Given N sensors in the target area Ω, sensor deployment before and after the relocation are, respectively, defined by $\mathbf{P}^0 = (p_1^0, \ldots, p_N^0) \subset \Omega^N$ and $\mathbf{P} = (p_1, \ldots, p_N) \subset \Omega^N$, where p_n^0 is Sensor n's initial location and p_n is Sensor n's final location. Let $\mathcal{I}_\Omega = \{1, \ldots, N\}$ be the set of sensors in the MWSN. A cell partition $\mathbf{R}(\mathbf{P})$ is defined as a collection of N disjoint subsets, $\{R_n(\mathbf{P})\}_{n\in\mathcal{I}_\Omega}$, whose union is Ω. We assume that Sensor n only monitors the events that occurred in its cell partition $R_n(\mathbf{P})$, $\forall n \in \mathcal{I}_\Omega$. Let $\|\cdot\|$ denote the Euclidean distance, $card(\mathcal{A})$ be the number of elements in set $\mathcal{A}$, ∂W be the boundary of a region $W \subset \Omega$, and $\mathbb{B}(c, r) = \{\omega |\ \|\omega - c\| \leq r\}$ be a ball centered at c with radius r.

In the binary disk communication model [2]–[9], [20]–[22], two sensor nodes can establish reliable communications within one hop if and only if the distance between the two is smaller than R_c, where R_c is referred to as the communication range. We define the access point (AP) as the sensor node that can communicate with the outside information world. Without loss of generality, we assume that Sensor 1 acts as the AP. Other sensor nodes can transfer data outside if and only if there exist paths from the sensors to the AP. Each path consists of a sequence of sensor nodes where each hop distance is smaller than R_c. Sensor nodes that are connected to the AP via one-hop or multi-hop communications construct the backbone network. Let $\mathcal{S}(\mathbf{P})$ be the backbone network when the sensor deployment is $\mathbf{P}$. The sensors in $\mathcal{S}(\mathbf{P})$ are referred to as active sensors while the sensors outside of $\mathcal{S}(\mathbf{P})$ are referred to as inactive sensors. Accordingly,

we define the active sensor deployment, $\mathcal{H}(\mathbf{P})$, as the vector of locations of active sensors. In particular, we have $\mathbf{P} = \mathcal{H}(\mathbf{P})$ and $card(\mathcal{S}(\mathbf{P})) = n$ when $\mathcal{S}(\mathbf{P})$ includes all sensor nodes. If all sensors are included in the backbone network, we call the network fully connected. Otherwise, the network is divided into several disconnected sub-graphs. For convenience, we assume that the initial sensor deployment constructs a fully connected network, i.e., $\mathbf{P}^0 = \mathcal{H}(\mathbf{P}^0)$.

To evaluate the sensing uncertainty in heterogeneous MWSNs, we consider the distortion function [32]–[34] defined by

$$D(\mathbf{P}) = \sum_{n=1}^{N} \int_{R_n(\mathbf{P})} \eta_n \|p_n - \omega\|^2 f(\omega) d\omega, \tag{1}$$

where the sensing cost parameters $\eta_n \in (0, 1]$ are constants that depend on Sensor n's characteristics, e.g. sensitivity, and $f(\omega) : \Omega \to \Re^+$ is a spatial density function that reflects the frequency of random events taking place over the target region. In homogeneous MWSNs, sensors have identical parameters, i.e., $\eta_n = \eta, \forall n \in I_\Omega$. Note that the sensing uncertainty is only determined by the final deployment $\mathbf{P}$.

However, as explained previously, when the communication range R_c is limited, some sensor nodes cannot transfer their data back to the AP. As a result, only the sensor nodes in the backbone network can contribute to the sensing and therefore the performance should be revised as

$$D(\mathbf{P}) = \sum_{n \in \mathcal{S}(\mathbf{P})} \int_{R_n(\mathcal{H}(\mathbf{P}))} \eta_n \|p_n - \omega\|^2 f(\omega) d\omega, \tag{2}$$

The optimal partition for the performance function (2) is Multiplicatively Weighted Voronoi Diagram (MWVD) [33], which can be applied to both homogeneous and heterogeneous MWSNs. The MWVD of Ω generated by $\mathbf{P}$ is the collection of sets $\{V_n(\mathbf{P})\}_{n \in I_\Omega}$ defined by

$$V_n(\mathbf{P}) = \{\omega \in \Omega | \eta_n \|\omega - p_n\|^2 \leq \eta_m \|\omega - p_m\|^2, \forall m \in I_\Omega\}. \tag{3}$$

In particular, the MWVD for homogeneous MWSNs degenerates to the Voronoi Diagram [32]. From now on, we use $\mathbf{V}(\mathbf{P}) = \{V_n(\mathbf{P})\}_{n \in \mathcal{I}_\Omega}$ to replace partition $\mathbf{R}(\mathbf{P}) = \{R_n(\mathbf{P})\}_{n \in \mathcal{I}_\Omega}$. Placing (3) back to (2), we can rewrite the distortion as

$$D(\mathbf{P}) = \sum_{n \in \mathcal{S}(\mathbf{P})} \int_{V_n(\mathcal{H}(\mathbf{P}))} \eta_n \|p_n - \omega\|^2 f(\omega) d\omega. \tag{4}$$

The same distortion can also be applied to formulate the communication energy consumption among densely deployed sensors where the $f(\cdot)$ presents the sensor density function [18].

Next, we review a classic energy consumption model for the mobile sensor networks. Since the

sensor movement dominates the power consumption, we only consider the power consumption for sensor movement. As we mentioned in Section I, the energy consumption for one-step movement is linearly related to the moving distance. Therefore, the energy consumption for Sensor n moving from a to b can be defined [21]–[23], [36], [39], [40] as

$$\mathscr{E}_n(a,b) = \xi_n \|b-a\|, \tag{5}$$

where the moving cost parameter ξ_n is a predetermined constant that depends on Sensor n's energy efficiency.

III. Centralized sensor deployment with a network lifetime constraint

In a centralized sensor deployment scenario, a fusion center or base station collects global information (all sensor locations and parameters) and then computes and determines the final destinations for the sensors. After receiving the decisions from the fusion center, sensors move to their final destinations directly. It is self-evident that this point-to-point relocation is the most efficient strategy in terms of energy consumption.

A. Problem formulation

Since sensors move to their final destinations directly, the energy consumption for Sensor n is formulated as

$$E_n(\mathbf{P}) = \mathscr{E}_n(p_n^0, p_n) = \xi_n \|p_n - p_n^0\|, \tag{6}$$

where p_n is Sensor n's final destination. Our main goal is minimizing the sensing uncertainty defined by (4) given a constraint on the network lifetime T. To guarantee a required network lifetime, each sensor should be assigned an energy threshold (or maximum movement distance) for relocation [28]–[31]. Therefore, the corresponding constrained optimization problem, which is referred to as Problem $\mathcal{A}$, is

$$\underset{\mathbf{P}}{\text{minimize}} \qquad D(\mathbf{P}) \tag{7}$$

$$\text{s.t.} \quad E_n(\mathbf{P}) \le \gamma_n, n \in I_\Omega, \tag{8}$$

where γ_n is the maximum energy consumption of Sensor n. Let e_n be the battery energy of Sensor n at the initial time and β (watt) be Sensor n's power consumption (which is dominated by communication, sensing, and computation) after the relocation. To ensure the network lifetime, T, we have $\min_n (e_n - E_n(\mathbf{P})) \ge \beta T$, and thus $\gamma_n = e_n - \beta T, n \in I_\Omega$.

B. The Optimal Sensor Deployment

Lemma 1. *Given a fully connected initial deployment, i.e.,* $\mathbf{P}^0 = \mathcal{H}(\mathbf{P}^0)$*, the optimal deployment* $\mathbf{P}^*$ *for Problem* $\mathcal{A}$ *in a homogeneous MWSN is also fully connected, i.e.,* $\mathbf{P}^* = \mathcal{H}(\mathbf{P}^*)$.

The proof is provided in Appendix A.

According to Lemma 1, homogeneous networks should keep connectivity after optimal sensor movements as long as the initial deployment is fully connected. To analyze the network connectivity, we introduce two important concepts: desired region (DR) and feasible region (FR). Let $\mathcal{I} \subseteq \mathcal{I}_\Omega$ be an arbitrary sensor set. For convenience, the sensors in $\mathcal{I}$ and $\mathcal{I}_\Omega - \mathcal{I}$ are referred to as internal and external sensors, respectively. For each sensor, n, the set of all locations of n that result in a connected $\mathcal{I}$ is called the DR of n. An internal sensor's DR for sensor set $\mathcal{I}$ is defined as the region in which if the sensor is placed, the sensors in $\mathcal{I}$ are connected. As a special case, if an internal sensor's DR is empty, the sensors in $\mathcal{I}$ cannot construct a connected network. Without the internal sensor, $n \in \mathcal{I}$, the rest of the internal sensors, $\mathcal{I} - \{n\}$, consists of $K_n(\mathbf{P}, \mathcal{I})$ disjoint components: $U_{n1}(\mathbf{P}, \mathcal{I}), U_{n2}(\mathbf{P}, \mathcal{I}), \cdots, U_{nK_n(\mathbf{P},\mathcal{I})}(\mathbf{P}, \mathcal{I})$, where the sensors in each component are connected and $\bigcup_{k=1}^{K_n(\mathbf{P},\mathcal{I})} U_{nk}(\mathbf{P}, \mathcal{I}) = \mathcal{I} - \{n\}$. The internal sensors are connected if and only if Sensor n connects to all $\{U_{nk}(\mathbf{P}, \mathcal{I})\}$s. Thus, internal sensors' DRs for set $\mathcal{I}$ are formulated as

$$\mathbb{D}_n(\mathbf{P}, \mathcal{I}) = \bigcap_{k=1}^{K_n(\mathbf{P},\mathcal{I})} \left[\bigcup_{j \in U_{nk}(\mathbf{P},\mathcal{I})} \mathbb{B}\left(p_j, R_c\right) \right], \forall n \in \mathcal{I}. \tag{9}$$

Although we represent DRs as functions of $\mathbf{P}$ for convenience, Sensor n's DR is in fact determined by all sensors except itself. For an internal sensor $n \in \mathcal{I}$, the condition $p_n \in \mathbb{D}_n(\mathbf{P}, \mathcal{I})$[1] guarantees that all internal sensors can communicate with each other. In particular, if the AP is included in $\mathcal{I}$, we have $\mathcal{I} \subseteq \mathcal{S}(\mathbf{P})$. In addition, it is trivial to show that for two sensors $m, n \in \mathcal{I}$, $p_m \in \mathbb{D}_m(\mathbf{P}, \mathcal{I})$ is equivalent to $p_n \in \mathbb{D}_n(\mathbf{P}, \mathcal{I})$.

An example for 12 sensors with $R_c = 1$ is illustrated in Fig. 1a. The internal sensor set $\mathcal{I}$ is defined as all sensors, i.e., $\mathcal{I} = \mathcal{I}_\Omega = \{1, \ldots, 12\}$. Consider $n = 1$, to calculate $\mathbb{D}_1\left(\mathbf{P}, \mathcal{I}_\Omega\right)$, the rest of the sensors are divided into $K_1 = 2$ components $U_{11} = \{2, 3, 4, 5, 6, 7\}$ and $U_{12} = \{8, 9, 10, 11, 12\}$. According to the definition of DR, the green overlap between the cyan region $\left[\bigcup_{j=2}^{7} \mathbb{B}\left(p_j, R_c\right)\right]$ and the yellow region $\left[\bigcup_{j=8}^{12} \mathbb{B}\left(p_j, R_c\right)\right]$ in Fig. 1a constructs Sensor 1's DR, $\mathbb{D}_1\left(\mathbf{P}, \mathcal{I}_\Omega\right)$. Obviously, if p_1 is placed within $\mathbb{D}_1\left(\mathbf{P}, \mathcal{I}_\Omega\right)$, all 12 sensors can communicate with each other. However, if the internal sensor set $\mathcal{I}$ is defined as $\{1, 4, 5, 6, 9, 10, 11\}$, the corresponding DR for Sensor 1 will be empty, indicating that the sensors $\{1, 4, 5, 6, 9, 10, 11\}$ cannot construct a connected network.

Next, we define $\mathcal{W}\left(\mathbf{P}, \mathcal{I}\right) \triangleq \bigcup_{n \in \mathcal{I}_\Omega - \mathcal{I}} \mathbb{B}\left(p_n, R_c\right)$. It is self-evident that the internal sensors

[1]Remark: When Sensor n is placed in its DR for $\mathcal{I}$, some external sensors, $m \in \mathcal{I}_\Omega - \mathcal{I}$, may also connect to internal sensors.

