Sensor Deployment with Limited Communication Range in Homogeneous and Heterogeneous Wireless Sensor Networks

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

Abstract

We study the heterogeneous wireless sensor networks (WSNs) and propose the necessary condition of the optimal sensor deployment. Similar to that in homogeneous WSNs, the necessary condition implies that every sensor node location should coincide with the centroid of its own optimal sensing region. Moreover, we discuss the dynamic sensor deployment in both homogeneous and heterogeneous WSNs with limited communication range for the sensor nodes. The purpose of sensor deployment is to improve sensing performance, reflected by distortion and coverage. We model the sensor deployment problem as a source coding problem with distortion reflecting sensing accuracy. Traditionally, coverage is the area covered by the sensor nodes. However, when the communication range is limited, a WSN may be divided into several disconnected sub-graphs. Under such a scenario, neither the conventional distortion nor the coverage represents the sensing performance as the collected data in disconnected sub-graphs cannot be communicated with the access point. By defining an appropriate distortion measure, we propose a Restrained Lloyd (RL) algorithm and a Deterministic Annealing (DA) algorithm to optimize sensor deployment in both homogeneous and heterogeneous WSNs. Our simulation results show that both DA and RL algorithms outperform the existing Lloyd algorithm when communication range is limited.

Index Terms

Sensor deployment, homogeneous, heterogeneous, source coding, coverage.

The authors are with Center for Pervasive Communications and Computing, University of California, Irvine (e-mail: guoj4@uci.edu; hamidj@uci.edu).
I. INTRODUCTION

As a bridge between the physical world and the virtual information world, wireless sensor networks (WSNs) collect data from the physical world and communicate it with the virtual information world, such as computers. Proper sensor deployment improves monitoring and controlling the physical environment. To accomplish their tasks, WSNs should address two needs: (i) Sensing in the target area and (ii) Communication between the sensor nodes. WSNs are utilized to collect physical information, such as temperature, humidity, voice and so on. But, the collected data is useless if it cannot be transmitted to the access point (AP) and to the outside information world through the AP node. When sensors are connected by wire lines, the connectivity is provided automatically. On the other hand, the connectivity of WSNs is not guaranteed. In this paper, we consider the sensing and connectivity together and redefine the goal of WSN design accordingly.

A huge body of literature exists on the topic of sensor deployment. The sensor coverage range model assumes that sensors can only monitor the points within a range of $R_s$. The range $R_s$ is called the sensing range and the coverage area is the area covered by at least one sensor node [1]. Three different connectivity criteria are proposed in [2]. Three movement-assisted protocols, the VECtor-based algorithm, the VORonoi-based algorithm and the Minimax algorithm, are designed to maximize coverage area in [3]. Lloyd algorithm has also been used as a tool to deploy sensors in homogeneous WSNs [1]. The convergence of the Lloyd algorithm has been studied in [4]–[6]. The analysis in [4]–[6] can be applied to the sensor deployment methods in [1]. However, [1] assumes an infinite communication range and ignores the connectivity limitation. When an infinite communication range is assumed, all the nodes in the network are connected to each other. In reality, each node has a limited communication range that will affect the connectivity of the network [7], [8]. A geometric analysis of the relationship between the sensing coverage and the connectivity is proposed in [9]. In [10], the authors have come up with some deployment patterns to achieve both sensing coverage and full connectivity.

Unfortunately, given a fixed number of sensor nodes and a finite communication range, connectivity is not guaranteed. When sensor nodes are divided into several disconnected subgraphs, the conventional Lloyd algorithm cannot converge to a proper deployment. The Critical
Sensor Density (CSD) in [11] is the number of nodes per unit area, required to provide full sensing coverage when the communication range is limited. When the sensor density is smaller than CSD, we cannot achieve the full sensing coverage. Under such a scenario, coverage may not be the right cost function to optimize. One needs to define an appropriate distortion measure to reflect the sensing accuracy. Distortion, as an important parameter in source coding can also be used to evaluate the WSN performance. Therefore, one can minimize distortion in WSNs through vector quantization techniques in [12] and [13]. The best possible distortion for a given number of sensors, i.e., the minimum distortion for a given rate, can be analyzed through the rate-distortion theory [?]. Even if the sensor density is larger than CSD, there is no existing sensor deployment algorithm designed to achieve the full connectivity and minimize the distortion at the same time. In this paper, when the communication range is limited, we take both distortion and connectivity into consideration. The existing coverage area model is a special case of our distortion measure. We propose a method, named Restrained Lloyd (RL) Algorithm, to distribute sensor nodes and to minimize distortion with full connectivity. Then, a more complex approach, named Deterministic Annealing (DA) Algorithm, is designed to avoid sub-optimal solutions.

In many practical situations, different sensors in the WSN have different characteristics such as computational power, sensing range, and sensing accuracy. The deployment and topology control of such heterogeneous WSNs that include the sensor nodes with different communication or sensing ranges, have been studied in [15] and [16]. Similar to [3], three movement-assisted protocols in [15] are designed to avoid coverage hole in heterogeneous WSNs. However, [16] deploys sensor nodes one-by-one and to deploy a new sensor uses the location information of all previously deployed nodes. Also, [16] assumes that sensor can monitor events within a circle with the radius equal to the sensing range. We generalize this model to a sensing accuracy which depends on the distance between the sensor and the event. Our distortion model will include the sensing range model in [16] as a special case in which the distortion is a step function. In such heterogeneous WSNs, weighted Voronoi diagrams [17] rather than conventional Voronoi diagrams [18] will provide the best regions as we will discuss in this paper. An algorithm to construct weighted Voronoi diagrams for a different application has been suggested in [19]. Based on the geometry of the optimal cell partitioning, we will analyze the objective functions in our
model and propose the necessary condition for the optimal sensor deployment in heterogeneous WSNs with different sensing abilities.

In the rest of this paper, we first introduce the system model for both homogeneous and heterogeneous WSNs and formulate the problems of sensing and connectivity in Section II. Section III analyzes the optimal deployment in heterogeneous sensor networks without communication constraint. Section IV proposes RL and DA algorithms to improve distortion and maintain connectivity. Section V presents simulation results and Section VI provides the conclusions.

II. SYSTEM MODEL AND GENERAL PROBLEMS

Let $Q$ be a simple convex polygon in $\mathbb{R}^2$ including its interior. Given $n$ sensors in the target area $Q$, sensor deployment is defined by $P = (p_1, \cdots, p_n) \subset Q^n$, where $p_i$ is Sensor $i$’s location. For any point $q \in Q$, $\lambda(q)$ is the probability density function of an event at point $q$. A cell partition $R$ of $Q$ is a collection of disjoint subsets of $\{R_i(P)\}_{i \in 1, \cdots, n}$ whose union is $Q$. Let $B(c, r) = \{q | \|q - c\| \leq r\}$ be a disk centered at $c$ with radius $r$ in two-dimensional space. For two points $a$ and $b$, let equation $Eq + F = 0$, where $E \in \mathbb{R}^{2\times 2}$ is a $2 \times 2$ matrix and $F \in \mathbb{R}$ is a constant, define the perpendicular bisector hyperplane between the two points. Then, the equations $Eq + F \geq 0$ and $Eq + F \leq 0$ define two half spaces. we denote the half space that contains point $a$ by $HS(a, b)$.

