

# LAACAD: Load bAlancing $k$ -Area Coverage through Autonomous Deployment in Wireless Sensor Networks

Feng Li \*, Jun Luo \*, Shi-Qing Xin \*, Wen-Ping Wang †, and Ying He\*

\*School of Computer Engineering, Nanyang Technological University, Singapore

†Department of Computer Science, The University of Hong Kong, China

{fli3,junluo,sqxin,yhe}@ntu.edu.sg, wenping@cs.hku.hk

**Abstract**—Although the problem of  $k$ -area coverage has been intensively investigated for dense wireless sensor networks (WSNs), how to arrive at a  $k$ -coverage sensor deployment that optimizes certain objectives in relatively sparse WSNs still faces both theoretical and practical difficulties. In this paper, we present a practical algorithm LAACAD (Load bAlancing  $k$ -Area Coverage through Autonomous Deployment) to move sensor nodes toward  $k$ -area coverage, aiming at minimizing the maximum sensing range required by the nodes. LAACAD enables purely autonomous node deployment as it only entails localized computations. We prove the convergence of the algorithm, as well as the (local) optimality of the output. We also show that our optimization objective is closely related to other frequently considered objectives. Therefore, our practical algorithm design also contributes to the theoretical understanding of the  $k$ -area coverage problem. Finally, we use extensive simulation results both to confirm our theoretical claims and to demonstrate the efficacy of LAACAD.

**Keywords**-Wireless sensor networks, area coverage,  $k$ -coverage, autonomous deployment, load balancing

## I. INTRODUCTION

One of the major functions of *wireless sensor networks* (WSNs) is to monitor a certain area in terms of whatever physical quantities demanded by applications [1]. In achieving this goal, a basic requirement imposed onto WSNs is their *area coverage*:<sup>1</sup> it indicates the monitoring quality of WSNs. Whereas many research proposals focus on either analyzing the performance of static sensor deployments (e.g., [2], [3]) or scheduling sensor activity to retain the coverage of given deployments (e.g., [7], [8]), there exists an unfailing trend in seeking autonomous deployments assisted by mobile sensor nodes to arrive at certain predefined objectives (e.g., [9], [10]). Our proposal in this paper falls into this trend.

Due to the vulnerability of sensor nodes, multiple-coverage ( $k$ -coverage) is often applied to enhance the fault tolerance in the face of node failures (e.g., [3], [7]). In addition,  $k$ -coverage may yield higher sensing accuracy through data fusion [11] or aggregation [12], [13]. Existing approaches in achieving  $k$ -coverage rely on either randomized (e.g., [2], [14]) or regular (e.g., [3], [15]) deployments.

\*This work is supported in part by AcRF Tier2 ARC15/11.

<sup>1</sup>In this paper, we only focus on approaches concerning area coverage (e.g., [2], [3]), as opposed to the *point coverage* (e.g., [4]–[6]).

Whereas randomized deployments require a substantially denser network (e.g., [2], [14]), regular deployments only serve as theoretical guidelines [3], [15] as they often require centralized coordinations and may not accommodate irregular network regions. Therefore, autonomous deployments, if made possible by employing movable nodes [16], are good complements to the randomized or regular deployments.

However, existing techniques for autonomous deployments may only handle 1-coverage, and extending them to  $k$ -coverage is highly nontrivial. First of all, autonomous deployments through (node) motion control require each node to compute its coverage in a localized manner (i.e., relying as much as possible on close-by nodes). Although quite a few localized algorithms have been proposed to perform such computations for 1-coverage (e.g., [9], [17]), no algorithm, to the best of our knowledge, exists for localized  $k$ -coverage computations. Secondly, even if the  $k$ -coverage computations can be performed locally, there is no guarantee whether a motion control strategy may converge, due to the significant difference between 1-coverage and  $k$ -coverage. Finally, existing approaches are almost heuristics that offer no provable guarantee on the quality of the eventual deployment. These are exactly the problems we want to tackle in our paper.