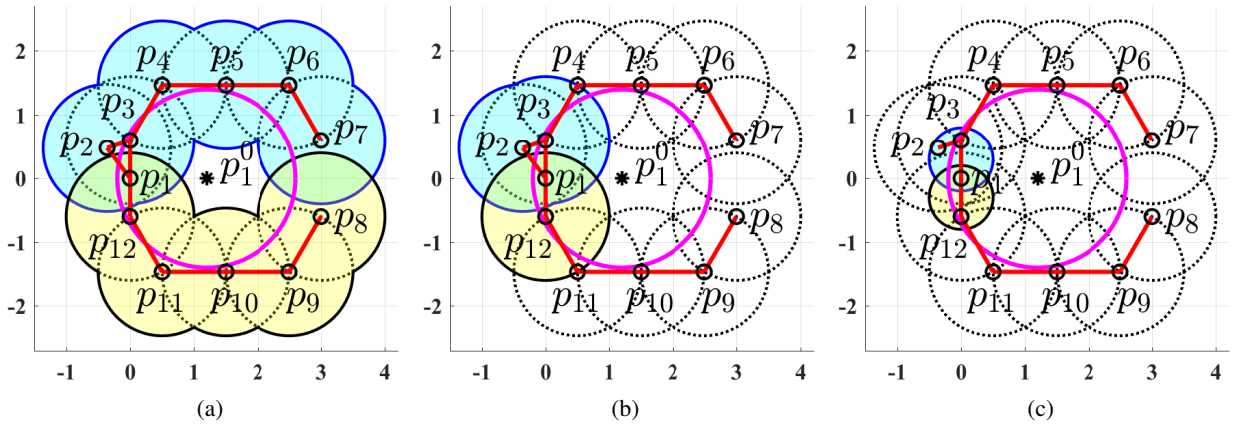


Fig. 1: Example1: (a) DR and FR for Sensor 1; (b) ADR and AFR for Sensor 1; (c) SDR and SFR for Sensor 1; DR, ADR, and SDR are shown by green. FR, AFR, and SFR are shown by the intersections of green regions and the magenta circles. Communication ranges, movement range, and Connections are, respectively, denoted by black doted curves, magenta solid curve, and red lines.

placed in $\mathcal{W}(\mathbf{P},\mathcal{I})$ connect to at least one external sensor. As a result, the sensor set $\mathcal{I}$ is the exact backbone network if and only if $1 \in \mathcal{I}$ and $p_n \in \mathbb{D}_n(\mathbf{P},\mathcal{I}) \bigcap \mathcal{W}^c(\mathbf{P},\mathcal{I})$, where $\mathcal{W}^c(\mathbf{P},\mathcal{I}) = \Omega - \mathcal{W}(\mathbf{P},\mathcal{I})$ is the complement of $\mathcal{W}(\mathbf{P},\mathcal{I})$. In what follows, we take the energy constraints (8) into account, and propose the concept of FR defined by

$$\mathbb{F}_n(\mathbf{P},\mathcal{I}) \triangleq \mathbb{D}_n(\mathbf{P},\mathcal{I}) \bigcap \mathbb{B}\left(p_n^0, \frac{\gamma_n}{\xi_n}\right), n \in \mathcal{I}, \tag{10}$$

where the energy constraint, $\xi_n \|p_n - p_n^0\| \leq \gamma_n$, is satisfied by the condition $p_n \in \mathbb{B}\left(p_n^0, \frac{\gamma_n}{\xi_n}\right)$.

The example of FR for Sensor 1 is illustrated in Fig. 1a. The magenta circle demonstrates Sensor 1's movement range $\mathbb{B}\left(p_1^0, \frac{\gamma_n}{\xi_n}\right)$. Then, the intersection of green regions and the magenta circle in Fig. 1a is Sensor 1's FR, $\mathbb{F}_1(\mathbf{P},\mathcal{I}_\Omega)$. Obviously, if p_1 is placed within $\mathbb{F}_1(\mathbf{P},\mathcal{I}_\Omega)$, we have (a) all 12 sensors can communicate with each other and (b) Sensor 1's energy constraint is also satisfied.

Note that $p_n \in \mathbb{F}_n(\mathbf{P},\mathcal{I})$ implicitly implies that $\mathbb{F}_n(\mathbf{P},\mathcal{I}) \neq \emptyset$. Accordingly, the set of deployments that not only constructs the backbone network $\mathcal{S}(\mathbf{P}) = \mathcal{I}$ but also satisfies the energy constraints can be formulated as

$$\Gamma(\mathcal{I}) = \{\mathbf{P} | 1 \in \mathcal{I}, p_n \in \mathbb{F}_n(\mathbf{P},\mathcal{I}) \bigcap \mathcal{W}^c(\mathbf{P},\mathcal{I}), \forall n \in \mathcal{I}\}. \tag{11}$$

Based on the aforementioned concepts, we propose the following necessary condition for the optimal deployment.

Theorem 1. *Let* $\mathbf{P}^* = (p_1^*, \ldots, p_N^*)$ *be the optimal deployment for Problem* $\mathcal{A}$*. The necessary conditions for the optimal deployment are*

(i) $c_n(\mathbf{P}^*) \notin \left[\mathbb{F}_n(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*)) \bigcap \mathcal{W}(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*))\right], \forall n \in \mathcal{S}(\mathbf{P}^*)$

$$\textit{(ii)} p_n^* = \begin{cases} c_n(\mathbf{P}^*), & \textit{if } c_n(\mathbf{P}^*) \in \mathbb{F}_n(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*)) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*)) \\ \arg \min\limits_{q \in \partial \mathbb{F}_n(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*))} \|q - c_n(\mathbf{P}^*)\|, & \textit{if } c_n(\mathbf{P}^*) \in \Omega - \mathbb{F}_n(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*)) \end{cases}, \forall n \in \mathcal{S}(\mathbf{P}^*)$$

The proof is provided in Appendix B.

According to Theorem 1, if n is a sensor in the backbone network $\mathcal{S}(\mathbf{P}^*)$, its optimal location, p_n^*, is either at the centroid $c_n(\mathbf{P}^*)$ or on the boundary of $\mathbb{F}_n(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*))$. Note that Theorem 1 only provides the necessary conditions for the sensors in the backbone network $\mathcal{S}(\mathbf{P}^*)$ because sensors that are not in the backbone network make no contribution to the distortion (4). In particular, by Lemma 1, all sensors in homogeneous MWSNs should be included in the backbone network, i.e., $\mathcal{S}(\mathbf{P}^*) = \mathcal{I}_\Omega$, and therefore the necessary conditions in Theorem 1 can be extended to all sensors in homogeneous MWSNs. Since there are no inactive sensors in homogeneous MWSNs, we have $\mathcal{W}(\mathbf{P}^*, \mathcal{S}(\mathbf{P}^*)) = \emptyset$. Then, the necessary conditions for the optimal deployment in homogeneous MWSNs can be refined as

$$p_n^* = \begin{cases} c_n(\mathbf{P}^*), & \text{if } c_n(\mathbf{P}^*) \in \mathbb{F}_n(\mathbf{P}^*, \mathcal{I}_\Omega) \\ \arg \min\limits_{q \in \partial \mathbb{F}_n(\mathbf{P}^*, \mathcal{I}_\Omega)} \|q - c_n(\mathbf{P}^*)\|, & \text{if } c_n(\mathbf{P}^*) \notin \mathbb{F}_n(\mathbf{P}^*, \mathcal{I}_\Omega) \end{cases}, \forall n \in \mathcal{I}_\Omega \tag{12}$$

With the help of the necessary conditions in Theorem 1, we design centralized Lloyd-like algorithms to find the optimal sensor deployment with a network lifetime constraint in the next subsection.

C. Centralized Lloyd-like Algorithms

To optimize the sensor deployments in homogeneous and heterogeneous MWSNs, we propose two centralized Lloyd-like algorithms: Centralized Constrained Movement Lloyd (CCML) Algorithm and Backward-stepwise Centralized Constrained Movement Lloyd (BCCML) Algorithm. CCML Algorithm, which is designed for homogeneous MWSNs, keeps all sensors in the backbone network. Based on CCML Algorithm, BCCML Algorithm recursively selects the optimal sensor set to construct the backbone network for heterogeneous MWSNs.

1) CCML Algorithm: According to our analysis in Section III-B, Sensor n's movement should be restrained within its desired region, $\mathbb{D}_n(\mathbf{P}, \mathcal{I}_\Omega)$, in order to keep full-connectivity, i.e., $\mathcal{I} = \mathcal{I}_\Omega$. Since the desired region is primarily influenced by the neighboring sensor nodes, we can

approximate it by

$$\tilde{\mathbb{D}}_n(\mathbf{P},\mathcal{I}_\Omega)=\bigcap_{k=1}^{K_n(\mathbf{P},\mathcal{I})}\left[\bigcup_{j\in U_{nk}(\mathbf{P},\mathcal{I}_\Omega)\bigcap\mathcal{N}_n(\mathbf{P})}\mathbb{B}\left(p_j,R_c\right)\right], \tag{13}$$

where $\mathcal{N}_n(\mathbf{P})$ is the set of Sensor n's neighbors. Then, FR is approximated by

$$\tilde{\mathbb{F}}_n(\mathbf{P},\mathcal{I}_\Omega)=\tilde{\mathbb{D}}_n(\mathbf{P},\mathcal{I}_\Omega)\bigcap\mathbb{B}\left(p_n^0,\frac{\gamma_n}{\xi_n}\right) \tag{14}$$

Note that the approximation in (14) can be calculated locally, but to calculate the exact feasible region, one needs global information. The above two approximations are referred to as approximated desired region (ADR) and approximated feasible region (AFR). The examples of ADR, and AFR for Sensor 1 are illustrated in Fig. 1b. Different from the DR shown in Fig. 1a, the ADR only considers Sensor 1's neighbors, $\mathcal{N}_1=\{2,3,12\}$. Thus, the green overlap between cyan region $\left[\bigcup_{j=2}^{3}\mathbb{B}\left(p_j,R_c\right)\right]$ and yellow region $\mathbb{B}\left(p_{12},R_c\right)$ in Fig. 1b construct Sensor 1's ADR, $\tilde{\mathbb{D}}_1\left(\mathbf{P},\mathcal{I}_\Omega\right)$. Then, the intersection of the green region and the magenta circle in Fig. 1b is Sensor 1's AFR, $\tilde{\mathbb{F}}_1\left(\mathbf{P},\mathcal{I}_\Omega\right)$. Note that Sensor 1's FR in Fig. 1a consists of two disconnected regions while Sensor 1's AFR in Fig. 1b is a connected region.

Now, we provide the details of CCML Algorithm. Like Lloyd Algorithm, the proposed algorithm iterates between two steps: (1) Partition optimization: Partitioning is done by MWVDs; (2) Location optimization: each sensor moves to the closest point to its centroid $c_n(\mathbf{P})$ within $\tilde{\mathbb{F}}_n(\mathbf{P},\mathcal{I}_\Omega)$. More details about CCML Algorithm are shown in Algorithm 1.

Algorithm 1 Centralized Constrained Movement Lloyd Algorithm

Input: Target area Ω; Probability density function $f(\cdot)$; the initial sensor deployment $\mathbf{P}^0$; the required network lifetime T; the stop threshold ϵ; the communication range R_c.
Output: Sensors deployment $\mathbf{P}$; Distortion $D(\mathbf{P})$.

```
Calculate the energy constraints {γ_n}_{n∈I_Ω} in terms of T
Initialize sensor deployment P = P^0
do
    Calculate the old distortion D_old = D(P)
    Do multiplicatively weighted Voronoi partition
    for n = 1 to N do
        Calculate the feasible region {F_n(P, I_Ω)}
        Calculate the critical point q, closest point to c_n(P) within {F_n(P, I_Ω)}
        Update sensor deployment p_n = q
        Calculate the new distortion D_new = D(P)
    end for
while (D_old − D_new)/D_old > ε
```

Theorem 2. *CCML Algorithm is an iterative improvement algorithm, i.e., the distortion decreases at each iteration and converges.*

Proof: CCML Algorithm is an iterative improvement algorithm only if both steps in CCML Algorithm do not increase the distortion (4) subject to the constraints (8). In Section II, we have proved that MWVD is the optimal cell partition for a given deployment. Therefore, Step (1) of CCML will not increase the distortion. During Step (2) of CCML, the cell partition is fixed as MWVD. In Appendix A, we show that Sensor n's optimal location should minimize its distance to the centroid $c_n(\mathbf{P})$ when the cell partition is fixed. In addition, by the analysis in Section III-B, Sensor n's movement should be restricted in $\tilde{\mathbb{F}}_n(\mathbf{P}, \mathcal{I}_\Omega)$ in order to guarantee both (i) energy constraints (8) and (ii) full-connectivity which has been proved (in Lemma 1) as a necessary condition for the optimum solution. Accordingly, Step (2) of CCML will not increase the distortion. Therefore, CCML Algorithm is an iterative improvement algorithm. Furthermoer, the distortion has a lower bound 0. As a result, the distortion of CCML Algorithm is non-increasing with a lower bound, indicating that the distortion converges. ■

2) BCCML Algorithm: Sensors with low-battery energy have small energy to spend on motion, which results in small movement ranges. To keep the connection with a low-battery node, e.g., Sensor n, the neighboring sensors' movements will be restricted by the limited communication range of Sensor n, even if Sensor n's neighbors have access to large battery energy. In this case, if Sensor n is not used in the MWSN, the neighboring sensors will have more freedom to move and probably further decrease the overall distortion. Given the current sensor set $\mathcal{I}$, sensors that decrease the distortion when removed from $\mathcal{I}$ are referred to as bottleneck sensors. To select the optimal sensor set as our backbone network, BCCML Algorithm starts with all sensors and repeatedly eliminates the least significant bottleneck sensor in terms of reducing the distortion until no bottleneck sensor is left. Intuitively, a bottleneck sensor n should satisfy the following conditions: (i) After eliminating n, the rest of sensors in $\mathcal{I}$ should be connected, i.e., $K_n(\mathbf{P}, \mathcal{I}) = 1$ and $U_{n1}(\mathbf{P}, \mathcal{I}) = \mathcal{I} - \{n\}$. In other words, Sensor n is a leaf node in the network. Otherwise, the network will be divided into multiple sub-graphs after eliminating n, and then fewer sensors will be used in the sensing task. (ii) Sensor n should already run out of its movement energy, i.e., $\xi_n \|p_n - p_n^0\| = \gamma_n$. (iii) At least one of its neighbors has redundant energy, i.e., $\exists m \in \mathcal{N}_n, \xi_m \|p_m - p_m^0\| < \gamma_m$. The above three conditions are referred to as the bottleneck criterion. To speed up the computation, BCCML Algorithm merely eliminates bottleneck sensors satisfying the bottleneck criterion. The details of BCCML Algorithm are shown in Algorithm 2.