As mentioned before, we define the AP as the sensor node that can communicate with the outside information world. Let $S(P)$ be the set of sensor nodes that can communicate with the AP when the sensor deployment is $P$. Note that in general not all nodes can communicate with the AP and $\text{card}(S(P)) \leq n$, where $\text{card}(A)$ is the number of elements in set $A$. We define a new sensor deployment, which is a subset of the all sensor locations, $H(P)$ as the vector of sensor locations for the $\text{card}(S(P))$ sensor nodes connected to the AP. When $S(P)$ includes all sensor nodes, we have $P = H(P)$ and $\text{card}(S(P)) = n$. Let $T$ be the set of sensor deployments that provide full connectivity, i.e., $T = \{P | \text{card}(S(P)) = n\}$. In our model, two sensor nodes can communicate with each other 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. A sensor node can transfer data outside if and only if there exists a path from the sensor to the AP. The path consists of a
sequence of sensor nodes where each hop distance is smaller than the communication range $R_c$. Sensor nodes that are connected to the AP construct the backbone network. 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.

Another important factor in analyzing the performance of a WSN is its sensing accuracy. Ideally, we would like to sense all events in the covered area. However, the sensing accuracy of a sensor node usually depends on the distance between the sensor and the event to be sensed. In other words, the accuracy of the gathered data from an event at point $q$ by its associated sensor node $i$ is a non-increasing function of the distance between $q$ and $p_i$. Therefore, to represent the average sensing accuracy in the target area, we define the following general distortion:

$$D(P) = \sum_{i=1}^{n} \int_{R_i(P)} \Phi_i(\|q - p_i\|) \lambda(q) dq,$$

where $\Phi_i(\cdot)$ is the cost function associated with sensing. The partition $\{R_i(P)\}_{i=1,...,n}$ in the above definition include all sensor nodes. However, as explained before, when the communication range is limited, some sensor nodes cannot transfer their data back. As a result, only the sensor nodes in the backbone network can contribute to the sensing and therefore the distortion should be revised as

$$D(P) = \sum_{i \in S(P)} \int_{R_i(H(P))} \Phi_i(\|q - p_i\|) \lambda(q) dq.$$

Note that to derive Eq. (2) from Eq. (1), one has to replace $P$ with $H(P)$, i.e., one has to consider only the sensor nodes that are in the backbone network. We reiterate that in the case of a fully connected network, $H(P) = P$ and Eq. (1) and Eq. (2) are identical.

Obviously, choosing different cost functions in Eq. (2) results in different problem formulations. One natural choice for the cost function is a continuous function defined by

$$\Phi_i(x) = \eta_i x^2,$$

where the cost parameter $\eta_i \in R^+$ is a constant that depends on the sensor characteristics. In homogeneous WSNs, every sensor node has the same sensing ability and the same cost parameter $\eta_i$. Therefore, the cost parameter can be ignored in homogeneous WSNs. However,
different sensors with different complexity, power and sensing ability are used in heterogeneous WSNs. Obviously, the cost parameters \( \{\eta_i\}_{i \in 1, \ldots, n} \) reflect the quality of sensor nodes. The smaller the cost parameter, the stronger the sensing ability.

The distortion definitions in Eqs. (1) and (2) can represent the sensor coverage area model as well. In such a model, the sensors can monitor events within a circle with a fixed radius called the sensing range. Consider a step function defined by

\[
I_i(x) = \begin{cases} 
1, & \text{for } \eta_i x^2 < R_s^2 \\
0, & \text{for } \eta_i x^2 \geq R_s^2.
\end{cases}
\] (4)

Adopting the above sensor coverage area model, choosing \( \Phi_i(x) = I_i(x) \) in Eq. (2) converts the distortion to the area covered by the sensors. In such a model, sensors can monitor events within a circle with the radius \( \frac{R_s}{\sqrt{\eta_i}} \). Obviously, the coverage area should be maximized while the distortion should be minimized. Since by definition \( \Phi_i(x) \) should be a non-decreasing function, to have the sensor coverage area model as a special case of our model, we should choose \( \Phi_i(x) = 1 - I_i(x) \).

Our main goal is to minimize the distortion defined in Eq. (2). It is easy to show that a necessary condition for such an optimal sensor deployment is to have a fully connected network. Moreover, the distortion is determined by both sensor deployment and cell partitioning. This is the topic of the discussion in the next section.

III. Optimal Deployment in Heterogeneous WSNs without Communication Constraint

In this section, we assume an infinite communication range which results in a connected network for any sensor deployment. Given a fully connected network, the distortion \( D(P) = D(P, R(P)) \) is determined by the sensor deployment \( P \) and the cell partition \( R(P) \). In homogeneous WSNs, given sensors’ locations, Voronoi partitions provide the smallest distortion. The Voronoi region (partition) for Sensor \( i \), denoted by \( V_i(P) \), is the intersection of half spaces \( HS(p_i, p_j), \forall j \neq i \). In other words, the Voronoi region for Sensor \( i \) is the set of all points that
are closer to Sensor $i$ than any other sensor. The Voronoi partition of $Q$ generated by $P$ with respect to the Euclidean norm is the collection of sets $\{V_i(P)\}_{i \in 1, \ldots, n}$ defined by

$$V_i(P) = \{ q \in Q \mid \|q - p_i\| \leq \|q - p_j\|, \forall j \in 1, \ldots, n \},$$

(5)

where $\| \cdot \|$ is the Euclidean norm. This is because all sensors in homogeneous WSNs have the same cost function $\Phi(x)$ which is only determined by the Euclidean distance $x = \|q - p_i\|$. Since an event in $V_i(P)$ is monitored by Sensor $i$, each event is sensed by the nearest sensor and therefore makes the smallest contribution to the global distortion.

However, in a heterogeneous WSNs, the cost function $\Phi_i(x)$ is also affected by the cost parameter $\eta_i$, which reflects the sensing ability of Sensor $i$. Sensor nodes in heterogeneous WSNs can be classified according to their cost parameters $\{\eta_i\}_{i \in 1, \ldots, n}$. Let us assume there are $m$ different sensor types with $m$ different cost parameters. Given sensor nodes’ locations, an event at point $q$ should be sensed by the sensor with the smallest cost such that its contribution to the global distortion is the lowest possible. Such an optimal partitioning is refer to as the weighted Voronoi partitioning. The weighted Voronoi partition of $Q$ generated by $P$ is the collection of sets $V_i^H(P)_{i \in 1, \ldots, n}$ defined by

$$V_i^H(P) = \{ q \in Q \mid \eta_i \|q - p_i\|^2 \leq \eta_j \|q - p_j\|^2, \forall j \in 1, \ldots, n \}.$$

(6)

Note that both Voronoi regions $V_i(P)_{i \in 1, \ldots, n}$ and weighted Voronoi regions $V_i^H(P)_{i \in 1, \ldots, n}$ are functions of $P$. Since the Voronoi partitioning can be considered as a special case of the weighted Voronoi partitioning, in which $\eta_i = 1, i = 1, \ldots, n$, we simply use $V_i^H(P)_{i \in 1, \ldots, n}$ to represent both. Using the result of weighted Voronoi partitioning as the partition in Eq. (1), the global distortion can be rewritten as

$$D(P) = \sum_{i=1}^{n} \int_{V_i^H(P)} \eta_i \|q - p_i\|^2 \lambda(q) dq.$$  

(7)

Our goal is to find the sensor deployment that minimizes the global distortion. In what follows, we generate the machinery to find the necessary condition of the optimal sensor deployment.