In this paper, we consider the problem of moving sensor nodes towards  $k$ -coverage. In particular, we assume that nodes are randomly deployed initially and that each node has a tunable sensing range. Our goal is to cover a certain *monitored area* to the extent that every point in this area is at least monitored by  $k$  sensor nodes and that the maximum sensing range used by the nodes is minimized. As a larger sensing range implies a larger energy consumption of a node, our LAACAD (Load bAlancing  $k$ -Area Coverage through Autonomous Deployment) approach aims at balancing the sensing load (thus prolonging network lifetime) while guaranteeing  $k$ -coverage, with the help of mobile nodes. The main contributions we are making in this paper are:

- We design the LAACAD algorithm such that it executes in a localized manner, i.e., only relying on information from close-by nodes.
- We prove the convergence of LAACAD as well as the (local) optimality of its output.

- We discuss the relation between the output of LAA-CAD and other commonly used optimization objectives, which provides a better understanding of optimal  $k$ -coverage deployments whose theoretical characterizations are hard to obtain under general settings.

To the best of our knowledge, we are the first to tackle the problem of  $k$ -coverage autonomous deployment.

The remaining of our paper is organized as follows. We briefly survey the closely related literature in Sec. II. We formally define our model and problem in Sec. III, in which we also review the basic mathematical tools we need in our later algorithm design. In Sec. IV, we present our LAACAD algorithm details and analyze performance, we also discuss the interpretation of our solution with respect to other optimization frameworks. The efficacy of LAACAD is further confirmed by extensive simulation results reported in Sec. V. We finally conclude our paper in Sec. VI.

## II. RELATED WORK

Given the vast literature for coverage in WSNs, we can only focus on the topics related to area coverage and mobile assisted autonomous deployments, but have to leave out other interesting topics such as point (or target) coverage [4]–[6], area coverage with random deployments [2], [7], and topological approaches [18].

The static and deterministic area coverage problem is essentially a geometry problem; the results for 1-coverage with a minimum number of nodes can be directly taken from pure mathematical research [19]. In later research proposals for WSN coverage, the focus is more on minimum node 1-coverage with certain connectivity requirement (e.g., [3], [20], [21]). While it is known that a 1-covered WSN is also connected if the transmission range  $R_t$  and the sensing range  $R_s$  satisfy  $R_t \geq \sqrt{3}R_s$ , a strip-based node deployment is proposed in [20] for other values of  $R_t$ , and this deployment strategy is proven to be asymptotically optimal in [21]. In fact, adding more strips may allow a WSN to achieve higher degree of connectivity (or  $k$ -connectivity).

Compared with  $k$ -connectivity, the progress on  $k$ -coverage appears to be relatively slower. A few 3-coverage heuristics that aim at bounding the minimum separation among sensor nodes is proposed in [22]; the paper also shows that bounding the min-separation may lead to lower coverage redundancy and is hence a good approximation to minimum node 3-coverage. To the best of our knowledge, the only optimality result in terms of minimum node  $k$ -coverage is presented in [3], where  $k = 2$ . It appears that minimum node  $k$ -coverage (for  $k > 2$ ) is better to be tackled indirectly due to its hardness. As we will show in Sec. IV-C, our objective of a  $k$ -coverage with mini-max sensing range may also imply minimum node  $k$ -coverage. Deploying WSNs for  $k$ -coverage using mobile nodes is also reported in [15], [23], but their approaches are not autonomous (hence the algorithms are

not localized) as they all rely on a “blueprint” to guide the node mobility.

Our work is also related to the sensing heterogeneity issue [10], [24]. However, unlike the previous proposals that aim to cope with sensing heterogeneity or evaluate its impact, we actively exploit the sensing heterogeneity to construct our algorithm that guides the autonomous deployment.

## III. PRELIMINARIES AND PROBLEM DEFINITION

We present the system model and formally define our optimization problem in this section. We also briefly introduce certain mathematical basics that are relevant to our later algorithm design.

### A. System Model

We assume a WSN consisting of a set  $\mathcal{N} = \{n_1, \dots, n_N\}$  of sensor nodes, and  $|\mathcal{N}| = N$ . Let  $\mathcal{U} = \{u_1, \dots, u_N\}$  denote the locations of sensor nodes. The nodes are initially deployed arbitrarily on a 2D *targeted area*  $\mathcal{A}$ . Each node  $n_i$  is equipped with certain mechanisms (e.g., motors plus wheels) that allow it to gradually change its location  $u_i$  [16]. All nodes have an identical transmission range  $\gamma$ , and we denote by  $\mathcal{N}(n_i)$  the one-hop neighbors of node  $n_i$ , i.e., the nodes within the transmission range of  $n_i$ .