Algorithm 2 Backwards-stepwise Centralized Constrained Movement Lloyd Algorithm

Input: Target area Ω; probability density function $f(\cdot)$; the initial sensor deployment $\mathbf{P}^0$; the required network lifetime T; the stop threshold ϵ; the communication range R_c.
Output: Sensors deployment $\mathbf{P}$; Distortion $D(\mathbf{P})$.

Calculate the energy constraints $\{\gamma_n\}_{n\in\mathcal{I}_\Omega}$ in terms of T
Initialize sensor set $\mathcal{I} = \mathcal{I}_\Omega$
Run CCML: $[\mathbf{P},\ D(\mathbf{P})] = \text{CCML}(\Omega, \mathbf{P}^0, R_c, \{\gamma_n\}_{n\in\mathcal{I}_\Omega}, f(\cdot), \epsilon, \mathcal{I})$
for k = 1 to $N-1$ **do**
    Identify bottleneck sensor set $\mathcal{I}_b$ by checking the bottleneck criterion
    $LSS = null$
    for $i \in \mathcal{I}_b$ **do**
        Generate a temporary sensor set $\hat{\mathcal{I}} = \mathcal{I} - i$
        Run CCML: $[\hat{\mathbf{P}}, D(\hat{\mathbf{P}})] = \text{CCML}(\Omega, \mathbf{P}^0, R_c, \{\gamma_n\}_{n\in\mathcal{I}}, f(\cdot), \epsilon, \hat{\mathcal{I}})$
        if $D(\hat{\mathbf{P}}) < D(\mathbf{P})$ **then**
            Update the least significant sensor (LSS): $LSS = i$, $\mathbf{P} = \hat{\mathbf{P}}$
        end if
    end for
    if $LSS \neq null$ **then**
        Eliminate the least significant sensor: $\mathcal{I} = \mathcal{I} - LSS$
    else
        break
    end if
end for

IV. Distributed sensor deployment with a network lifetime constraint

A. Problem formulation

In the distributed scenario, there is no fusion center, and sensors determine their own destinations. In general, sensors are supposed to only collect neighboring information (the locations and parameters of its neighbors and itself). As we discussed in Section III, the most energy-efficient relocation strategy is moving a sensor from its initial location to the final destination in one step. Unfortunately, this one-step relocation strategy requires global information and cannot be implemented in a distributed scenario. In fact, the distributed sensor relocation methods in the literature are categorized into continuous and discrete time systems [3]–[10], [14]–[17], [33]. In the continuous time systems[2], sensors keep communicating with their neighbors during

[2]Remark: The energy formulation (5) works for a one-step movement where the optimal velocity and acceleration is determined by the distance [39]. However, the movement in continuous time systems is not step-wise and the corresponding motion energy is a function of velocity [14]. Thus, the energy model in this paper cannot be applied to continuous time systems. The continuous sensor relocation in MWSNs is an interesting future work.

continuous movements. Then, dynamic systems are widely used to control sensors's first order dynamics, velocity, and/or second order dynamics, acceleration [7], [8], [14]. However, in the discrete time system, sensors only communicate with their neighbors at some discrete time instances, and their relocation is divided into multiple steps [10]. Regarding the discrete nature of the relocation, the sensors should be synchronized with each other in some fashion. Some iterative algorithms, such as Lloyd-like algorithms and virtual-force based algorithms, have been applied to this scenario [9], [10], [15]–[17], [33], [34]. To reduce the communication costs during the relocation, we use a discrete time system to control sensors' movements.

In what follows, we concentrate on the sensor relocation with multiple stops. The sensor deployment at the k-th stop is defined by $\mathbf{P}^k = (p_1^k, \ldots, p_N^k) \subset \Omega^N$, where p_n^k is Sensor n's location at the k-th stop. Let K be the maximum number of stops (iterations) for each sensor. For convenience, each sensor is extended to have K stops. For a sensor with J physical stops, e.g. Sensor n, its redundant stops are extended as $p_n^k = p_n^J, \forall k \in \{J+1, \ldots, K\}$. In particular, $\mathbf{P}^0 = (p_1^0, \ldots, p_N^0) \subset \Omega^N$ and $\mathbf{P}^K = (p_1^K, \ldots, p_N^K) \subset \Omega^N$ are the initial and final deployments, respectively. Sensor n's total movement distance is $\sum_{k=1}^{K} \|p_n^k - p_n^{k-1}\|$, and therefore Sensor n's individual energy consumption is formulated as

$$\sum_{k=1}^{K} \mathscr{E}(p_n^{k-1}, p^k) = \xi_n \sum_{k=1}^{K} \|p_n^k - p_n^{k-1}\|. \tag{15}$$

Now, we discuss the distributed realization for the node deployment with (i) network lifetime constraint and (ii) limited communication range. According to the analysis in [10], the movement distance should be constrained at each iteration in order to avoid zigzag movements. Therefore, we limit Sensor n's movement distance in the k-th iteration by an upper bound d_n^k. Note that d_n^k can be a constant or a function of the previous and current deployments. For instance, $d_n^k = \alpha\|p^{k-1} - c^{k-1}\|$ in Lloyd-α [10], where $\alpha \in (0, 1]$. Moreover, to guarantee the required network lifetime, another constraint $\xi_n \sum_{i=1}^{k} \|p_n^i - p_n^{i-1}\| \leq \gamma_n$ should be taken into account. Furthermore, full-connectivity, which is ignored in most distributed sensor relocation algorithms, is definitely required to obtain neighboring information. Then, another constraint $\mathcal{H}\left(\mathbf{P}^k\right) = \mathbf{P}^k$ should also be considered. With the above constraints, each sensor in the distributed scenario optimizes its next stop, $\mathbf{P}^k$, in terms of the previous and current neighboring information, $\mathbf{P}^i, \forall i < k$. In particular, the cell partition in Lloyd-like algorithms is generated by the current deployment [9], [10], [15]–[17], [33], [34], $R_n = V_n(\mathbf{P}^{k-1})$. The corresponding optimization problem, which is

referred to as Problem $\mathcal{B}$, is thus represented as

$$\underset{\mathbf{P}^k}{\text{minimize}} \qquad \sum_{n=1}^{N} \int_{V_n(\mathbf{P}^{k-1})} \eta_n \|p_n^k - \omega\|^2 f(\omega) d\omega \tag{16}$$

$$\text{s.t.} \qquad \mathcal{H}\left(\mathbf{P}^k\right) = \mathbf{P}^k \tag{17}$$

$$\|p_n^k - p_n^{k-1}\| \leq \min\left(\frac{\tilde{e}_n^k}{\xi_n}, d_n^k\right), n \in I_\Omega \tag{18}$$

where $\tilde{e}_n^k = \gamma_n - \xi_n \sum_{i=1}^{k-1} \|p_n^i - p_n^{i-1}\|$ is the residual energy at the k-th iteration. Since full-connectivity is guaranteed by constraint (17), all sensors contribute to the distortion (16).

B. Semi-desired Region and Semi-feasible Region

Before studying the optimal solution for (16), we analyze the constraints (17) and (18). In the distributed scenario, sensors are supposed to relocate simultaneously. However, the full-connectivity strategy used in CCML and BCCML requires a one-by-one relocation scheme, which is not possible in large-scale distributed networks. To follow full-connectivity constraint (17) in a large-scale distributed network, we introduce another important concept, semi-desired region (SDR), which is a shrunk version of the approximated desired region $\tilde{\mathbb{D}}_n\left(\mathbf{P}, \mathcal{I}_\Omega\right)$. Let $\mathcal{G}(\mathbf{P}) = (\mathcal{V}(\mathbf{P}), \mathcal{E}(\mathbf{P}))$ be the undirected connectivity graph comprising a set of vertices $\mathcal{V} = \{p_1, \ldots, p_N\}$ and a set of edges $\mathcal{E} = \{e_{ij}\}$. The edge cost, $w_{ij} = \|p_i - p_j\|$, is defined as the Euclidean distance between the end vertices of the edge, e_{ij}, and there exists an edge between p_i and p_j in $\mathcal{G}(\mathbf{P})$ if and only if $w_{ij} \leq R_c$. For a fully connected graph, $\mathcal{G}(\mathbf{P})$, the corresponding minimum spanning tree (MST) is defined as $\tilde{\mathcal{G}}(\mathbf{P}) = \left(\mathcal{V}(\mathbf{P}), \tilde{\mathcal{E}}(\mathbf{P})\right)$, where $\tilde{\mathcal{E}}(\mathbf{P}) \subset \mathcal{E}(\mathbf{P})$ is a subset of size $|\mathcal{V}(\mathbf{P})| - 1$. Let $\mathcal{N}_n^s(\mathbf{P}) = \{m | \tilde{e}_{nm} \in \tilde{\mathcal{E}}(\mathbf{P}) \text{ or } \tilde{e}_{mn} \in \tilde{\mathcal{E}}(\mathbf{P})\}$ be the set of Sensor n's neighbors in MST. Then, the SDRs are defined as

$$\mathbb{D}_n^s(\mathbf{P}) = \bigcap_{m \in \mathcal{N}_n^s(\mathbf{P})} \mathbb{B}\left(\frac{p_m + p_n}{2}, \frac{R_c}{2}\right), \forall n \in \mathcal{I}_\Omega. \tag{19}$$

An example of SDR is illustrated in Fig. 1c. In this example, 12 sensors with communication range $R_c = 1$ are deployed on the plane, indicating that $\mathcal{I}_\Omega = \{1, \ldots, 12\}$. The edges in MST are denoted by red lines in Fig. 1c. Also, Sensor 1's movement range, $\mathbb{B}\left(p_1^0, d_n\right)$, is demonstrated by a magenta circle. According to the definition of semi-desired region, the green overlap between cyan region $\mathbb{B}\left(\frac{p_1+p_2}{2}, \frac{R_c}{2}\right)$ and yellow region $\mathbb{B}\left(\frac{p_1+p_{12}}{2}, \frac{R_c}{2}\right)$ in Fig. 1c constructs Sensor 1's semi-desired region, $\mathbb{D}_1^s\left(\mathbf{P}\right)$. From Figs. 1a, 1b, and 1c, it is also clear that the semi-desired region is a subset of the approximated desired region and the desired region, i.e.,

$\mathbb{D}_1^s(\mathbf{P}) \subseteq \tilde{\mathbb{D}}_1(\mathbf{P}, \mathcal{I}_\Omega) \subseteq \mathbb{D}_1(\mathbf{P}, \mathcal{I}_\Omega)$.

Theorem 3. *Starting with a fully connected network ($\mathcal{S}(\mathbf{P}^k) = \mathcal{I}_\Omega$), the network is still fully connected ($\mathcal{S}(\mathbf{P}^{k+1}) = \mathcal{I}_\Omega$) if sensors simultaneously move within their respective semi-desired regions i.e., $p_n^{k+1} \in \mathbb{D}_n^s\left(\mathbf{P}^k\right), \forall n \in \mathcal{I}_\Omega$.*

The proof is provided in Appendix C.