**Proposition 1.** Consider a two-dimensional heterogeneous sensor network, in which the cost
function between Sensor \( k \) and point \( q \) is \( \eta_k \|q - p_k\|^2 \), the optimal partition for a given deployment \( P \) is

\[
V_k^H(P) = \{ q \in Q | \eta_k \|q - p_k\| \leq \eta_t \|q - p_t\|^2, \forall t \in 1, \ldots, n \}
\]

\[
= \left[ \bigcap_{i: \eta_i < \eta_k} B(c_{ik}, r_{ik}) \right] \bigcap \left[ \bigcap_{t: \eta_t = \eta_k} H_k(p_k, p_t) \right] - \bigcup_{j: \eta_j > \eta_k} B(c_{kj}, r_{kj}) \right), \tag{8}
\]

where \( c_{ij} = \frac{p_i - (\eta_i/\eta_j)p_j}{1 - (\eta_i/\eta_j)} \) and \( r_{ij} = \sqrt{\frac{\eta_i}{\eta_j}} \|p_i - p_j\| \).

**Proof:** The proof is provided in Appendix A.

Before we discuss the optimal sensor deployment in heterogeneous WSNs, we need to present the following definitions and lemmas.

**Definition 1.** A set \( S \subseteq \mathbb{R}^n \) is called star-shaped if and only if there exists a point \( p \in \text{int}(S) \) such that for all \( s \in \partial S \) and all \( \lambda \in (0, 1] \), one has \( \lambda p + (1 - \lambda)s \in \text{int}(S) \), where \( \text{int}(S) \) is the interior of \( S \) and \( \partial S \) is the boundary of \( S \). The point \( p \) is the reference point.

**Definition 2.** A set \( S \subseteq \mathbb{R}^n \) is called a convex region if and only if for every pair of points \( x, y \in S \) and all \( \lambda \in (0, 1) \), one has \( \lambda x + (1 - \lambda)y \in \text{int}(S) \).

**Lemma 1.** If a set \( S \subseteq \mathbb{R}^n \) is convex, then \( S \) is star-shaped.

**Proof:** For any convex region \( S \subseteq \mathbb{R}^n \), pick a point \( p \in \text{int}(S) \subset S \). For any point \( s \in \partial S \subset S \) and all \( \lambda \in (0, 1) \), one has \( \lambda p + (1 - \lambda)s \in S \). When \( \lambda = 1 \), \( \lambda p + (1 - \lambda)s = p \in \text{int}(S) \). Therefore, the set \( S \) is star-shaped.

We will use the fact that the intersection of any collection of convex sets is convex [20], [21] and as a result star-shaped according to Lemma 1.

**Lemma 2.** The union of star-shaped sets that are associated with the same reference point \( p \) is star-shaped.

**Proof:** Given \( m \) star-shaped sets \( S_i, i = 1, \ldots, m \) with the same reference point \( p \), the corresponding union is \( S = \bigcup_{i=1}^m S_i \). Since point \( p \) is the reference point of star-shaped sets \( S_i \), where \( i \in \{1, \ldots, m\} \), we have \( p \in \bigcap_{i=1}^m S_i \) and therefore \( \bigcap_{i=1}^m S_i \neq \emptyset \). The boundary of \( S \)
comes from the boundaries of \( m \) star-shaped sets \( S_i \), where \( i \in \{1, \cdots, m\} \). Thus, for any point \( s \in \partial S \), we have \( s \in \bigcup_{i=1}^{m} \partial S_i \). Because of \( m \) star-shaped sets, for all \( s \in \partial S \) and all \( \lambda \in (0, 1] \), we will have \( \lambda p + (1 - \lambda) s \in \text{int}(S) \). Thus, for all \( s \in \partial S \) and for all \( \lambda \in (0, 1] \), one can find a subset \( S_i \) such that \( s \in \partial S_i \) and so \( \lambda p + (1 - \lambda) s \in \text{int}(S_i) \subset \text{int}(S) \).

\[
\text{Lemma 3. Let } S(x) = \bigcup_{i=1}^{m} S_i(x) \text{ be a star-shaped set that consists of } m \text{ disjoint sub-sets } S_i, \text{ where } i \in \{1, \cdots, m\}. \text{ We then have}
\[
\int_{\partial S} \varphi(\gamma)n^t(\gamma)d\gamma = \sum_{i=1}^{m} \int_{\partial S_i} \varphi(\gamma)n^t(\gamma)d\gamma,
\]
\[
\text{where } \varphi(\cdot) \text{ is a continuous function of } \gamma, \text{ and } n^t(q) \text{ is the unit outward normal to } \bigcup_{i=1}^{m} \partial S_i \text{ at } q.
\]

\[
\text{Proof:}
\]
\[
\sum_{i=1}^{m} \int_{\partial S_i} \varphi(\gamma)n^t(\gamma)d\gamma = \sum_{i=1}^{m} \left[ \sum_{j \neq i} \int_{\partial(S_i \cap S_j)} \varphi(\gamma)n^t(\gamma)d\gamma + \int_{\partial(S_i \cap S)} \varphi(\gamma)n^t(\gamma)d\gamma \right]
\]
\[
\text{For any } i \text{ and } j \text{ such that } S_i \cap S_j = \emptyset, \text{ the corresponding curve integral } \int_{\partial(S_i \cap S_j)} \varphi(\gamma)n^t(\gamma)d\gamma \text{ is 0. On the other hand, for any } i \text{ and } j \text{ such that } S_i \cap S_j \neq \emptyset, \text{ the corresponding curve integral}
\]
\[
\int_{\partial(S_i \cap S_j)} \varphi(\gamma)n^t(\gamma)d\gamma = -\int_{\partial(S_j \cap S_i)} \varphi(\gamma)n^t(\gamma)d\gamma \text{ because of opposite unit outward normal. Therefore, we have}
\]
\[
\sum_{i=1}^{m} \int_{\partial S_i} \varphi(\gamma)n^t(\gamma)d\gamma = \sum_{i=1}^{m} \left[ \int_{\partial(S_i \cap S)} \varphi(\gamma)n^t(\gamma)d\gamma \right] = \int_{\partial S} \varphi(\gamma)n^t(\gamma)d\gamma
\]

\[
\text{Lemma 4. Let } S = \bigcup_{i=1}^{m} S_i \text{ be a star-shaped set that consists of } m \text{ disjoint subsets } S_i, i = 1, \cdots, m. \text{ Then for any point } p \in S \text{ we have}
\]
\[
(p - C_S)M_S = \sum_{i=1}^{m} (p - C_{S_i})M_{S_i},
\]
\[
\text{where } M_S = \int_S \lambda(q)dq \text{ and } C_S = \frac{\int_S q\lambda(q)dq}{M_S} \text{ are, respectively, the mass and the center of mass with respect to the probability density function } \lambda(\cdot) \text{ of the set } S.
\]
Proof: We rewrite the left side of Eq. (11) to derive

\[(p - C_S)M_s = pM_s - \sum_{i=1}^{m} \int_{S_i} q\lambda(q) dq\]

\[= \sum_{i=1}^{m} \left[ pM_{S_i} - \int_{S_i} q\lambda(q) dq \right] \]

\[= \sum_{i=1}^{m} (p - C_{S_i})M_{S_i} \quad (12)\]

Now, we have enough tools to derive the main results in this section. Proposition A.1. in [1] presents how to calculate the gradient of the distortion when sensing regions are star-shaped. Unfortunately, it is possible that the sensing regions in heterogeneous WSNs are not star-shaped. In what follows, we show how to calculate the gradient of the distortion in heterogeneous WSNs.