We define the omnidirectional sensing model as a disk centered at  $u_i$  with sensing range  $r_i$ . We assume the sensing ranges are adjustable according to different application requirements. A point  $v \in \mathcal{A}$  is said to be covered by node  $n_i$  iff the Euclidean distance between  $v$  and node location  $u_i$  is no longer than  $r_i$ , i.e.,  $\|v - u_i\|_2 \leq r_i$ . We use  $f(v, u_i, r_i)$  to indicate whether or not  $v$  is covered by node  $n_i$ .

$$f(v, u_i, r_i) = \begin{cases} 1 & \text{if } \|v - u_i\|_2 \leq r_i \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

In other words, if  $v$  is covered by  $n_i$ ,  $f = 1$ ; otherwise,  $f = 0$ . We rely on the ranging ability of each node to construct a local coordinate system for motion control, so location information is not essential to our algorithm.

In terms of energy cost, we only focus on the cost induced by the sensing activities of a node. Because our network deployment strategy aims at achieving a **constant** (and long-term) coverage by moving sensor nodes in the **initial** phase, the communication cost becomes negligible as the data transmission activities only take place sporadically, while the energy spent in moving is only a one-time investment. We assume that the energy consumed by a sensor node  $n_i$  is an increasing function  $E(r_i)$  of its sensing range  $r_i$ , and this function is identical for all nodes.

### B. Problem Formulation

The node locations and sensing ranges,  $\{u_i, r_i\}_{i=1, \dots, N}$ , define a network deployment with a certain coverage.

**Definition 1.** A network deployment  $\{u_i, r_i\}$  is said to achieve  $k$ -coverage iff for any point  $v \in \mathcal{A}$ , there exist at least  $k$  sensor nodes covering it, or  $\sum_i f(v, u_i, r_i) \geq k$ .

To allow sensor nodes properly cover the targeted area, we divide  $\mathcal{A}$  into several disjoint areas  $\{\mathcal{A}_j^k\}_{j=1,2,\dots}$ , and at least  $k$  sensor nodes are allocated to take charge of each area. In other words, each sensor node  $n_i$  takes care of multiple subareas, and we indicate this relation by  $n_i(\mathcal{A}_j^k)$ : it equals 1 if  $n_i$  covers  $\mathcal{A}_j^k$ ; otherwise 0. We also denote by  $\mathcal{A}_{n_i}^k$  the area covered by  $n_i$ : we have  $\mathcal{A}_{n_i}^k = \bigcup_{n_i(\mathcal{A}_j^k)=1} \mathcal{A}_j^k$ . Obviously, the sensing range  $r_i$  of  $n_i$  is determined by the farthest point in  $\mathcal{A}_{n_i}^k$  from  $u_i$ , i.e.,  $r_i = \max_{v \in \mathcal{A}_{n_i}^k} \|v - u_i\|_2$ , so that  $\mathcal{A}_{n_i}^k$  can be totally covered by  $n_i$ .

Our  $k$ -coverage sensor deployment problem ( $k$ -CSDP) can be formulated as follows.

$$\begin{aligned} & \underset{\{u_i\}, \{\mathcal{A}_j^k\}, \{n_i(\mathcal{A}_j^k)\}}{\text{minimize}} && R \\ & \text{subject to} && \sum_{i=1}^N f(v, u_i, r_i) \geq k, \quad \forall v \in \mathcal{A} \\ & && \|v - u_i\|_2 \leq R, \quad \forall i, v \in \mathcal{A}_{n_i}^k \\ & && \mathcal{A}_{j_1}^k \cap \mathcal{A}_{j_2}^k = \emptyset, \quad \bigcup_j \mathcal{A}_j^k = \mathcal{A} \end{aligned} \quad (2) \quad (3) \quad (4) \quad (5)$$

Literally,  $k$ -CSDP aims at choosing the node locations  $\{u_i\}$ , the area partition  $\{\mathcal{A}_j^k\}$ , and the covering relations  $\{n_i(\mathcal{A}_j^k)\}$ , such that the maximum sensing range among all nodes is minimized. As energy consumption is an increasing function of sensing range,  $k$ -CSDP is equivalently balancing the energy consumption over a whole WSN and hence maximizing the lifetime of the WSN. As the problem is generally not convex due to its non-convex feasible region, we have to be contented with **local minimum**.