Then, we define the semi-feasible regions as

$$\mathbb{F}_n^s(\mathbf{P}, \tilde{e}_n^k, d_n^k) = \mathbb{D}_n^s(\mathbf{P}) \bigcap \mathbb{B}\left(p_n^0, \min\left(\frac{\tilde{e}_n^k}{\xi_n}, d_n^k\right)\right), \forall n \in \mathcal{I}_\Omega, \tag{20}$$

It is trivial to show that both (17) and (18) are satisfied if sensors move within their semi-feasible regions. Let $\mathcal{P}^k = \{\mathbf{P}^k | \mathcal{H}\left(\mathbf{P}^k\right) = \mathbf{P}^k, \|p_n^k - p_n^{k-1}\| \leq \min\left(\frac{\tilde{e}_n^k}{\xi_n}, d_n^k\right), \forall n \in \mathcal{I}_\Omega\}$ be the set of deployments that follow constraints (17) and (18), and $\widehat{\mathcal{P}}^k = \{\mathbf{P}^k | p_n^k \in \mathbb{F}_n^s(\mathbf{P}, \tilde{e}_n^k, d_n^k), \forall n \in \mathcal{I}_\Omega\}$ be the set of deployments that are placed within semi-feasible regions. Then, we have $\widehat{\mathcal{P}}^k \subseteq \mathcal{P}^k$. To simplify the problem, we replace $\mathcal{P}^k$ by $\widehat{\mathcal{P}}^k$, and the optimization problem is represented as N independent problems:

$$\underset{\mathbf{P}^k}{\text{minimize}} \qquad \int_{V_n(\mathbf{P}^{k-1})} \eta_n \|p_n^k - \omega\|^2 f(\omega) d\omega \tag{21}$$

$$\text{s.t.} \qquad p_n^k \in \mathbb{F}_n^s(\mathbf{P}, \tilde{e}_n^k, d_n^k) \tag{22}$$

where $n \in \mathcal{I}_\Omega$. By parallel axis theorem [41], (21) can be rewritten as

$$\int_{V_n(\mathbf{P}^{k-1})} \eta_n \|c_n^{k-1} - \omega\|^2 f(\omega) d\omega + \eta_n \|p_n^k - c_n^{k-1}\|^2 v_n^{k-1}, \tag{23}$$

where $v_n^{k-1} = \int_{V_n(\mathbf{P}^{k-1})} f(\omega) d\omega$ and $c_n^{k-1} = \frac{\int_{V_n(\mathbf{P}^{k-1})} \omega f(\omega) d\omega}{\int_{V_n(\mathbf{P}^{k-1})} f(\omega) d\omega}$. Since the first term in (23) is a constant, Sensor n's distortion is an increasing function of the distance from p_n^k to c_n^{k-1}. Accordingly, the optimal solution for (21) with constraint (22) is the point closet to c_n^{k-1} within $\mathbb{F}_n^s(\mathbf{P}, \tilde{e}_n^k, d_n^k)$, i.e., $p_n^k = \arg\min_{q \in \mathbb{F}_n^s(\mathbf{P}, \tilde{e}_n^k, d_n^k)} \|q - c_n^{k-1}\|$. By moving sensors to $\arg\min_{q \in \mathbb{F}_n^s(\mathbf{P}, \tilde{e}_n^k, d_n^k)} \|q - c_n^{k-1}\|$ at each iteration, we will have a distributed realization, Distributed Constrained Movement Lloyd (DCML) Algorithm. According to the above analysis, DCML Algorithm will result in a deployment that guarantees both connectivity and the required network lifetime. Like CCML Algorithm, DCML Algorithm is an iterative improvement algorithm in which the distortion is non-increasing and converges. The proof is similar to that of Theorem 2 and therefore omitted here. More details about DCML Algorithm are shown in Algorithm 3.

Algorithm 3 Distributed Constrained-Movement Lloyd Algorithm in heterogeneous MWSNs

Input: Target area Ω; probability density function $f(\cdot)$; the initial sensor deployment $\mathbf{P}^0$; the required network lifetime T; The number of stops K; the communication range R_c.

Output: Sensors deployments at K stops $\{\mathbf{P}^k\}_{k\in\{1,\ldots,K\}}$; Distortion at the final deployment $D(\mathbf{P}^K)$.

1: Calculate energy constraints γ_n and initialize the residual energy $\tilde{e}_n^1 = \gamma_n, \forall n \in \mathcal{I}_\Omega$
2: **for** k = 1 to K **do**
3: Determine the maximum movement distances $d_n^k = \min\left(\frac{\tilde{e}_n^k}{\xi_n}, \frac{2\tilde{e}_n^k}{K-k+1}\right), n \in \mathcal{I}_\Omega$
4: Sensors obtain their MST neighbors $\{\mathcal{N}_n^s\}_{n\in\mathcal{I}_\Omega}$ by GHS Algorithm [42]
5: Do multiplicatively weighted Voronoi partition, and then update centroid $\{c_n(\mathbf{P}^{k-1})\}_{n\in\mathcal{I}_\Omega}$
6: Calculate semi-desired regions $\{\mathbb{D}_n^s(\mathbf{P}^{k-1}, d_n^k)\}_{n\in\mathcal{I}_\Omega}$
7: Calculate the critical point q_n^k, closest point to $c_n(\mathbf{P}^{k-1})$ within $\mathbb{D}_n^s(\mathbf{P}^{k-1}, d_n^k)$, $\forall n \in \mathcal{I}_\Omega$
8: Each sensor moves to its next stop $p_n^k = p_n^{k-1} + \min\left(\frac{\gamma_n}{2K\xi_n}, \|q_n^k - p_n^{k-1}\|\right)\frac{q_n^k - p_n^{k-1}}{\|q_n^k - p_n^{k-1}\|}, \forall n \in \mathcal{I}_\Omega$
9: Update residual energy $\tilde{e}_n^{k+1} = \tilde{e}_n^k - \xi_n\|p_n^k - p_n^{k-1}\|, \forall n \in \mathcal{I}_\Omega$
10: **end for**

V. Algorithm complexity and communication overhead

A. Algorithm Complexity

Before we calculate the complexity of different algorithms, we need to study the computational complexity of FR, AFR, SFR, $c_n(\mathbf{P})$ and $v_n(\mathbf{P})$. Finding Sensor n's FR, $\mathbb{F}_n(\mathbf{P}, \mathcal{I})$, includes the following four stages: (i) Determine $K_n(\mathbf{P}, \mathcal{I})$ disjoint components $\{U_{nk}(\mathbf{P}, \mathcal{I})\}_{k=1,\ldots,K_n(\mathbf{P},\mathcal{I})}$ by Breadth First Search (BFS) or Depth First Search (DFS) with time complexity $O(card(\mathcal{I}))$[3]. (ii) Calculate the union of communication balls in each component, i.e., $\bigcup_{j\in U_{nk}(\mathbf{P},\mathcal{I})} \mathbb{B}(p_j, R_c)$, $\forall k \in \{1, \ldots, K_n(\mathbf{P}, \mathcal{I})\}$. Note that the complexity of calculating the union of two regions is a constant, $O(1)$. Thus, the complexity of Stage (ii) is $O(card(\mathcal{I}))$. (iii) Obtain $\mathbb{D}_n(\mathbf{P}, \mathcal{I})$ by calculating the intersection of $K_n(\mathbf{P}, \mathcal{I})$ regions obtained from the previous stage. Since the computational complexity of intersection is $O(1)$, the complexity in Stage (iii) is $O(K_n(\mathbf{P}, \mathcal{I}))$. (iv) Compute the intersection between $\mathbb{D}_n(\mathbf{P}, \mathcal{I})$ and $\mathbb{B}_n(p_n^0, \frac{\gamma_n}{\xi_n})$ with complexity $O(1)$.

Therefore, the computational complexity of $\mathbb{F}_n(\mathbf{P}, \mathcal{I})$ is $O(card(\mathcal{I}) + K_n(\mathbf{P}, \mathcal{I}))$. In total, the complexity of finding $card(\mathcal{I})$ sensors' FRs is $O(card(\mathcal{I})^2 + card(\mathcal{I}) \cdot K_n(\mathbf{P}, \mathcal{I}))$. In general, $K_n(\mathbf{P}, \mathcal{I}) \ll card(\mathcal{I})$, and then the corresponding complexity becomes $O(card(\mathcal{I})^2)$. The computations of AFR and SFR are similar to that of FR, and have the same complexity $O(card(\mathcal{I})^2)$.

Next, we study the computational complexity of $c_n(\mathbf{P})$ and $v_n(\mathbf{P})$. Let $\mu(R) = \int_R \omega d\omega$ be the volume of region R with the uniform distribution. Many integral algorithms, such as uniform sampling, stratified sampling, importance sampling, sequential Monte Carlo, and Risch algorithm

[3]The exact complexity of BFS and DFS is $O(N + E)$, where E is the number of connections in the communication graph. Since each sensor has very limited neighbours in the range of R_c, we have $O(N) = O(E)$ and thus $O(N + E) = O(N)$.

are available in the literature [43], [44]. For simplicity, we assume the integrals in $c_n(\mathbf{P})$ and $v_n(\mathbf{P})$ are calculated by uniform sampling[4]. In this case, the computational complexity of $c_n(\mathbf{P})$ and $v_n(\mathbf{P})$ is proportional to the number of samples, $O(\frac{\mu(V_n(\mathbf{P}))}{\epsilon})$, where ϵ is the sample size. Thus, the total complexity of computing all $c_n(\mathbf{P})$s and $v_n(\mathbf{P})$s is $O(\frac{\sum_{n=1}^{N}\mu(v_n(\mathbf{P}))}{\epsilon}) = O(\frac{\mu(\Omega)}{\epsilon})$.

Now, we have enough materials to derive the complexity of different algorithms. Since CCML is deigned for homogeneous MWSN, the backbone network includes all sensors, i.e., $card(\mathcal{I}) = N$. Therefore, CCML's algorithm complexity is $O((N^2 + \frac{\mu(\Omega)}{\epsilon})K)$, where K is the number of iterations. Let Z be the number of sensors out of the final backbone network. BCCML's algorithms complexity is calculated as $O((\sum_{z=0}^{Z}(N-z)^2 + O(\frac{\mu(\Omega)}{\epsilon}))K)$. The complexity of the worst case, $Z = 1$, is then $O((N^2 + \frac{\mu(\Omega)}{\epsilon})NK)$. Different from CCML and BCCML, DCML is a distributed algorithm. Thus, we focus on Sensor n's complexity in DCML. Note that MST can be obtained by a distributed minimum spinning tree algorithm, GHS Algorithm [42], with complexity $O(N\log(N))$. At the k-th stop, DCML calculates (a) $c_n(\mathbf{P})$s and $v_n(\mathbf{P})$, (b) $\mathbb{F}_n(\mathbf{P}, \mathcal{I})$ and (c) MST with complexities $O(\frac{\mu(V_n(\mathbf{P}^k))}{\epsilon})$, $O(N)$, and $O(N\log(N))$, respectively. In total, DCML's complexity during K stops is then $O(KN\log(N) + \sum_{k=1}^{K}\frac{\mu(V_n(\mathbf{P}^k))}{\epsilon})$. Obviously, as a distributed algorithm, DCML's complexity is smaller than that of CCML and BCCML. i.e., $O(KN\log(N) + \sum_{k=1}^{K}\frac{\mu(V_n(\mathbf{P}^k))}{\epsilon}) < O((N^2 + \frac{\mu(\Omega)}{\epsilon})K) < O((N^2 + \frac{\mu(\Omega)}{\epsilon})NK)$.

B. Communication Overhead

In centralized implementations, the relocation includes the following stages: (i) Sensors send their spatial information to AP via multi-hop communications; (ii) AP calculates the final deployment by running CCML/BCCML in terms of the collected information; (iii) AP sends final deployment to each sensor by multi-hop communications; (iv) Sensors are relocated to their destinations based on the received deployment. In the worst case, where sensors are connected with a line topology, the sensor with a hops to AP will transfer $N - a$ messages from itself and $N - 1 - a$ farther sensors in Stage (2). Thus, the communication overhead in Stage (ii) is $O(\sum_{a=1}^{N-1} a) = O(N^2)$. Similarly, the communication overhead in Stage (iv) is also $O(N^2)$. Therefore, the total communication overhead in centralized implementations (CCML and BCCML) is $O(N^2)$.

In what follows, we analyze the communication in the distributed implementation. At each stop, sensors need to compute (a) their own Voronoi Diagrams and (b) MST in terms of

[4]The integral is approximated by the summation over uniform samples.

their neighbor information. Sensor n can compute its Voronoi Diagram by exchanging message with its one-hop neighbors [6], [10]. Since each sensor has very limited number of one-hop neighbours, the communication overhead of N sensors for Voronoi Diagram is $O(N)$. In addition, communication overhead for distributed MST for each stop is $O(N\log(N))$ [42]. Therefore, the total communication overhead in a K-stop distributed implementation (DCML) is $O(K(N\log(N)+N)) = O(KN\log(N))$.

VI. Extension

In this section, we extend the proposed algorithms, CCML and DCML, to other sensing tasks: area coverage and target coverage. We employ the binary coverage model [2]–[13], [32], [33] in which Sensor n can only detect the points within its sensing radius $R_{s,n}$. Intuitively, in order to decrease the sensing uncertainty, CCML and DCML deploy sensors into high-density regions, and thus the points with high density are more likely to be covered. To cover the objects in different tasks, the density function $f(\omega)$ in (2) should be predetermined to highlight the points around the objects of interest. In the following three subsections, we introduce three kinds of coverage and propose the corresponding density functions.

A. Area Coverage

Without any prior information about the target region, the density function is chosen to be uniform, i.e., $f(\omega) = \frac{1}{\int_\Omega d\omega}, \forall \omega \in \Omega$. Under such circumstances, maximizing the area covered by sensors is a primary task. To evaluate the corresponding sensing performance, we employ area coverage [6], [23], [33] (the proportion of covered area) defined by

$$C^{\mathcal{A}}(\mathbf{P}) = \frac{\int_{\bigcup_{n=1}^{N}\mathbb{B}(p_n,R_{s,n})} d\omega}{\int_\Omega d\omega} = \frac{\sum_{n=1}^{N}\int_{V_n(\mathbf{P})\bigcap\mathbb{B}(p_n,R_{s,n})} d\omega}{\sum_{n=1}^{N}\int_{V_n(\mathbf{P})} d\omega}. \tag{24}$$

The experimental results in Section VII show that CCML and DCML algorithms with uniform density function provide a large area coverage in addition to a small distortion.