Proposition 2. In a heterogeneous sensor network including \( m \) kinds of sensors, let \( P = [p_1, p_2, \ldots, p_n] \) be the sensor deployment, and \( W \in \mathbb{R}^2 \) be an arbitrary convex set. Let a series of functions \( \varphi_i : \mathbb{R}^2 \times (a, b) \rightarrow \mathbb{R} \), where \( i = 1, \ldots, n \), be continuous on \( \mathbb{R}^2 \times (a, b) \), continuously differentiable with respect to its second argument for all \( p_i \in (a, b)^2 \), where \( i = 1, \ldots, n \), and almost all \( q \in \mathbb{R}^2 \), and such that for each \( p_i \in (a, b)^2 \), the maps \( q \mapsto \varphi_i(q, p_i) \) and \( q \mapsto \frac{\partial \varphi_i}{\partial x}(q, x) \) are measurable, and integrable on \( \mathbb{R}^2 \). Then the function

\[\int_{V^H_i(P) \cap W} \varphi_i(q, p_i) dq\]  

is continuously differentiable and

\[\frac{\partial \int_{V^H_i(P) \cap W} \varphi_i(q, p_i) dq}{\partial p_j} = \int_{V^H_i(P) \cap W} \frac{\partial \varphi_i}{\partial p_j}(q, p_i) dq + \int_{\partial [V^H_i(P) \cap W]} \varphi_i(\gamma, p_i)n^i(\gamma) \frac{\partial \gamma}{\partial p_j} dq. \quad (14)\]

Proof: The proof is provided in Appendix B.

Note that the sensing cell \( V^H_i(P) \) in Proposition 2 is a weighted Voronoi region and different from the Voronoi region in [1]. The weighted Voronoi region \( V^H_i(P) \) can be a non-star-shaped region and therefore we need to use Proposition 2.
Next, we derive the necessary condition for the optimal deployment in heterogeneous WSNs when the communication range is infinite. The format of the result, as proved in the next proposition, is similar to that of homogeneous WSNs.

**Proposition 3.** When the communication range is infinite, the necessary condition for the optimal deployment in heterogeneous WSNs is

\[
p_j^* = c_j(P), \forall j \in \{1, \cdots, n\},
\]

where \(p_j^*\) is the optimal position for sensor node \(j\), \(M_j(P) = \int_{V_j^H(P)} q \lambda(q) dq\) and \(c_j(P) = \int_{V_j^H(P)} q \lambda(q) dq / M_j(P)\) are, respectively, the mass and the center of weighted Voronoi cell \(V_j^H(P)\) with respect to the probability density function \(\lambda(\cdot)\) in target region \(Q\).

**Proof:** Let \(W = \mathbb{R}^2\) and \(\varphi_i(q, p_i) = \Phi_i(\|q - p_i\|)\lambda(q)\), where \(\Phi_i(\cdot)\) is the cost function, we can use Proposition 2 to calculate the partial derivative of the local distortion as follows:

\[
\frac{\partial}{\partial p_j} \int_{V_i^H(P)} \eta_i \|q - p_i\|^2 \lambda(q) dq = \begin{cases} 
2\eta_i(p_i - c_i) M_i + \int_{\partial V_i^H(P)} \eta_i \|\gamma - p_i\|^2 n^t(\gamma) \frac{\partial}{\partial p_i} n(\gamma) d\gamma, & i = j \\
\int_{\partial V_i^H(P)} \eta_i \|\gamma - p_i\|^2 n^t(\gamma) \frac{\partial}{\partial p_j} n(\gamma) d\gamma, & i \neq j
\end{cases}
\]

where \(\frac{\partial}{\partial p_i} \neq 0\) if and only if \(\gamma\) is on the boundary of \(V_j^H(P)\).

Note that \(\partial V_i^H(P) = \bigcup \partial (V_i^H(P) \cap V_j^H(P))\), \(\forall l \neq i\). Thus, in Eq. (16) only the curve integral for \(j\), i.e., on \(\partial (V_i^H(P) \cap V_j^H(P))\), needs to be taken into account. For each curve integral in the second case \((i \neq j)\), one can find a curve integral with the opposite unit outward normal in the first case \((i = j)\). As a result, the curve integrals in the partial derivative of the global distortion are canceled with each other. Therefore,

\[
\frac{D(P)}{\partial p_j} = \sum_{i=1}^{n} \frac{\partial}{\partial p_j} \int_{V_i^H(P)} \eta_i \|q - p_i\|^2 \lambda(q) dq = 2\eta_j(p_j - c_j(P)) M_j(P), j = 1, \cdots, n.
\]

Both \(M_j(P)\) and \(c_j(P)\) are functions of \(P\). The optimal deployment \(P^*\) will have a zero gradient. We define the cost parameters to be positive. Thus, when the communication range is infinite, the necessary condition for optimal deployment is the same as Eq. (15).
In this section, we design algorithms to minimize the distortion when the communication range is limited. First, we quickly review the conventional Lloyd algorithm. Lloyd Algorithm has two basic steps in each iteration: (1) Sensor nodes move to their centroid; (2) Partitioning is done by assigning the corresponding Voronoi region to each sensor node. Lloyd Algorithm provides good performance and is simple enough to be implemented distributively. It converges to a minimum distortion when the communication range is infinite [1]. Unfortunately, it also has three shortcomings. First, since minimizing distortion is a non-convex optimization problem, Lloyd Algorithm may end at a large local minimum point rather than the optimal global minimum. Second, Lloyd Algorithm may not result in a connected network. Third, when WSNs are divided into several disconnected sub-graphs, Lloyd Algorithm is not feasible. In other words, since there is no global information available about the sensor locations, each sub-graph will run the algorithm independently. To deploy a network with full connectivity and lower distortion, we add some restraints on sensors’ movements. We design a class of algorithms based on the Lloyd algorithm, referred to as RL Algorithm.