### C. High Order Voronoi Diagram

In this section, we briefly introduce the ideas and theories on *high order Voronoi diagram* [25]. They are closely related to our autonomous deployment strategy.

In a  $k$ -order Voronoi diagram, the targeted area  $\mathcal{A}$  is segmented into  $\hat{N}^k$  disjoint areas  $\{\mathcal{V}_j^k\}_{j=1,\dots,\hat{N}^k}$ ,<sup>2</sup> each of which is associated with  $k$  closest generators (sensor nodes in our case), i.e., a subset  $\mathcal{N}_j^k \subseteq \mathcal{N}$  with  $|\mathcal{N}_j^k| = k$ . The  $k$ -order Voronoi cell  $\mathcal{V}_j^k$  is defined as

$$\mathcal{V}_j^k = \left\{ v \in \mathcal{A} \mid \begin{array}{l} \|v - u_i\|_2 \leq \|v - u_{i'}\|_2, \\ \forall n_i \in \mathcal{N}_j^k, \quad n_{i'} \in \mathcal{N} / \mathcal{N}_j^k \end{array} \right\} \quad (6)$$

The set  $\mathcal{N}_j^k$  is called the *generator set* of  $\mathcal{V}_j^k$ . It is straightforward to see that each sensor node  $n_i$  is associated with multiple Voronoi cells. Let  $\mathcal{V}_{n_i}^k$  denote the union of the Voronoi cells for which  $n_i$  serves as a generator; we term

<sup>2</sup>In 1-order Voronoi diagram, the number of Voronoi cells equals the number of generators (i.e.,  $\hat{N}^1 = N$ ), while in generalized  $k$ -order Voronoi diagram ( $k \geq 1$ ),  $\hat{N}^k$  is  $\mathcal{O}(k(N - k))$  [25].

$\mathcal{V}_{n_i}^k$  the *dominating region* of  $n_i$  (hence  $n_i$  the *dominator* of  $\mathcal{V}_{n_i}^k$ ). We also have the following proposition.

**Proposition 1.** A point  $v \in \mathcal{A}$  is said to belong to  $\mathcal{V}_{n_i}^k$  iff there exist **at most**  $k - 1$  other generators such that their distance to  $v$  is less than  $\|v - u_i\|_2$ .

*Proof:* Assume  $v \in \mathcal{V}_{n_i}^k$  but there were another  $k$  nodes  $\{n_j\}_{j \in J, i \notin J, |J|=k}$  such that  $\|u_j - v\|_2 < \|u_i - v\|_2$ . Then the point  $v$  would **strictly** belong to the set of  $k$ -order Voronoi cells generated by  $\{n_j\}$ , which does not include  $n_i$  as a generator; a contradiction. Conversely, if there are at most  $k - 1$  nodes,  $\{n_j\}_{j \in J, i \notin J, |J| \leq k-1}$ , that are closer to  $v$  than  $n_i$ , we can find the set of  $v$ 's  $k$ -nearest nodes (which obviously contains  $\{n_j\} \cup \{n_i\}$ ) to generate a set of  $k$ -order Voronoi cells containing  $v$ . As  $n_i$  is a generator,  $v \in \mathcal{V}_{n_i}^k$ . ■

Based on the above proposition, assuming

$$\mathcal{S}_{n_i}^k(v) = \{n_j \in \mathcal{N} \mid \|u_j - v\|_2 < \|u_i - v\|_2, j \neq i\}$$

we can re-define the dominating region of  $n_i$  as

$$\mathcal{V}_{n_i}^k = \{v \in \mathcal{A} \mid |\mathcal{S}_{n_i}^k(v)| \leq k - 1\} \quad (7)$$