B. Target Coverage

In another popular scenario, sensors are deployed to collect detailed information from the targets with known locations [24]–[27]. Let $\mathcal{T} = \{t_1, t_2, \ldots, t_M\}$ be the set of known targets, $\widehat{\mathcal{T}} = \{t | \min_n \left(\frac{\|t-p_n\|}{R_{s,n}}\right) \leq 1, t \in \mathcal{T}\}$ be the set of targets that covered by at least one sensor. Then, area coverage $C^{\mathcal{T}}(\mathbf{P})$ - the proportion of covered target points - can be written as

$$C^{\mathcal{T}}(\mathbf{P}) = \frac{\mathbf{card}(\widehat{\mathcal{T}})}{\mathbf{card}(\mathcal{T})} = \frac{\sum_{n\in\mathcal{S}(\mathbf{P})}\int_{V_n(\mathbf{P})\bigcap\mathbb{B}(p_n,R_{s,n})}\left[\sum_{m=1}^{M}\phi\left(\|\omega-t_m\|\right)\right]d\omega}{\sum_{n\in\mathcal{S}(\mathbf{P})}\int_{V_n(\mathbf{P})}\left[\sum_{m=1}^{M}\phi\left(\|\omega-t_m\|\right)\right]d\omega}, \tag{25}$$

where $\phi(\cdot)$ is the unit impulse response, $\mathbf{card}(S)$ is the cardinality of the set S.

To emphasize the importance of discrete targets, we model the density function as a Gaussian mixture centered at discrete targets. The corresponding density function can be written as

$$f(w) = \sum_{m=1}^{M} A_m e^{-\frac{\|w-t_m\|^2}{R_{s,n}^2}} \tag{26}$$

where A_m reflects the comparative importance of the target q_m. Similarly, CCML and DCML can also be extended to maximize barrier coverage [28]–[30].

VII. Performance Evaluation

We provide the simulation results for two different MWSNs: (1) MWSN1: A homogeneous MWSN in which all sensors have the same characteristics. (2) MWSN2: A heterogeneous MWSN including sensors with different sensing, moving cost parameters, and battery energies. In addition, we employ uniform density function, $f(\omega) = 1$, for MWSN1 while the non-uniform density function in [6], [33] is employed for MWSN2. The non-uniform density function is the sum of five Gaussian functions of the form $5exp(6(-(x-x_{center})^2-(y-y_{center})^2))$. The centers (x_{center}, y_{center}) are (2,0.25), (1,2.25), (1.9,1.9), (2.35,1.25) and (0.1,0.1). Moreover, the target region, Ω, which is also the same as in [6], [33], is determined by the polygon vertices (0,0), (2.125,0), (2.9325,1.5), (2.975,1.6), (2.9325,1.7), (2.295,2.1), (0.85,2.3), (0.17,1.2). Also, we set the power consumption after sensor relocation as $\beta = 1$. As a result, the energy constraints can be calculated by $\gamma_n = e_n - T$, where e_n is Sensor n's battery energy. Other parameters are provided in Table I. Moreover, we generate initial sensor deployments randomly, i.e., every node location is generated with uniform distribution on Ω. To guarantee the initial full-connectivity, we sequentially generate random node locations, and only keep a node if it connects with at least one previous node. The maximum number of iterations is set to 100[5].

TABLE I: Simulation Parameters

Parameters	N	$\eta_1-\eta_8$	$\eta_9-\eta_{32}$	$\xi_1-\xi_8$	$\xi_9-\xi_{32}$	$R_{s,1}-R_{s,8}$	$R_{s,9}-R_{s,32}$	e_1-e_{28}	$e_{29}-e_{32}$
MWSN1	32	1	1	1	1	0.2	0.2	2	2
MSWN2	32	1	4	2	1	0.3	0.15	2	0.8

To evaluate the performance, we compare the distortion (4) and area coverage (24) of CCML, BCCML, and DCML with those of VFA [35], Lloyd-α [10], and DEED [10]. We run the algorithms for: (i) the centralized scheme where each sensor's energy consumption for relocation is determined by the distance from the initial location and the final location; and (ii) the distributed scheme where each sensor's energy consumption for relocation is determined by the total distance

[5]100 is large enough for the proposed algorithms to converge. Therefore, instead of using the stop threshold ϵ in CCML and BCCML, we run all algorithms 100 iterations in the experiments.

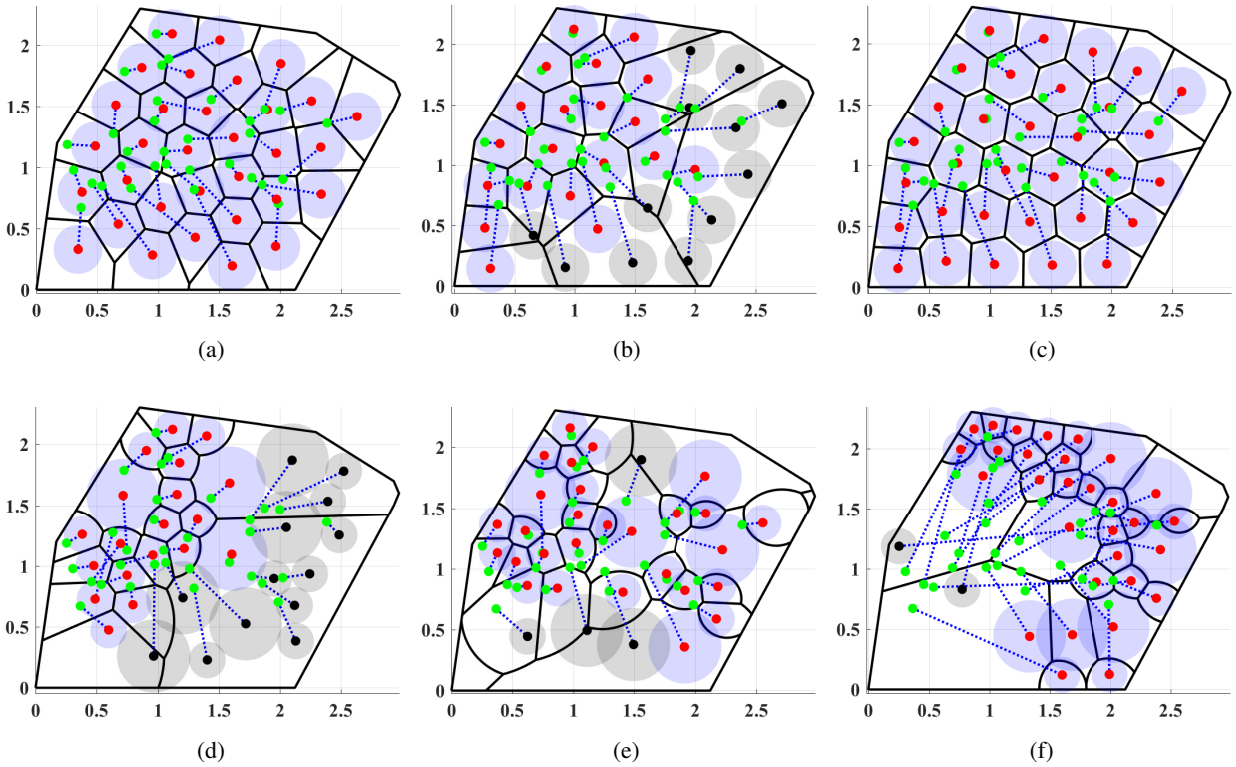


Fig. 2: Centralized sensor deployments: (a) VFA in MWSN1; (b) Lloyd-α in MWSN1; (c) CCML in MWSN1; (d) VFA in MWSN2; (e) Lloyd-α in MWSN2; (f) BCCML in MWSN2. The initial sensor locations are denoted by green dots. The final locations of active and inactive sensors are denoted by red and black dots. The sensing regions of active and inactive sensors are denoted by blue and black. The movement paths are denoted by blue lines.

of its specific (100-stop) movement path. Several important simulation details are provided as follows. Since network lifetime is not considered in VFA [35], it is impossible to apply the original VFA to satisfy the required network lifetime. Thus, we propose a variant of VFA in which each sensor stops moving after the predetermined energy, γ_n, is consumed. Furthermore, when the communication range R_c is limited, the lack of full-connectivity prevents VFA, Lloyd-α, and DEED from operating in a distributed scheme. To compare them with our DCML Algorithm, sensors need to have global information in VFA, Lloyd-α, and DEED. Another issue is that sensors may be divided into multiple disconnected sub-graphs after running VFA, Lloyd-α, and DEED because of the limited communication range. Under such circumstances, we compute the distortions associated with different sub-graphs and report the minimum one. In other words, we focus on the best performances that VFA, Lloyd-α, and DEED can reach in MWSNs when communication range is limited. Nonetheless, when we compute the distortion for our proposed algorithms, only sensors in the actual backbone network (the sub-graph including AP, i.e., Sensor 1), are taken into account, which gives our algorithm more advantage over the existing algorithms.

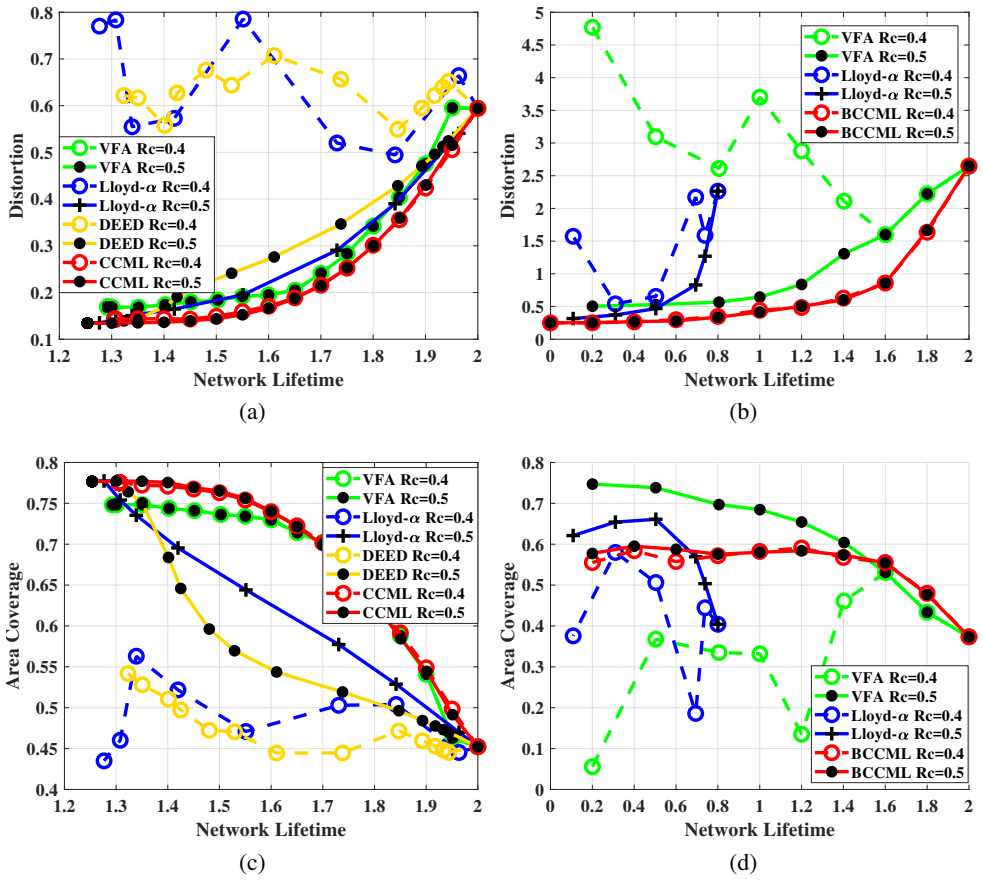


Fig. 3: Performance comparison for centralized sensor deployment. (a) Distortion in MWSN1; (b) Distortion in MWSN2; (c) Area coverage in MWSN1; (d) Area coverage in MWSN2.

Simulation results for the centralized scheme are provided in Figs. 2 and 3. From Figs. 2a and 2c, we observe that both VFA and CCML algorithms generate fully connected final deployments for the required network lifetime, $T = 1.3$. By setting α to 0.2, Lloyd-α achieves a similar network lifetime $T = 1.31$. However, Lloyd-0.2 generates a disconnected network where 12 sensors are placed out of the backbone network. The corresponding distortions for VFA, Lloyd-0.2, and CCML are, respectively, 0.17, 0.78, and 0.14.

The centralized sensor relocations in heterogeneous MWSN (MWSN2) are illustrated in Figs. 2d, 2e, and 2f. In MWSN2, BCCML activates 30 sensors to sense the target region while the other two sensors are deactivated because of their low battery energy. VFA and Lloyd-α attempt to use all sensors to finish the sensing task, but AP can only collect information from the active sensors shown by red dots. Like the comparisons in MWSN1, BCCML's distortion, 0.29, is much smaller than that of VFA and Lloyd-α, which are 4.16 and 0.66, respectively.