A. Restrained Lloyd Algorithm

Before we introduce the details of our RL Algorithm, we introduce the concept of a desired region. Let us assume we are trying to move Sensor $i$ at a given step. Our goal is to keep the connectivity of the backbone network after moving Sensor $i$. Therefore, we define the areas in which Sensor $i$ will be connected to the backbone network as its desired region, denoted by $L_i(P)$. Note that this region may not be a star-shaped set. In our RL Algorithm, if Sensor $i$ is in the backbone network, we will restrain its movement within its desired region. To achieve this goal, we need to find the desired region $L_i(P)$. Given a deployment $P$, if Sensor $i$ from the backbone network is removed, the rest of the sensor nodes in the backbone network will be divided into $K_i$ components: $U_{i1}(P), U_{i2}(P), \cdots, U_{iK_i}(P)$, where $U_{ij}(P)$ is a set of sensors included in the $j$th component. Note that $K_i$ may be equal to one. Then, we can calculate the
desired region as
\[ L_i(P) = \bigcap_{k=1}^{K_i} \left[ \bigcup_{j \in U_{ik}} B(p_j, R_c) \right]. \]  
(18)

Since the desired region is primarily influenced by the neighboring sensor nodes, we can approximate it by
\[ \tilde{L}_i(P) = \bigcap_{k=1}^{K_i} \left[ \bigcup_{j \in U_{ik} \cap N_i(P)} B(p_j, R_c) \right], \]  
(19)
where \( N_i(P) \) consists of Sensor \( i \)'s neighbors when the deployment is \( P \). Note that the approximation in (19) can be calculated locally, but to calculate the exact desired region, one needs global information. Also, according to Lemma 2, the approximate desired region \( \tilde{L}_i(P) \) is a star-shaped set.

Now, we provide the details of our RL Algorithm. The algorithm iterates between two steps:

1. Sensors in the backbone network move one by one. Every sensor in the backbone network calculates its own approximate desired region \( \tilde{L}_i(P) \) and moves to a location which is the closest point to its centroid \( c_i(P) \) within \( \tilde{L}_i(P) \). Sensors outside the backbone network move randomly and check if there is a path to the AP. Unlike the conventional Lloyd algorithm, these new locations may not be the centroid of the partition regions;
2. The target area, \( Q \), is partitioned to weighted Voronoi regions for sensors in the backbone network, \( S(P) \).

The main difference between RL Algorithm and the Lloyd algorithm is in the first step. In what follows, we show that Step (1) in RL Algorithm will not increase the global distortion.

The global distortion is the sum of local distortions defined by
\[ D_i(P) = \int_{V_i^{\mu(H(P))}} \|q - p_i\|^2 \lambda(q) dq, \quad i \in \{1, \ldots, m\}, \]  
(20)
where \( m \) is the number of sensors in the backbone network. The local distortion, whether in homogeneous WSNs or heterogeneous WSNs, is a convex function. According to the parallel
axis theorem, the local distortions can be rewritten as

\[
D_i(P) = \int_{V_i^H(H(P))} ||q - c_i(P)||^2 \lambda(q) dq + \int_{V_i^H(H(P))} \lambda(q) dq \cdot ||p_i - c_i(P)||^2, \quad i \in \{1, \cdots, m\},
\]

(21)

where \(c_i(P) = \frac{1}{\int_{V_i^H(H(P))} \lambda(q) dq} \int_{V_i^H(H(P))} q \lambda(q) dq\) is the centroid of the partition region \(V_i^H(H(P))\) with respect to the probability density function. Both \(\int_{V_i^H(H(P))} ||q - c_i(P)||^2 \lambda(q) dq\) and \(\int_{V_i^H(H(P))} \lambda(q) dq\) are constants when the integral area \(V_i^H(H(P))\) is fixed. In other words, the local distortion is directly proportional to \(||p_i - c_i(P)||^2\), i.e., the sensor’s distance to its centroid. Therefore, the movements in Step (1) minimize the local distortions. As the sum of local distortions, the global distortion will not increase. Since the sequence of the global distortion values is a non-increasing sequence with a lower bound of zero, it will converge.

We also show that our RL Algorithm guarantees the connectivity of the network with high probability after enough number of iterations. Note that once a sensor node finds a path to the AP, our RL Algorithm will keep it in the backbone network. Intuitively, as we have more iterations, the sensors outside the backbone network will move randomly and eventually connect to the AP as well. Quantitatively, for the deployment after \(k\) iterations, the area in which a sensor can communicate with the backbone network can be calculated by \(A_k = \text{AREA} (Q \bigcap \bigcup_{i \in \text{backbone}} B(p_i, R_c))\). Then, the probability that a sensor outside the backbone network is not connected to the AP in its next move is \(\frac{\text{AREA}(Q) - A_k}{\text{AREA}(Q)}\). After \(N\) iterations, the probability that a sensor is still out of the backbone network can be calculated by \(P_{out}(N) = \prod_{k=1}^{N} \left[\frac{\text{AREA}(Q) - A_k}{\text{AREA}(Q)}\right] \leq \left[\frac{\text{AREA}(Q) - \min A_k}{\text{AREA}(Q)}\right]^N\) and then \(\lim_{N \to \infty} P_{out}(N) = 0\) because of \(\min A_k > 0\). In other words, as long as the number of iterations is large enough, almost all sensor nodes will be included in the backbone network, indicating full connectivity, with high probability.

B. Deterministic Annealing Algorithm

Like any other steepest-descent algorithm, RL Algorithm may converge to a local minimum a large distortion. One approach to improve the sub-optimal solution or find the global optimal solution, is to use annealing methods. Simulated Annealing (SA) \cite{22}, \cite{23} is a method in which a candidate sensor movement is generated randomly. However, SA ignores the characteristics.
of the objective function and requires burdensome computations. In this paper, we design a DA algorithm which combines RA with annealing to minimize the distortion. Unlike SA, the proposed DA generates two new sensor positions deterministically at each iteration; however, choose one of the two options randomly. Like RL Algorithm, our DA Algorithm iterates between two steps. The second step is identical to that of RL Algorithm. In the first step, the algorithm creates two candidate locations for each node in the backbone network. One candidate is the RL Algorithm’s candidate that minimizes the local distortion. On the other hand, the second candidate increases the local distortion. It is easy to show that to maximize the local distortion for Sensor $i$ in the backbone network, one should move it to the point $o$ on the boundary of the desired region $\tilde{L}_i(P)$ that has the largest distance to the centroid $c_i(P)$. But the goal of the second candidate is to increase the distortion and not necessarily maximizing it. Moreover, the distortion is more sensitive to the sensors with smaller cost parameters. In order to avoid increasing distortion too fast, Sensor $i$ moves to the point $p_i + \frac{\eta_i}{\min_j(\eta_j)}(o - p_i)$. The algorithm will choose the first candidate with probability $p$ and increases $p$ from 0 to 1. Otherwise, the algorithm will choose the second candidate. In our algorithm, the probability $p$ is increased in proportion to $\log k$, where $k$ is the iteration count and for the last $M$ iterations we force the probability $p = 1$. Like RL Algorithm, DA Algorithm guarantees connectivity and convergence. The proof is similar to that of RL Algorithm and is omitted.