We illustrate  $k$ -order Voronoi partition ( $k = 1, 2, 3, 4$ ) generated by 30 nodes in Figure 1. The cells shown in

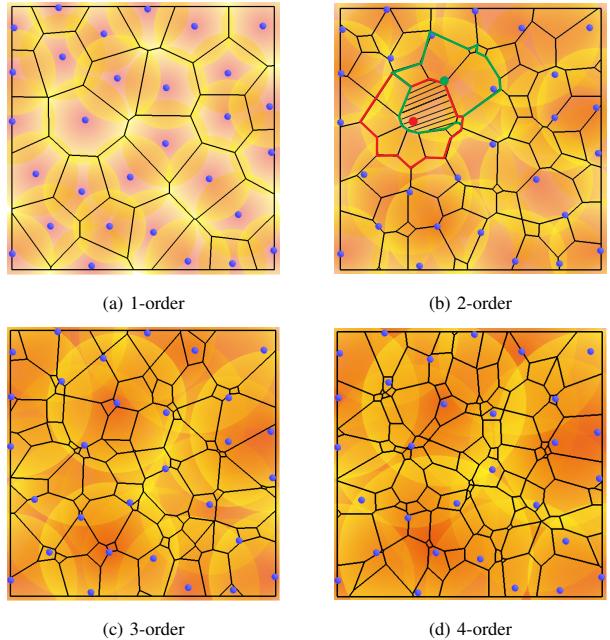


Figure 1.  $k$ -order Voronoi partition for  $k = 1, 2, 3, 4$ . The disks at the backdrop of each figure represent the (overlapping) sensing ranges of individual sensor nodes.

each figure are  $\{\mathcal{V}_j^k\}$ . Taking 2-order Voronoi partition for example, as shown in Figure 1(b), the area enclosed by red (resp. green) polygon is actually the dominating region of the red node (resp. green node). The hatched area is the Voronoi cell generated by the two nodes, i.e., the points in this area are closer to the two nodes than any other nodes.

#### IV. LAACAD: LOCALIZED $k$ -COVERAGE NODE DEPLOYMENT ALGORITHM

In this section, we first develop two optimality conditions for  $k$ -CSDP. Then we present the LAACAD algorithm details. The correctness of LAACAD is then proven, and we finally discuss the relation between  $k$ -CSDP and other optimization problems related to  $k$ -coverage deployment, along with the corresponding properties of LAACAD.

##### A. Optimal Conditions

To motivate our algorithm design, we first develop two optimality conditions for  $k$ -CSDP. Firstly, we show that, if we fix  $\{u_i\}$ , then  $k$ -order Voronoi diagram is the optimal solution to  $k$ -CSDP.

**Proposition 2.** *If we fix the sensor locations  $\{u_i\}_{i=1,\dots,N}$ , the  $k$ -order Voronoi diagram  $\{\mathcal{V}_1^k, \dots, \mathcal{V}_{N^k}^k\}$  generated by  $\{u_i\}_{i=1,\dots,N}$  is an optimal partition of  $\mathcal{A}$ . Also,  $n_i(\mathcal{V}_j^k) = 1$  if  $\mathcal{V}_j^k \subseteq \mathcal{V}_{n_i}^k$ ; otherwise  $n_i(\mathcal{V}_j^k) = 0$ .*

*Proof:* We prove the proposition by contradiction. Suppose for fixed  $\{u_i\}_{i=1,\dots,N}$ , there exists an optimal solution to  $k$ -CSDP denoted by  $R^*, \{\bar{\mathcal{A}}_j^k\}, \{n_i(\bar{\mathcal{A}}_j^k)\}$ . Let  $r_i^* = \max_{v \in \bar{\mathcal{A}}_{n_i}^k} \|v - u_i\|_2$  and  $r_i^V = \max_{v \in \mathcal{V}_{n_i}^k} \|v - u_i\|_2$ . Also assume that the optimal value is obtained for  $n_i^*$ , i.e.,  $R^* = r_i^*$ . If  $r_i^V = r_i^*$ , then it is straightforward to see that  $r_i^V = \max\{r_i^V\}$ , otherwise a contradiction to the definition of  $\mathcal{V}_{n_i}^k$ : some regions are not covered by the  $k$ -closest nodes. Therefore, in this case the  $k$ -order Voronoi diagram is equally optimal. If  $r_i^V > r_i^*$ , then it means that  $n_i^*$  could cede a certain region to have it covered by other nodes while reducing  $\max\{r_i^V\}$ . However, this again contradicts the definition of  $\mathcal{V}_{n_i}^k$ : as  $n_i^*$  is already one of the  $k$ -closest nodes that can cover the ceded region, ceding this region to some other nodes would simply increase  $\max\{r_i^V\}$ . ■