More detailed performance comparisons are provided in Figs. 3a, 3b, 3c, and 3d. In Fig. 3a, for any given network lifetime, CCML Algorithm provides a lower distortion compare to

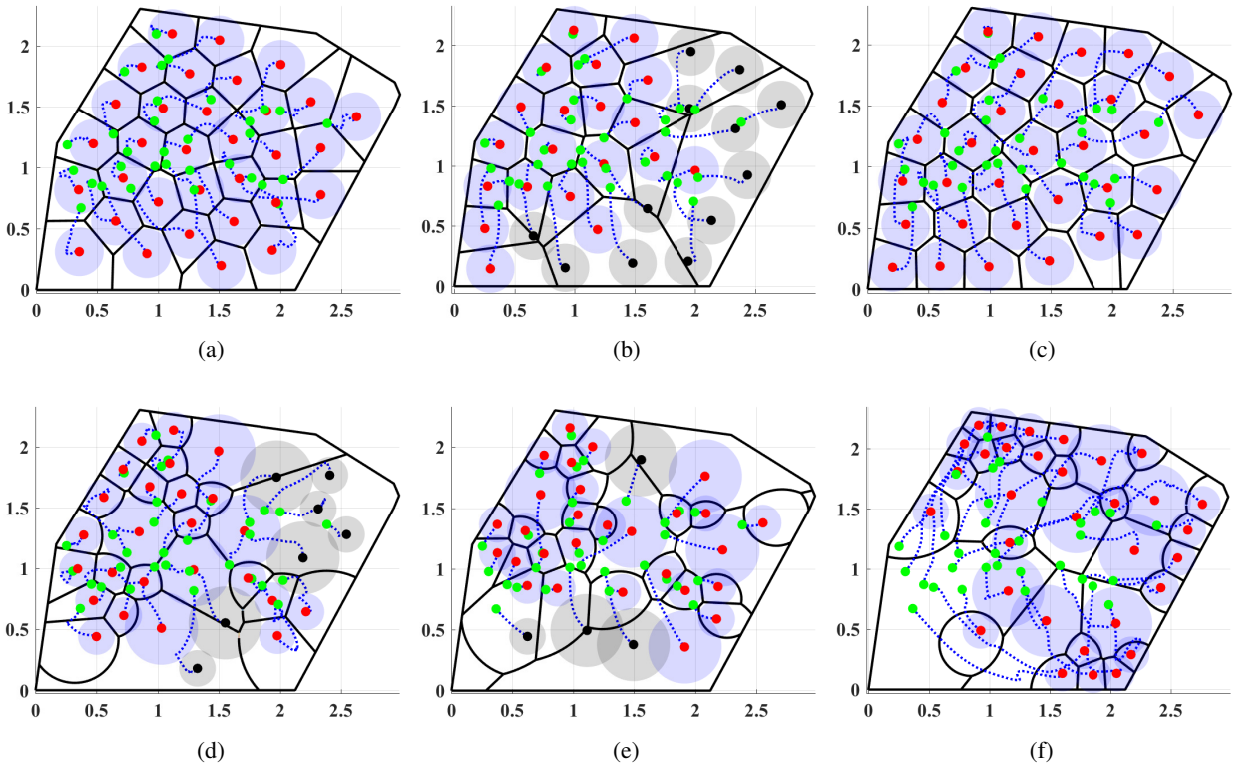


Fig. 4: Distributed sensor deployments: (a) VFA in MWSN1; (b) Lloyd-α in MWSN1; (c) DCML in MWSN1; (d) VFA in MWSN2; (e) Lloyd-α in MWSN2; (f) DCML in MWSN2. The initial sensor locations are denoted by green dots. The final locations of active and inactive sensors are denoted by red and black dots. The sensing regions of active and inactive sensors are denoted by blue and black. The movement paths are denoted by blue lines.

other algorithms in the homogeneous MWSN1. Similarly, BCCML Algorithm outperforms other algorithms in heterogeneous MWSNs shown in Fig. 3b. It is also noteworthy that low distortion is accompanied with high area coverage in MWSN1 where the density function is uniform. Unfortunately, such a relationship does not hold for MWSN2 where the density function is non-uniform. Consequently, one can approximately optimize the area coverage by CCML and BCCML with a uniform density function.

There are two primary reasons why the proposed CCML and BCCML perform better than the existing algorithms, VFA, Lloyd-α, and DEED. (i) The existing algorithms do not take connectivity into consideration. As a result, when the communication range is small, VFA, Lloyd-α, and DEED may generate disconnected networks and then large distortions. (ii) In the existing algorithms, each sensor attempts to save its own energy consumption which results in unbalanced energy consumption among sensors and then short network lifetime. However, CCML and BCCML determine the relocation considering all sensors' residual energy.

Besides, our proposed algorithms perform well in both homogeneous and heterogeneous

MWSNs, but the existing algorithms, which are designed for homogeneous MWSNs, have very restricted performance in heterogeneous MWSNs or even cannot be applied to heterogeneous MWSNs. Note that the implementation of DEED needs both gradient and Hessian matrix of the objective function (4). To our best knowledge, the theoretical computation of Hessian matrix in heterogeneous MWSNs is still an open problem. Although one can approximate the second-order derivatives by numerical methods, the corresponding extreme time complexity prevents DEED from being a feasible solution. Therefore, DEED cannot be extended to heterogeneous MWSNs. Different from DEED, Lloyd-α, which only needs gradient, can be extended to heterogeneous MWSNs as the calculation of the gradient in heterogeneous MWSNs already proposed in our previous work [33]. Unfortunately, when sensors are equipped with variant battery energies, Lloyd-α can only achieve a short network lifetime. In MWSN2, where 4 sensors are equipped with a low battery energy, 0.8, Lloyd-α still uses all sensors to sense the target region. Note that the network will die after the first node runs out of its battery energy. As a result, Lloyd-α cannot achieve a network lifetime larger than 0.8 in MWSN2 (see Fig. 3b). However, the proposed BCCML Algorithm appropriately selects a subset of sensors to finish the sensing task.

Another advantage of BCCML Algorithm over Lloyd-α and DEED is its tractability. BCCML Algorithm directly controls the network lifetime while Lloyd-α and DEED indirectly influence network lifetime by tuning the hyperparameters α and δ, respectively. There is no explicit relationship between α (or δ) and network lifetime. Thus, one has to attempt different values of α (or δ) in Lloyd-α (or DEED) to reach the required network lifetime.

In what follows, we analyze the impact of communication range, R_c, on the performance. The sensors with larger communication range, R_c, are more likely to construct or maintain a connected network. According to our simulation results, VFA, Lloyd-α, DEED, CCML and BCCML keep full-connectivity for the two considered MWSNs when the communication range is large, e.g., $R_c = 0.5$. However, when $R_c = 0.4$, VFA, Lloyd-α, DEED lose connectivity in some cases. In general, a shorter network lifetime implies that more energy can be used for relocation, and then smaller distortions can be achieved. We observe from Figs 3a and 3b that all algorithms provide non-decreasing Distortion-Lifetime functions for the cases of $R_c = 0.5$. However, when $R_c = 0.4$, the distortions of VFA, Lloyd-α, and DEED fluctuate because of the unpremeditated loss of connectivity in some cases. On the contrary, the distortions of CCML and BCCML are still non-decreasing functions of network lifetime because the backbone network is appropriately selected and maintained by CCML and BCCML. In sum, for VFA, Lloyd-α, and

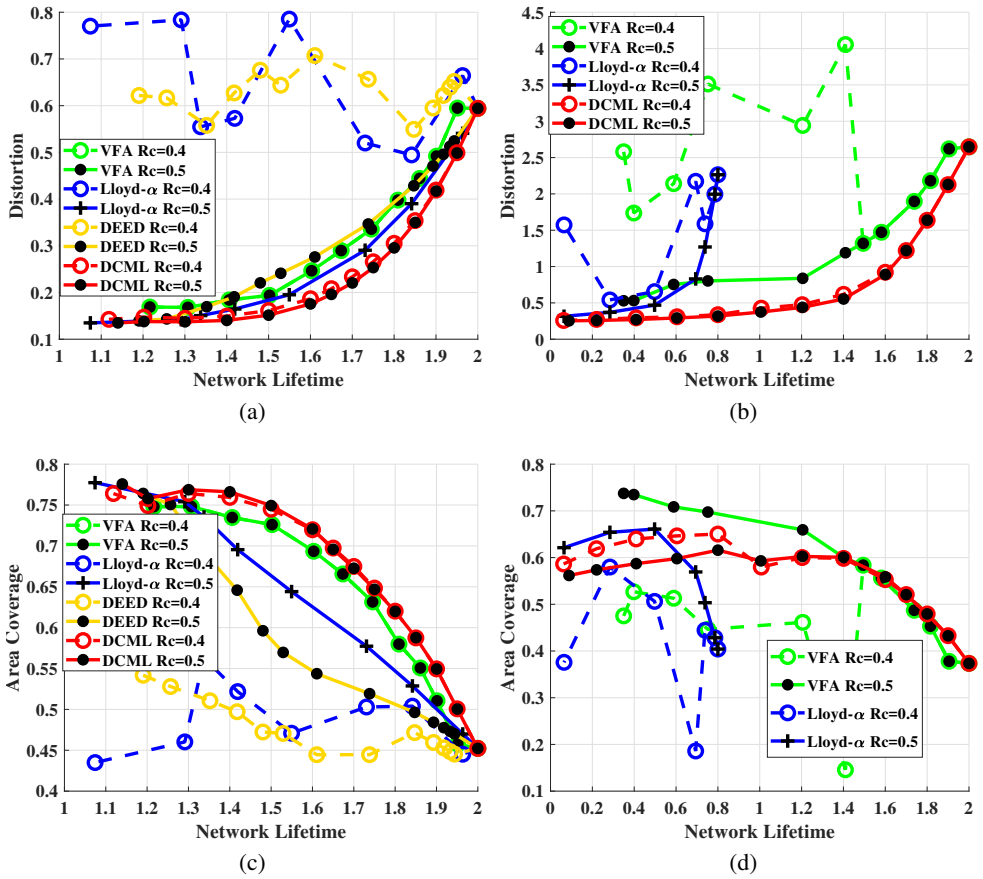
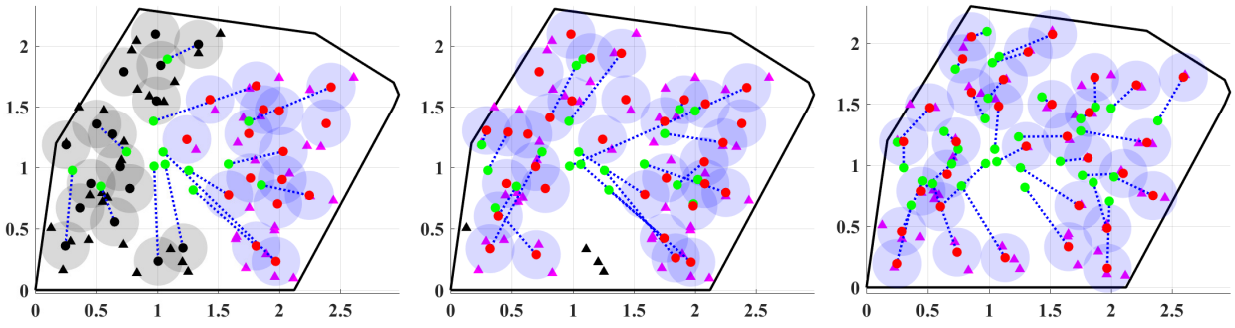


Fig. 5: Performance comparison for distributed sensor deployment. (a) Distortion in MWSN1; (b) Distortion in MWSN2; (c) Area coverage in MWSN1; (d) Area coverage in MWSN2.

DEED algorithms, the distortions are significantly increased when R_c is decreased from 0.5 to 0.4. However, CCML/BCCML provides similar performance for both R_cs.

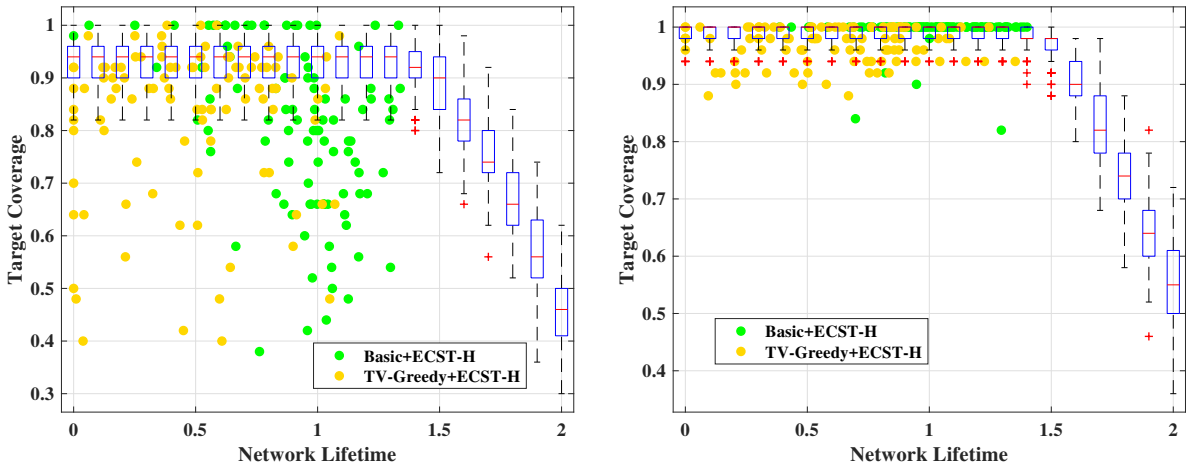
The above analysis also works for the distributed sensor relocation scheme. The distributed relocations in both MWSNs are illustrated in Fig. 4, and the corresponding performances are compared in Figs. 5a, 5b, 5c, and 5d. Different from the sensor relocation in the centralized implementation (Fig. 2) where movement paths are straight lines from initial locations to final locations, the sensor relocation in the distributed implementation (Fig. 4) are represented by broken lines with multiple stops.