V. PERFORMANCE EVALUATION

We compare the performance of RL Algorithm, DA Algorithm and Lloyd Algorithm in sensor networks. We provide simulations in three sensor networks: (1) WSN1: A homogeneous WSN in which all sensors have the same cost parameter $\eta_i = 1, i = 1, \cdots, 16$; (2) WSN2: A heterogeneous WSN including 2 kinds of sensors: four strong sensors with $\eta_i = 1$ and twelve weak sensors with $\eta_j = 16$; (3) WSN3: A heterogeneous WSN including three kinds of sensors: two strong sensors with $\eta_i = 1$, four medium sensors with $\eta_j = 4$ and ten weak sensors with $\eta_k = 16$. Sixteen sensors are provided in each sensor network. The AP is chosen from the sixteen sensors randomly. However, when we report the distortion or coverage area for the Lloyd algorithm, we report that of the largest connected subgraph which may not be connected to the
AP. Obviously, this will be advantageous for the Lloyd algorithm, but our proposed algorithms still outperform the Lloyd algorithm. We use ten random initial deployments for each algorithm. To have a fair comparison, we consider the same target domain \(Q\) as in [1]. \(Q\) 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). The distribution of the events is also the same as [1]. The probability density function is the sum of five Gaussian functions of the form \(5\exp(6(- (x-x_{center})^2 - (y-y_{center})^2))\). 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). We use 0.5 as the communication range \(R_c\). Also, when reporting the coverage area using (4), we use 0.25 as the sensing range \(R_s\). In DA Algorithm, the first candidate is accepted at the \(i\)th iteration by a probability of \(p(i) = \log(i + 1)/\log(N + 1)\), where \(N\) is the number of regular iterations. Additional \(M = 25\) iterations are used in DA Algorithm to avoid ending with a process that increases the local distortions.

Figs. 1a and 1b show one example of the initial and the final deployments of Lloyd Algorithm in WSN1. Lloyd Algorithm assumes an infinite communication range and requires the global knowledge of the sensor locations. Otherwise, disconnected sub-graphs run Lloyd Algorithm independently and there is no guarantee for convergence. Nonetheless, the calculation of the final distortion only considers sensors in the backbone network. In the final deployment of the example in Fig. 1b, there are four sensors disconnected from the backbone network, resulting in a large distortion \(D(P) = 2.21\). Fig. 1c shows the outcome of RL Algorithm in WSN1. After 500 iterations, the distortion is decreased from 11.30 to 0.60. Simultaneously, the coverage area is increased from 0.15 to 6.26 and the final deployment is connected. Fig. 1d shows the final deployment of DA Algorithm in WSN1. After 500 iterations, the distortion is decreased from 11.30 to 0.32, which is better than that of RL Algorithm. Simultaneously, the coverage area is increased from 0.15 to 6.99 and full connectivity is provided. Unlike Lloyd Algorithm, both RL Algorithm and DA Algorithm guarantee connectivity.

Fig. 2 illustrates the performance of the above algorithms for 10 random initial deployments. As can be seen from the figure, unlike other algorithms, the performance of DA Algorithm is not sensitive to the initial deployment. In other words, DA Algorithm avoids most poor local minimum solutions. Fig. 2 shows that DA Algorithm has the best performance among the three
Fig. 1. Sensor deployments in WSN1. (a) The initial sensor deployment and the corresponding Voronoi regions. (b) The final deployment of Lloyd Algorithm after 500 iterations. (c) The final deployment of RL Algorithm after 500 iterations. (d) The final deployment of DA Algorithm after 500 iterations. Sensors in the backbone network are marked by circles. Sensors disconnected from the backbone network are denoted by dots. Voronoi region centroids are marked by stars. The radius of each gray circle is $R_c/2 = 0.25$.

algorithms.

Fig. 3 compares the final coverage area of RL Algorithm and DA Algorithm with that of Lloyd Algorithm. In most cases, decreasing the distortion results in increasing the coverage area as well. Intuitively, this behavior can be explained by considering coverage area as a hard-decision version of distortion. Next, the relationship between performance (distortion and coverage area) and communication range $R_c$ in homogeneous WSN1 using DA Algorithm is depicted in Fig. 4.

Figs. 5a and 5b show one example of the initial and final deployments of Lloyd Algorithm in WSN2. As usual, the final distortion only considers sensors in the backbone network. Initially, two strong sensors and four weak sensors are consisted in the backbone network shown in Fig.
Fig. 2. Comparison of distortion for different algorithms in WSN1.

Fig. 3. Comparison of coverage for different algorithms in WSN1.
In Fig. 5a, only one strong sensor and four weak sensors are included in the backbone network, resulting in a large distortion $D(P) = 12.67$ which is only 0.48 smaller than the initial distortion. Fig. 5c shows the outcome of RL Algorithm in WSN2. After 500 iterations, the distortion is decreased from 13.15 to 5.52. Simultaneously, the coverage area is increased from 0.08 to 1.46 and the final deployment is connected. Fig. 5d shows the final deployment of DA Algorithm in WSN2. After 500 iterations, the distortion is decreased from 13.15 to 1.10, which is better than that of RL Algorithm. Simultaneously, the coverage area is increased from 0.08 to 2.48 and full connectivity is provided.

Figs. 6a and 6b show one example of the initial and the final deployments of Lloyd Algorithm in WSN3. Initially, one strong sensor, two medium sensors and four weak sensors are consisted in the backbone network shown in Fig. 6a. In Fig. 6b, two strong sensors, one medium sensors and one weak sensors are disconnected from the backbone network, resulting in a large distortion $D(P) = 19.83$. Fig. 6c shows the outcome of RL Algorithm in WSN3. After 500 iterations, the distortion is decreased from 10.96 to 3.22. Simultaneously, the coverage area is increased from...
Fig. 5. Sensor deployments in WSN2. (a) The initial sensor deployment and the corresponding weighted Voronoi regions. (b) The final deployment of Lloyd Algorithm after 500 iterations. (c) The final deployment of RL Algorithm after 500 iterations. (d) The final deployment of DA Algorithm after 500 iterations. Strong sensors and weak sensors in the backbone network are, respectively, denoted by hollow circles and squares. Sensors out of the backbone network are denoted by dots. The corresponding centroid for strong sensors and weak sensors are, respectively, denoted by stars and crosses. The radius of each gray circle is $R_c/2 = 0.25$.

0.08 to 1.51 and the final deployment is connected. Fig. 6d shows the final deployment of DA Algorithm in WSN3. After 500 iterations, the distortion is decreased from 10.96 to 1.35, which is better than that of RL Algorithm. Simultaneously, the coverage area is increased from 0.03 to 2.07 and full connectivity is provided.

Figs. 7 and 8 illustrate the performance of the above algorithms for 10 random initial deployments in WSN2. Figs. 9 and 10 show similar performances in WSN3. The trends in heterogeneous WSNs 2 and 3 are similar to those in homogeneous WSN1.
VI. CONCLUSIONS

We studied the deployment of sensors in heterogeneous wireless sensor networks. Similar to homogeneous WSNs, the necessary condition for optimal deployment implies that every sensor node location should coincide with the centroid of its own optimal sensing region. Moreover, we considered a limited communication range for the sensor nodes and modeled the sensor deployment problem as a source coding problem with distortion reflecting sensing accuracy. By defining an appropriate distortion measure, we proposed a Restrained Lloyd algorithm and a Deterministic Annealing algorithm to optimize sensor deployment in both homogeneous and heterogeneous WSNs. Our simulation results show that both DA and RL algorithms outperform
Fig. 7. Comparison of distortion with different algorithms in WSN2.

Fig. 8. Comparison of coverage area with different algorithms in WSN2.
Fig. 9. Comparison of distortion with different algorithms in WSN3.

Fig. 10. Comparison of coverage area with different algorithms in WSN3.
the existing Lloyd algorithm when communication range is limited and provide a fully connected network. The DA is not sensitive to initial conditions.