Before stating the second optimality condition, we need the following definition.

**Definition 2.** *Given an arbitrary set  $\mathcal{S}$  in Euclidean space, the Chebyshev center  $u_c$  is defined as:*

$$u_c = \arg \min_{\hat{u}} \left( \max_{u \in \mathcal{S}} \|u - \hat{u}\|_2 \right)$$

Using this definition, for an arbitrary partition  $\{\mathcal{A}_j^k\}$  and its dominator allocation  $\{n_i(\mathcal{A}_j^k)\}$ , the optimal locations of  $\{n_i\}$  can be obtained.

**Proposition 3.** *If we fix the partition  $\{\mathcal{A}_j^k\}$  and its dominator allocation  $\{n_i(\mathcal{A}_j^k)\}$ , the optimal sensor location  $u_i^*$  for  $k$ -CSDP is given by the Chebyshev center of  $\mathcal{A}_{n_i}^k$ .*

*Proof:* As  $n_i$  needs to cover  $\mathcal{A}_{n_i}^k$  and the objective of  $k$ -CSDP is to minimize the maximum sensing range among all sensors, the optimal solution under a fixed partition is achieved if each sensor individually minimizes its own

sensing range. This exactly coincides with the property of Chebyshev center, hence the proposition follows. ■

##### B. The Algorithm

Given the two optimality conditions stated in Sec. IV-A, we immediately have an iterative algorithm to solve  $k$ -CSDP. The algorithm proceeds in rounds. At the beginning of each round, the  $k$ -order Voronoi diagram is computed for the whole WSN, resulting in  $\{\mathcal{V}_1^k, \dots, \mathcal{V}_{N^k}^k\}$  along with  $\{\mathcal{V}_{n_1}^k, \dots, \mathcal{V}_{n_N}^k\}$ . Then each node computes the Chebyshev center of its dominating region, and moves to that location to end this round. The algorithm terminates if each node is indeed located at the Chebyshev center of its dominating region. As a perfect matching is impossible in the face of numerical errors, we use a small value  $\varepsilon$  as the stopping tolerance: the algorithm terminates if the distance from the node's current location to the Chebyshev center of its dominating region is smaller than  $\varepsilon$ . Also, in order to avoid oscillation, a step size  $\alpha < 1$  is chosen to confine the motion of the nodes. At the termination, each node tunes its sensing range to be the minimum value (the circumradius of its dominating region) that covers its dominating region. The pseudo-codes of our LAACAD algorithm are presented in **Algorithm 1**. As a dominating region is always a polygon,

---

##### Algorithm 1: LAACAD

---

**Input:** For each  $n_i \in \mathcal{N}$ , initial location  $u_i^{(0)}$ , stopping tolerance  $\varepsilon$   
**Output:**  $\{u_i^*\}$  and  $\{r_i^*\}$   
1 For every node  $n_i \in \mathcal{N}$  periodically (every  $\tau$  ms):  
2 Compute its dominating region  $\mathcal{V}_{n_i}^k$   
3 Compute the Chebyshev center  $c_i$  of  $\mathcal{V}_{n_i}^k$   
4 **if**  $\|u_i - c_i\|_2 > \varepsilon$  **then**  
5    $| u_i^+ \leftarrow u_i + \alpha(c_i - u_i)$            /\* $\alpha$  is the step size\*/  
6 **end**  
7  $u_i^* \leftarrow c_i, r_i^* \leftarrow \max_{u \in \mathcal{V}_{n_i}^k} \|u - u_i\|_2$

---

we apply Welzl's algorithm [26] to compute the Chebyshev center by taking the vertices of the region as the input.