Last, we provide sensor relocations of three different algorithms (Basic+ECST-H [27], TV-Greedy+ECST-H [27], and CCML) in Figs. 6 and 7 to confirm that CCML Algorithm can be successfully extended to target coverage problems. Both Basic+ECST-H and TV-Greedy+ECST-H consist of three stages: (i) A subset of sensors are placed to cover all targets; and (ii) other sensors are placed to guarantee connectivity; (iii) Hungarian Algorithm is employed to reduce the total energy consumption. According to our experiments, Basic+ECST-H and TV-Greedy+ECST-



(a) (b) (c)

Fig. 6: The target coverage in MWSN1. (a) Basic+ECST-H; (b) TV-Greedy+ECST-H; (c) CCML. The covered targets and uncovered targets are denoted by magenta triangles and black triangles, respectively.



(a) (b)

Fig. 7: The target coverage in MWSN1. (a) $R_c = 0.4$, $R_s = 0.2$; (b) $R_c = 0.5$, $R_s = 0.25$. The target coverage of Basic+ECST-H, TV-Greedy+ECST-H, and CCML are, respectively, denoted by green dots, yellow dots and blue boxes.

H require more sensors than CCML to achieve both full-coverage and full-connectivity. In Fig. 6a, all sensors are scheduled to cover targets in Stage (i) of Basic+ECST-H, and then no sensor is available in Stage (ii). As a result, Basic+ECST-H with $R_c = 0.4$ is terminated with four disconnected subgraphs in which the largest sub-graph merely covers 48% of targets. In Fig. 6b, 32 sensors are not enough to achieve full-coverage in Stage (i) of TV-Greedy+ECST-H, and only 92% of targets are covered. However, with the same number of sensors, our proposed CCML Algorithm covers all targets and ensures full-connectivity. To perform a statistical performance analysis, we run the relocation algorithms with 100 random target deployments[6]. The detailed comparisons[7] of Basic+ECST-H, TV-Greedy+ECST-H, and CCML are provided in Figs. 7a and

[6]Each target deployment consists of 50 target locations.

[7]For each required network lifetime, a boxplot is employed to display minimum, first quartile, median, third quartile, and maximum distortions of CCML with 100 target deployments. However, the target coverages of Basic+ECST-H and TV-Greedy+ECST-H are represented by, in total, 200 dots since network lifetime is not explicitly controlled by Basic+ECST-H and TV-Greedy+ECST-H.

7b. Given a specific target coverage, CCML, on average, achieves a longer network lifetime.

VIII. Conclusions and Discussion

The trade-off between sensing quality and energy consumption, which is dominated by movement, is discussed in this paper. We studied the optimal sensor deployment to minimize sensing uncertainty with a network lifetime constraint in both homogeneous and heterogeneous mobile wireless sensor networks. To make the model more practical, we take connectivity, which has a crucial influence on sensing performance, into consideration. According to our analysis, full-connectivity is necessary to minimize the sensing uncertainty in homogeneous MWSNs. The necessary condition for an optimal deployment implies that sensors should move towards the centroid within their own feasible regions, determined by both the battery energies and the communication range. With the help of these necessary conditions, two centralized sensor relocation algorithms, Centralized Constrained Movement Lloyd Algorithm and Backwards-stepwise Centralized Constrained Movement Lloyd Algorithm, are designed for homogeneous and heterogeneous MWSNs, respectively. Moreover, a distributed realization, Distributed Constrained Movement Lloyd Algorithm, whose performance is similar to the centralized scheme, is also provided in this paper. Furthermore, by manually changing the density function, we extend the proposed sensor relocation algorithms to target coverage. Our simulation results show that the proposed algorithms outperform the existing algorithms in the literature (VFA, Lloyd-α, DEED) when a minimum network lifetime is given in homogeneous and heterogeneous MWSNs. Compared with the existing target coverage algorithms, such as Basic+ECST-H and TV-Greedy+ECST-H, CCML Algorithm provides a more flexible trade-off between target coverage and network lifetime.

Appendix A
Proof of Lemma 1

Lemma 1 focuses on homogeneous MWSNs where $\eta_n = \eta$, $\xi_n = \xi$, and $\gamma_n = \gamma, \forall n \in \mathcal{I}_\Omega$. Let $\mathbf{P}^0 = (p_1^0, \ldots, p_N^0)$ and $\mathbf{P}^* = (p_1^*, \ldots, p_N^*)$ be the initial and the optimal sensor deployments in an MWSN with performance function (4) and constraints (8), respectively. For convenience, let $\mathcal{I}_\Omega$ be the set of all sensors, $\mathcal{S}(\mathbf{P})$ be the set of sensors that can communicate with the AP when the sensor deployment is $\mathbf{P}$, and $\mathcal{N}_n(\mathbf{P})$ be Sensor n's neighbors given deployment $\mathbf{P}$. $\mathcal{S}(\mathbf{P})$ is also referred to as the backbone network in Section II. In Lemma 1, we assumed that $\mathbf{P}^0$ provides a fully connected network, i.e., $S(\mathbf{P}^0) = \mathcal{I}_\Omega$. Now, we assume that the optimal deployment is associated with a disconnected network, i.e., $\mathcal{S}(\mathbf{P}^*) \neq \mathcal{I}_\Omega$. In this case, we can

find a sensor, $n \in \mathcal{S}(\mathbf{P}^*)$, such that one of its neighbors in the initial deployment, $m \in \mathcal{N}_n(\mathbf{P}^0)$, is not in the final backbone network, $m \notin \mathcal{S}(\mathbf{P}^*)$. An alternative point is defined as

$$p'_m = p^0_m + \min\left(0, \|p^*_n - p^0_m\| - R_c\right) \frac{p^*_n - p^0_m}{\|p^*_n - p^0_m\|}. \tag{27}$$

Replacing p^*_m by p'_m, we get an alternative deployment $\mathbf{P'} = (p^*_1, \ldots, p'_m, \ldots, p^*_N)$. Next, we check if $\mathbf{P'}$ satisfies the network lifetime constraints (8). First, since sensors m and n are neighbors, we have $\|p^0_m - p^0_n\| \leq R_c$. Second, following the constraints (8), we have $\|p^*_n - p^0_n\| \leq \frac{\gamma}{\xi}$. Third, using the triangular inequality, we have $\|p^0_n - p^0_m\| + \|p^*_n - p^0_n\| > \|p^0_m - p^*_n\|$. Combining the above three inequalities, we obtain $\|p^*_n - p^0_m\| < R_c + \frac{\gamma}{\xi}$. According to (27), p'_m is placed between p^*_n and p^0_m, and the moving distance is

$$\|p'_m - p^0_m\| = \min\left(0, \|p^*_n - p^0_m\| - R_c\right) < \frac{\gamma}{\xi}. \tag{28}$$

Thus, the deployment $\mathbf{P'}$ satisfies the network lifetime constraints.

In what follows, we verify that $\mathbf{P'}$ provides a smaller distortion compared to $\mathbf{P}^*$. The distance between p'_m and p^*_n can be calculated as

$$\|p^*_n - p'_m\| = \|p^*_n - p^0_m\| - \|p'_m - p^0_m\| = \begin{cases} \|p^*_n - p^0_m\|, & \text{if } \|p^*_n - p^0_m\| \leq R_c \\ R_c, & \text{otherwise} \end{cases}, \tag{29}$$

which means $\|p^*_n - p'_m\| \leq R_c$. In other words, when the deployment is $\mathbf{P'}$, Sensor m connects with n and then should be taken into the calculation of distortion (4). In our model, sensors are initially deployed in the target region Ω, i.e., $p^0_n \in \Omega$, where Ω is a convex region [33]. Moveover, it is self-evident that the optimal sensor locations are also in the target region, i.e., $p^*_n \in \Omega, \forall n \in \mathcal{I}_\Omega$. By properties of a convex region, any point between p^*_n and p^0_m should be in the target region, e.g., $p'_m \in \Omega$. Therefore, Sensor m is associated with a non-empty MWVD $V_m(\mathcal{H}(\mathbf{P'})) = \{\omega | \omega \in \Omega, \|\omega - p'_m\| < \|\omega - p^*_i\|, \forall i \in \mathcal{S}(\mathbf{P'}) i \neq m\}$. The difference between the distortions at $\mathbf{P'}$ and $\mathbf{P}^*$ lays on $V_m(\mathcal{H}(\mathbf{P'}))$ and can be calculated as

$$\begin{aligned} D(\mathbf{P'}) - D(\mathbf{P}^*) &= \int_{V_m(\mathcal{H}(\mathbf{P'}))} \eta \|\omega - p'_m\|^2 f(\omega) d\omega - \sum_{n \in \mathcal{S}(\mathcal{H}(\mathbf{P}^*))} \int_{V_m(\mathcal{H}(\mathbf{P'})) \bigcap V_n(\mathcal{H}(\mathbf{P}^*))} \eta \|\omega - p'_m\|^2 f(\omega) d\omega \\ &= \int_{V_m(\mathcal{H}(\mathbf{P'}))} \eta \left(\|\omega - p'_m\|^2 - \min_{n \in \mathcal{S}(\mathbf{P}^*)} \|\omega - p^*_n\|^2 \right) f(\omega) d\omega < 0 \end{aligned} \tag{30}$$

Consequently, $\mathbf{P'}$ is a better solution than $\mathbf{P}^*$, which contradicts our assumption[8].

[8]Remark: In this proof, we ignore two special cases: (i) p'_m is placed on top of another sensor, i.e., $p'_m = p^*_i, i \neq m$. In this case, p'_m should be moved towards p^0_m a little bit to avoid overlap without breaking the constraints, and then we will have the same contradiction. (ii) After replacing p^*_m by p'_m, more than one sensor joins the backbone network. In this case, the distortion at $\mathbf{P'}$ will be further reduced, and therefore we will have the same contradiction.

APPENDIX B
PROOF OF THEOREM 1

Let $\mathbf{R}^* = (R_1^*, \ldots, R_N^*)$ be the optimal partition and $\mathcal{I}^*$ be the optimal backbone network. For simplicity, let $c_n^* = \frac{\int_{R_n^*} \omega f(\omega) d\omega}{\int_{R_n^*} f(\omega) d\omega}$ and $v_n^* = \int_{R_n^*} f(\omega) d\omega$ be, respectively, the geometric centroid and the Lebesgue measure (volume) of R_n^*, $\forall n \in \mathcal{I}_\Omega$. Let $m \in \mathcal{I}^*$ be a specific sensor in the backbone network, $\mathbf{P}^*$. First, we assume that other sensor locations, $\{p_n^*\}_{n \in (\mathcal{I}_\Omega - \{m\})}$, are known, and then derive the constraint for p_m^*. Since $\mathcal{I}^*$ is the backbone network, we have[9]. $p_m^* \in \mathbb{D}_m(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*)$. Moreover, since Sensor m should satisfy the energy constraints (8), we have $p_m^* \in \mathbb{B}(p_m^0, \frac{\gamma_m}{\xi_m})$. In summary, Sensor m's location is restrained by

$$\mathbb{D}_m(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathbb{B}\left(p_m^0, R_c\right) = \mathbb{F}_m(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*). \tag{31}$$

Second, the minimum distortion can be rewritten as

$$\begin{aligned} D(\mathbf{P}^*) &= \sum_{n \in \mathcal{I}^*} \int_{\mathbf{R}_n^*} \eta_n \|p_n^* - \omega\|^2 f(\omega) d\omega & (32)\\ &= \int_{\mathbf{R}_m^*} \eta_m \|p_m^* - \omega\|^2 f(\omega) d\omega + \sum_{n \in (\mathcal{I}^* - \{m\})} \int_{\mathbf{R}_n^*} \eta_n \|p_n^* - \omega\|^2 f(\omega) d\omega & (33)\\ &= \eta_m \|c_m^* - p_m^*\|^2 v_m^* + \int_{\mathbf{R}_m^*} \eta_m \|c_m^* - \omega\|^2 f(\omega) d\omega + \sum_{n \in (\mathcal{I}^* - \{m\})} \int_{\mathbf{R}_n^*} \eta_n \|p_n^* - \omega\|^2 f(\omega) d\omega, & (34) \end{aligned}$$

where the third equation follows from the parallel axis theorem. Given the optimal partition $\mathbf{R}^*$ and optimal locations $\{p_n^*\}_{n \in (\mathcal{I}_\Omega - \{m\})}$, the second and third terms in (34) are constants. Therefore, p_m^* should be a minimizer of the first term with the constraint (31), i.e.,

$$p_m^* = \arg \min_{p_m : p_m \in \mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*)} \eta_m \|p_m - c_m^*\|^2 v_m^*. \tag{35}$$

Eq. (35) implies that the optimal solution should minimize the distance to c_n^* within $\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*)$. In what follows, we discuss 3 different cases of c_m^*.