**APPENDIX A**

**PROOF OF PROPOSITION 1**

*Proof:* Let $\text{Part}_{ij} = \{ q \mid \eta_i \| q - p_i \|^2 \leq \eta_j \| q - p_j \|^2 \}$ be the pairwise weighted Voronoi region of Sensor $i$ when we only consider Sensors $i$ and $j$. Then, the exact weighted Voronoi region of Sensor $i$ is the intersection of these pairwise weighted Voronoi regions, i.e., $V^H_i(P) = \bigcap_{j \neq i} \text{Part}_{ij}$. We define the coordinates of $q = (x, y)$, $p_i = (p_{ix}, p_{iy})$ and $p_j = (p_{jx}, p_{jy})$ and define $\eta = \sqrt{\frac{\eta_i}{\eta_j}} > 0$. Then, expanding the hyperplane equation $\eta_i \| q - p_i \|^2 = \eta_j \| q - p_j \|^2$ results in

$$(\eta^2 - 1)(x^2 + y^2) + 2(p_{jx} - \eta^2 p_{ix})x + 2(p_{jy} - \eta^2 p_{iy})y = p_{jx}^2 - \eta^2 p_{ix}^2 + p_{jy}^2 - \eta^2 p_{iy}^2$$

(22)

When $\eta = 1$, the hyperplane equation is

$$2(p_{jx} - p_{ix})x + 2(p_{jy} - p_{iy})y + p_{ix}^2 + p_{iy}^2 - p_{jx}^2 - p_{jy}^2 = 0$$

(23)

This hyperplane is the boundary of the half space $HS(p_i, p_j)$. When $\eta < 1$, $\text{Part}_{ij}$ is defined by

$$(q - c)^2 \geq r^2.$$  

(24)

When $\eta > 1$, $\text{Part}_{ij}$ is defined by

$$(q - c)^2 \leq r^2,$$  

(25)

where $c = (\frac{p_{ix} - \eta^2 p_{ix}}{1-\eta^2}, \frac{p_{iy} - \eta^2 p_{iy}}{1-\eta^2})$, $r = \frac{\eta}{1-\eta^2} \| p_i - p_j \|$. Therefore, we have

$$V^H_i(P) = \{ q \in Q \mid \eta_k \| q - p_k \| \leq \eta_t \| q - p_t \|^2 , \forall t \in 1, \ldots, n \}$$

$$= \left[ \bigcap_{i : \eta_i < \eta_k} B(c_{ik}, r_{ik}) \right] \bigcap \left[ \bigcap_{i : \eta_i = \eta_k} HS(p_k, p_i) \right] \bigcap \left[ \bigcap_{j : \eta_j > \eta_k} B(c_{kj}, r_{kj}) \right]^c$$

(26)
where $A^c$ denotes the complementary set of $A$.  

\section*{Appendix B}

\textbf{Proof of Proposition 2}

\textit{Proof:} Let $\tilde{V}_k^H(P)$ be the weighted Voronoi region of Sensor $k$ when we ignore sensors with larger cost parameters. Accordingly, $\tilde{V}_k^H(P)$ is defined by

\begin{equation}
\tilde{V}_k^H(P) = \{ q \in Q | \eta_k\|q - p_k\|^2 \leq \eta_i\|q - p_i\|^2, \text{ for any } i \text{ such that } \eta_k \geq \eta_i \} \tag{27}
\end{equation}

Review the definition of $V_k^H(P)$ in Eq. (6), we have

\begin{equation}
V_k^H(P) = \{ q \in Q | \eta_k\|q - p_k\|^2 \leq \eta_j\|q - p_j\|^2, \eta_k\|q - p_k\|^2 \leq \eta_j\|q - p_j\|^2, \text{ for any } i, j \text{ such that } \eta_j > \eta_k, \eta_k \geq \eta_i \} \tag{28}
\end{equation}

Obviously, $\tilde{V}_k(P)$ is the intersection of convex regions and therefore star-shaped. The relationship between $\tilde{V}_i^H(P)$ and $V_i^H(P)$ is

\begin{equation}
\tilde{V}_k^H(P) = V_k^H(P) \bigcup W_k, \tag{29}
\end{equation}

where $W_k = \{ q \in Q | \eta_k\|q - p_k\|^2 \geq \eta_j\|q - p_j\|^2, \eta_k\|q - p_k\|^2 \leq \eta_i\|q - p_i\|^2 \text{ for any } i, j \text{ such that } \eta_j > \eta_k, \eta_k \geq \eta_i \}$. For any point $q$ such that $|\eta_j\|q - p_j\|^2 \leq \eta_k\|q - p_k\|^2$ and $\eta_k\|q - p_k\|^2 \leq \eta_i\|q - p_i\|^2$, we have $\eta_j\|q - p_j\|^2 \leq \eta_i\|q - p_i\|^2$. So $W_k$ can be rewritten as

\begin{equation}
W_k = \{ q \in Q | \eta_j\|q - p_j\|^2 \leq \eta_i\|q - p_i\|^2, \eta_j\|q - p_j\|^2 \leq \eta_k\|q - p_k\|^2, \eta_k\|q - p_k\|^2 \leq \eta_i\|q - p_i\|^2 \text{ for any } i, j \text{ such that } \eta_j > \eta_k, \eta_k \geq \eta_i \}. \tag{30}
\end{equation}

Due to the definitions of $\tilde{V}_k^H(P)$ and $V_k^H(P)$, we have

\begin{equation}
W_k = \bigcup_{j: \eta_j > \eta_k} V_j^H(P) \cap \tilde{V}_k^H(P) = \bigcup_{j: \eta_j > \eta_k} \left[ V_j^H(P) \cap \tilde{V}_k^H(P) \right] \tag{31}
\end{equation}

Replacing $W_k$ from Eq. (31) in Eq. (29), we get the final result

\begin{equation}
\tilde{V}_k^H(P) = V_k^H(P) \bigcup_{j: \eta_j > \eta_k} \left[ V_j^H(P) \cap \tilde{V}_k^H(P) \right] \tag{32}
\end{equation}
The elements in the right side are disjoint subsets. Without loss of generality, let us assume that the \( m \) disjoint cost parameters are ordered such that the \( k \)-level cost parameter is larger than the \( k+1 \)-level cost parameter. We also call the set including the indices of all sensors with a \( k \)-level cost parameter \( Z_k \). Then:

1. For any level-1 sensor \( i \), \( V^H_i(P) \) is a convex set. Accordingly, the intersection \( V^H_i(P) \cap W \) is a convex set and therefore Eq. (14) holds due to Proposition A.1 in [1].

2. Assume the equation holds for any sensor whose level is smaller than or equal to \( k \). Consider a sensor \( i \) whose level is \( k+1 \), by using the relationship between \( \tilde{V}^H_i(P) \) and \( V^H_i(P) \) in Eq. (32), we rewrite the objective function as

\[
\int_{V^H_i(P) \cap W} \varphi(q,p_i) dq = \int_{\tilde{V}^H_i(P) \cap W} \varphi(q,p_i) dq - \sum_{l=1}^{k} \sum_{t \in Z_l} \left[ \int_{V^H_l(P) \cap \tilde{V}^H_i(P) \cap W} \varphi(q,p_i) dq \right].
\]  

(33)

Since \( V^H_i(P) \cap W \) is a convex set and therefore star-shaped, the partial derivative of the first term can be solved by proposition A.1 in [1]. Sensors’ levels in the second term are smaller than \( k \) and thus the partial derivative of the second term can be solved by our assumption in Step (2). Therefore, the partial derivative becomes

\[
\frac{\partial}{\partial p_j} \int_{V^H_i(P) \cap W} \varphi_i(q,p_i) dq = \int_{\tilde{V}^H_i(P) \cap W} \frac{\partial \varphi_i(q,p_i)}{\partial p_j} dq + \int_{\partial[\tilde{V}^H_i(P) \cap W]} \varphi_i(\gamma,p_i) n^i(\gamma) \frac{\partial \gamma}{\partial p_j} d\gamma
\]

\[
- \sum_{l=1}^{k} \sum_{t \in Z_l} \left[ \int_{V^H_l(P) \cap \tilde{V}^H_i(P) \cap W} \frac{\partial \varphi_i(q,p_i)}{\partial p_j} dq \right]
\]

\[
- \sum_{l=1}^{k} \sum_{t \in Z_l} \left[ \int_{\partial[V^H_l(P) \cap \tilde{V}^H_i(P) \cap W]} \varphi_i(\gamma,p_i) n^i(\gamma) \frac{\partial \gamma}{\partial p_j} d\gamma \right].
\]  

(34)
Note that $\overline{V}_i^H(P) \cap W = [V_i^H(P) \cap W] \cup G(P)$, where $G(P) = \bigcup_{l=1}^{k} \bigcup_{t \in Z_l} (V_t(P) \cap V_i^H(P) \cap W)$, is a star-shaped set consisting of several disjoint subsets. By using Lemma 3, we have

$$\int_{\partial[V_i^H(P) \cap W]} \varphi_i(\gamma, p_i) n^t(\gamma) \frac{\partial \gamma}{\partial p_j} d\gamma - \sum_{l=1}^{k} \sum_{t \in Z_l} \int_{\partial[V_t^H(P) \cap V_i^H(P) \cap W]} \varphi_i(\gamma, p_i) n^t(\gamma) \frac{\partial \gamma}{\partial p_j} d\gamma$$

$$= \int_{\partial[V_i^H(P) \cap W]} \varphi_i(\gamma, p_i) n^t(\gamma) \frac{\partial \gamma}{\partial p_j} d\gamma.$$

(35)

Also, we have

$$\int_{\overline{V}_i^H(P) \cap W} \frac{\partial \varphi_i(q, p_i)}{\partial p_j} dq - \sum_{l=1}^{k} \sum_{t \in Z_l} \int_{\overline{V}_t^H(P) \cap \overline{V}_i^H(P) \cap W} \frac{\partial \varphi_i(q, p_i)}{\partial p_j} dq = \int_{\overline{V}_i^H(P) \cap W} \frac{\partial \varphi_i(q, p_i)}{\partial p_j} dq.$$

(36)

Eq. (14) is derived by replacing Eqs. (35) and (36) in Eq. (34). In other words, Eq. (14) is correct for sensors whose level is smaller than or equal to $k + 1$. In summary, Eq. (14) is correct for sensors in all levels of heterogeneous WSNs.

ACKNOWLEDGMENT

The authors would like to thank Dr. Erdem Koyuncu for helpful discussions.

REFERENCES

[1] 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.

[2] X. Liu, “Coverage with Connectivity in Wireless Sensor Networks,” Broadband Communications, Networks and Systems, pp. 1-8, Oct. 2006.

[3] G. Wang, G. Cao, and T. F. La Porta, “Movement-assisted sensor deployment,” IEEE Trans. Mob. Comput., vol. 5, no. 6, pp. 640–652, June 2006.

[4] J. C. Kieffer, “Exponential rate of convergence for Lloyd’s method I,” IEEE Trans. Inf. Theory, vol. 28, no. 2, pp. 205–210, Mar. 1982.

[5] X. Wu, “On convergence of Lloyd’s method I,” IEEE Trans. Inf. Theory, vol. 38, no. 1, pp. 171–174, Jan. 1992.

[6] Q. Du, M. Emelianenko, and L. Ju, “Convergence of the Lloyd algorithm for computing centroidal Voronoi tessellations,” SIAM J. Numer. Anal., vol. 44, no. 1, pp. 102–119, Feb. 2006.

[7] C. Hu, X. Wang, Z. Yang, J. Zhang, Y. Xu, and X. Gao, “A geometry study on the capacity of wireless networks via percolation,” IEEE Trans. Commun., vol. 58, no. 10, pp. 2916–2925, Oct. 2010.

[8] H. Yousefi’zadeh, H. Jafarkhani, and J. Kazemitarb, “A study of connectivity in MIMO fading ad-hoc networks,” IEEE/KICS J. Commun. Networks (JCN), vol. 11, no. 1, pp. 47–56, Feb. 2009.
[9] 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.

[10] X. Bai, S. Kumar, D. Xuan, Z. Yun and T. H. Lai, “Deploying Wireless Sensors to Achieve Both Coverage and Connectivity,” MobiHoc, pp. 131-142, May 2006.

[11] B. Wang “Coverage Problems in Sensor Networks: A Survey,” ACM Computing Surveys, vol. 43, Article 32, Oct. 2011.

[12] R. M. Gray and D. L. Neuhoff, “Quantization,” IEEE Trans. Inf. Theory, vol. 44, no. 6, pp. 2325–2383, Oct. 1998.

[13] A. Gersho and R. M. Gray, Vector Quantization and Signal Compression. Boston, MA: Kluver, 1992.

[14] T. M. Cover and J. A. Thomas, Elements of Information Theory. John Wiley and Sons, Inc., New York, 1991.

[15] H. Mahboubi, K. Moezzi, A. G. Aghdam, K. Sayrafian-Pour, and V. Marbukh, “Self-deployment algorithms for coverage problem in a network of mobile sensors with unidentical sensing ranges,” IEEE Global Telecommun. Conf. (GLOBECOM), Dec. 2010.

[16] C. Wu and Y. Chung, “Heterogeneous Wireless Sensor Network Deployment and Topology Control Based on Irregular Sensor Model,” Advances in Grid and Pervasive Computing, vol. 4459 of the series Lecture Notes in Computer Science, pp. 78-88, 2007.

[17] 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.

[18] F. Aurenhammer, “Voronoi diagrams – a survey of fundamental geometric data structure,” ACM Computing Surveys, vol. 23, no. 3, pp. 345–405, Sept. 1991.

[19] F. Aurenhammer and H. Edelsbrunner, “An optimal algorithm for constructing the weighted Voronoi diagram in the plane,” Pattern Recognition, vol. 17, no. 2, pp. 251–257, 1984.

[20] S. Valeriu, “Introduction to the Axiomatic Theory of Convexity,” Russian, 1984.

[21] W. P. Soltan, “Abstract Convex Analysis,” Canadian Mathematical Society series of monographs and advanced texts, John Wiley and Sons, Inc., New York, 1997.

[22] P.J.M van Laarhoven and E.H.L Aarts. “Simulated Annealing: Theory and Applications,” D. Reidel Publishing Company, Dordrecht, Holland, 1987.

[23] S. Geman and D. Geman. “Stochastic Relaxation, Gibbs Distortion and the Bayesian Restoration of Images,” IEEE Trans. Pattern Anal. and Mach. Int., 11(6). pp. 689-691, 1984.