(a) If $c_m^* \in [\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}(\mathbf{P}^*, \mathcal{I}^*)]$, we have $p_m^* \neq c_m^*$ because $p^* \in [\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c]$. However, replacing p_m^* by c_m^*, one can get a better solution $\mathbf{P}' = \left(p_1^*, \ldots, p_{m-1}^*, c_m^*, p_{m+1}^*, \ldots, p_N^*\right)$ which not only follows the energy constraints but also provides a smaller distortion. As a result, $c_m^* \in [\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}(\mathbf{P}^*, \mathcal{I}^*)]$ is an impossible case for the optimal deployment. In other words, we have the condition

$$c_m^* \notin \left[\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}(\mathbf{P}^*, \mathcal{I}^*)\right] \tag{36}$$

[9]The definitions of $\mathbb{D}_m(\mathbf{P}, \mathcal{I})$ and $\mathcal{W}(\mathbf{P}, \mathcal{I})$ and their relationships to backbone network are already provided in Section III

Since MWVD is the optimal partition, we have $\mathbf{R}^* = V(\mathbf{P}^*)$ and then

$$c_m^* = \frac{\int_{V_m(\mathbf{P}^*)} \omega f(\omega) d\omega}{\int_{V_n(\mathbf{P}^*)} f(\omega) d\omega} = c_m(\mathbf{P}^*), \forall n \in I_\Omega. \tag{37}$$

Moreover, the backbone network $\mathcal{I}^*$ is a function of deployment $\mathbf{P}^*$ and can be represented by

$$\mathcal{I}^* = \mathcal{S}(\mathbf{P}^*). \tag{38}$$

Substituting (37) and (38) to (36), we get condition (i).

(b) If $c_m^* \in [\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*)]$, Sensor m should be placed at c_m^*, i.e., $p_m^* = c_m^*$. Replacing c_m^* by (37) , we get the first case in condition (ii).

(c) If $c_m^* \notin \mathbb{F}(\mathbf{P}^*, \mathcal{I}^*)$, a simple geometric argument reveals that the optimal solution should be on the boundary of $\mathbb{F}(\mathbf{P}^*, \mathcal{I}^*)$. Therefore, the optimal location can be represented as $p_m^* = \arg \min_{q \in \partial[\mathbb{F}_m(\mathbf{P}^*, \mathcal{I}^*) \bigcap \mathcal{W}^c(\mathbf{P}^*, \mathcal{I}^*)]} \|q - c_m^*\|$. Replacing c_m^* and $\mathcal{I}^*$ by (37) and (38), respectively, we get the second case in condition (ii).

APPENDIX C
PROOF OF THEOREM 3

Let $\mathbf{P}^k = \left(p_1^k, \ldots, p_N^k\right)$ and $\mathbf{P}^k = \left(p_1^{k+1}, \ldots, p_N^{k+1}\right)$ be the current and next sensor deployment. Let m and n be two sensors such that they are MST neighbors at $\mathbf{P}^k$. Then, the distance between p_m^k and p_n^k is no larger than the communication range R_c, i.e., $\|p_m^k - p_n^k\| \leq R_c$. According to the definition of semi-desired region (19), we have $\mathbb{D}_m^s(\mathbf{P}) \subset \mathbb{B}\left(\frac{p_m^k + p_n^k}{2}, \frac{R_c}{2}\right)$ and $\mathbb{D}_n^s(\mathbf{P}) \subset \mathbb{B}\left(\frac{p_m^k + p_n^k}{2}, \frac{R_c}{2}\right)$. Moreover, m and n move within their semi-desired regions, i.e., $p_m^{k+1} \in \mathbb{D}_m^s\left(\mathbf{P}^k\right)$ and $p_n^{k+1} \in \mathbb{D}_n^s\left(\mathbf{P}^k\right)$. Thus, both p_m^{k+1} and p_n^{k+1} lie in the circle $\mathbb{B}\left(\frac{p_m^k + p_n^k}{2}, \frac{R_c}{2}\right)$. Therefore, the distance between p_m^{k+1} and p_n^{k+1} is no larger than the communication range, i.e., $\|p_m^{k+1} - p_n^{k+1}\| \leq R_c$. In other words, Sensors m and n are still connected with each other after the relocation. Consequently, the network at $\mathbf{P}^{k+1}$, $\mathcal{G}\left(\mathbf{P}^{k+1}\right)$, retains all edges in the MST at $\mathbf{P}^k$, indicating that the network, $\mathcal{G}\left(\mathbf{P}^{k+1}\right)$, is fully connected.

REFERENCES

[1] J. Guo and H. Jafarkhani, "Movement-efficient Sensor Deployment in Wireless Sensor Networks," *IEEE International Conference on Communications*, May 2018.

[2] C. Zhu, C. Zheng, L. Shu, and G. Han, "A survey on coverage and connectivity issues in wireless sensor networks," *Journal of Network and Computer Applications*, vol. 35, no. 2, pp. 619-632, Mar. 2012.

[3] B. Wang,"Coverage Problems in Sensor Networks: A Survey," ACM Computing Surveys, vol. 43, no. 32, Oct. 2011.

[4] M. R. Senouci, A. Mellouk, K. Asnoune, and F. Y. Bouhidel, "Movement-Assisted Sensor Deployment Algorithms: A Survey and Taxonomy," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2493-2510, 2015.

[5] J. Cortes, S. Martinez, T. Karatas, and F. Bullo, "Coverage Control for Mobile Sensing Networks," *IEEE Transactions on Robotics and Automation*, vol. 20, no. 2, Apr. 2004.

[6] J. Cortes, S. Martinez and F. Bullo, "Spatially-Distributed Coverage Optimization and Control with Limited-Range Interactions," *ESAIM: COCV* , vol. 11 , pp. 691-719, Oct. 2005.

[7] K. Sugimoto, T. Hatanaka, M. Fujita, and N. Huebel, "Experimental study on persistent coverage control with information decay," *Annual Conference of Society of Instrument and Control Engineers of Japan (SICE)*, pp. 164-169, 2015.

[8] M. T. Nguyen, L. Rodrigues, C. S. Maniu, and S. Olaru, "Discretized optimal control approach for dynamic Multi-Agent decentralized coverage," *IEEE International Symposium on Intelligent Control (ISIC)*, pp. 335-340, 2016.

[9] F. Li, J. Luo, S. Xin, and Y. He, "Autonomous deployment of wireless sensor networks for optimal coverage with directional sensing model," *Computer Networks*, vol. 108, pp. 120-132, Oct. 2016.

[10] Y. Song, B. Wang; Z. Shi, K. R. Pattipati, and S. Gupta, "Distributed Algorithms for Energy-Efficient Even Self-Deployment in Mobile Sensor Networks," *IEEE Transactions on Mobile Computing*, vol. 13, no. 5, pp. 1035-1047, May 2014.

[11] G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, and C. Gill, "Integrated Coverage and Connectivity Configuration for Energy Conservation in Sensor Networks," *ACM Transactions on Sensor Networks*, vol. 1, pp. 36-72, Aug. 2005.

[12] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. H. Lai, "Deploying Wireless Sensors to Achieve Both Coverage and Connectivity," *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pp. 131-142, Florence, Italy, 2006.

[13] H. Yousefi'zadeh, H. Jafarkhani, and J. Kazemitabar, "A study of connectivity in MIMO fading ad-hoc networks," *IEEE/KICS Journal of Communications and Networks*, vol. 11, no. 1, pp. 47–56, Feb. 2009.

[14] M. Moarref and L. Rodrigues, "An Optimal Control Approach to Decentralized Energy-Efficient Coverage Problems," *The International Federation of Automatic Control*, pp. 6038-6043, Aug. 2014.

[15] X. Liu, K. Wub, Y. Zhu, L. Kong, and M. Wua, "Mobility increases the surface coverage of distributed sensor networks," *Computer Networks*, vol. 57, no. 11, pp. 2348-2363, Aug. 2013.

[16] V. N. Had, "Decentralized Control of Three Dimensional Mobile Robotic Sensor Networks,", available at https://arxiv.org/abs/1606.00122.

[17] E. Koyuncu, R. Khodabakhsh, N. Surya, H. Seferoglu, "Deployment and trajectory optimization for UAVs: A quantization theory approach," available at arXiv:1708.08832.

[18] J. Guo, E. Koyuncu, and H. Jafarkhani, "A source coding perspective on node deployment in two-tier networks," *IEEE Transactions on Communications*, Early Access, 2018.

[19] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dbariwal and G. S. Sukhatme, "Robomote: Enabling Mobility in Sensor Network," *Proceedings of the 4th international symposium on Information processing in sensor networks*, no. 55, Apr. 2005.

[20] J. Wu and S. Yang, "Optimal Movement-Assisted Sensor Deployment and Its Extensions in Wireless Sensor Networks," *12th International Conference on Parallel and Distributed Systems*, Jul. 2006.

[21] W. Li, "On Wireless Sensors Covering and Movement Problem," *2007 International Conference on Wireless Communications, Networking and Mobile Computing*, pp. 2592-2595, Sept. 2007.

[22] M. R. Senouci and A. Mellouk, "Localized Movement-Assisted Sensor Deployment Algorithm for Hole Detection and Healing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 5, pp. 1267-1276, May 2014.

[23] Y. Zou and K. Chakrabarty, "Sensor deployment and target localization in distributed sensor networks," *ACM Transactions on Embedded Computing Systems*, vol. 3, no. 1, pp. 6191, 2004.

[24] M. Rout and R. Roy, "Self-Deployment of Mobile Sensors to Achieve Target Coverage in the Presence of Obstacles, *IEEE SENSORS JOURNAL*, vol. 16, no. 14, pp. 5837-5842, Jul. 2016.

[25] Z. Chen, X. Gao, F. Wu and G. Chen, "A PTAS to Minimize Mobile Sensor Movement for Target Coverage Problem, *2016 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1-9, Apr. 2016.

[26] A. N. Njoya, W. Abdou, A. Dipanda, and E. Tonye, "Evolutionary-based Wireless Sensor Deployment for Target Coverage," *International Conference on Signal-Image Technology & Internet-Based Systems*, pp. 739-745, Nov. 2015.

[27] Z. Liao, J. Wang, S. Zhang, J. Cao, and G. Min, "Minimizing Movement for Target Coverage and Network Connectivity in Mobile Sensor Networks," *IEEE Transactions on Parallel and Distributed Systems*, VOL. 26, no. 7, pp. 1971-1983, 2015.

[28] D. Z. Chen, Y. Gu, J. Li, H. Wang, "Algorithms on Minimizing the Maximum Sensor Movement for Barrier Coverage of a Linear Domain", *Discrete & Computational Geometry*, vol. 50, no. 2, pp. 374408, Jul. 2013.

[29] S. Li and H. Shen, "Minimizing the Maximum Sensor Movement for Barrier Coverage in the Plane," *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 244-252, May. 2015.

[30] Z. He, B. Zhang, "An Improved Algorithm for Minimizing the Maximum Sensor Movement in Linear Barrier Coverage," *Global Communications Conference (GLOBECOM), 2016 IEEE*, Dec. 2016.

[31] A. S. S and J. D. Beegum, "Enhancing The Life Time Of Sensor Network With Energy Awareness And Clutter Adaptability," International Conference on Control Communication & Computing India (ICCC), pp. 691-696, Mar. 2016.

[32] A. Okabe, B. Boots, K. Sugihara, and S. N. Chiu, *Spatial Tessellations: Concepts and Applications of Voronoi Diagrams*, 2nd ed., Wiley Series in Probability and Statistics. New York, NY: John Wiley & Sons, 2000.

[33] J. Guo and H. Jafarkhani, "Sensor Deployment with Limited Communication Range in Homogeneous and Heterogeneous Wireless Sensor Networks," *IEEE Transactions on Wireless Communications*, vol. 15, no. 10, pp. 6771-6884, Oct. 2016.

[34] E. Koyuncu and H. Jafarkhani, "On the Minimum Average Distortion of Quantizers With Index-Dependent Distortion Measures," *IEEE Transactions on Signal Processing*, vol. 65, no. 17, pp. 4655-4669, Sept. 2017.

[35] Y. Yoon and Y.-H. Kim, "An efficient genetic algorithm for maximum coverage deployment in wireless sensor networks," IEEE Transactions on Cybernetics, vol. 43, no. 5, pp.1473-1483, Oct. 2013.

[36] Z. Yong and W. Li, "A Sensor Deployment Algorithm for Mobile Wireless Sensor Networks," *Chinese Control and Decision Conference*, pp. 4642-4647, Jun. 2009.

[37] M. Gani, "Optimal deployment control for a heterogeneous mobile sensor network," *9th Int. Conf. Control, Autom., Robot. and Vision*, Dec. 2006.

[38] H. Yousefi'zadeh, H. Jafarkhani, and M. Moshfeghi, "Power Optimization of Wireless Media Systems with Space-Time Code Building Blocks," IEEE Transactions on Image Processing, Jul. 2004.

[39] G. Wang, M. J. Irwin, P. Berman, H.Fu, and T. L. Porta, "Optimizing Sensor Movement Planning for Energy Efficiency," *Proceedings of the 2005 International Symposium on Low Power Electronics and Design*, pp. 215-220, Oct. 2005.

[40] Y. Mei, Y. Lu, Y. C. Hu, and C.S. G. Lee, "Energy-Efficient Motion Planning for Mobile Robots," *Proceedings of the 2004 IEEE International Conference on Robotics & Automation*, pp. 4344-4349, Apr. 2004.

[41] B. Paul, *Kinematics and Dynamics of Planar Machinery* , 1st ed., Aurora, IL, U.S.A., Prentice-Hall, 1979.

[42] Robert G. Gallager, Pierre A. Humblet, and P. M. Spira, "A distributed algorithm for minimum-weight spanning trees," *ACM Transactions on Programming Languages and Systems*, vol. 5, no. 1, pp. 6677, Jan. 1983.

[43] W. H. Press, S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery, *Numerical Recipes: The Art of Scientific Computing*, 3rd ed., New York: Cambridge University Press, 2007.

[44] K. O. Geddes, S. R. Czapor, and G. Labahn, *Algorithms for Computer Algebra*, Boston, MA: Kluwer Academic Publishers, 1992.