Similar algorithms have been applied in [9], [10]. However, they were used to indirectly optimize a different objective (we refer to Sec. IV-C for a detailed discussion). Consequently, their approaches do not abide by the optimality conditions and employ a very different termination condition. Therefore, the convergence proofs given in [9], [10] do not apply to our case even for  $k = 1$ . Most importantly, as our algorithm deals with a more general  $k$ -coverage, we are facing the following new challenges.

- C1 How to compute  $\mathcal{V}_{n_i}^k$  in a localized manner without involving all nodes in a WSN?
  - C2 Does the algorithm converge for  $k \geq 1$ ? If yes, what is the relation between convergence and the step size  $\alpha$ ?
- In the following, we tackle these two challenges one by one.

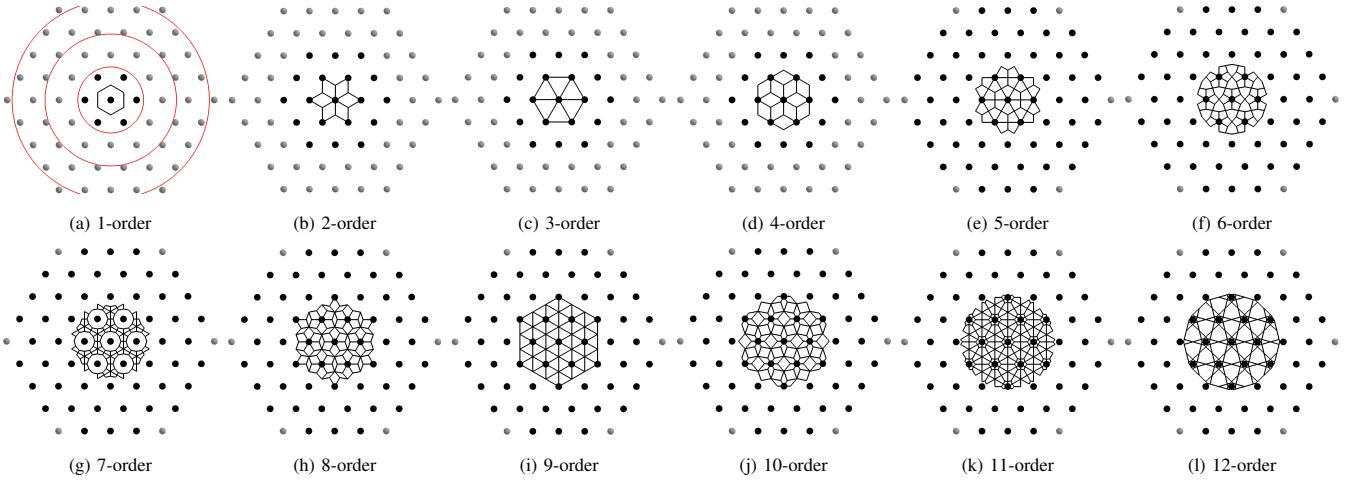


Figure 2. The dominating region of the central node in  $k$ -order Voronoi diagram for  $k$  ranging from 1 to 12. The central node needs to collect location (or range) information from its neighboring nodes (the dark nodes) via multi-hop communication according to **Algorithm 2**. Additionally, we illustrate multi-hop transmission range using red circles in (a). While the cases for  $k = 1$  can be handled by involving only the 6 closest nodes (1-hop neighbors) to the central node, computing the 2-, 3- and 4-order dominating regions requires 2-hop neighbors. When  $k > 4$ , all sensor nodes within 3 hops are involved.

1) *Localized Algorithm for Computing  $\mathcal{V}_{n_i}^k$* : Unlike 1-order Voronoi diagram that can be computed (mostly) by only interacting with one-hop neighbors  $\mathcal{N}(n_i)$  of a given node  $n_i$ ,  $\mathcal{N}(n_i)$  may not be sufficient to obtain  $k$ -order Voronoi cells, especially when  $k$  is large. The reason is simple: at least  $k+1$  nodes should be involved to compute a dominating region of  $n_i$  [27]. Therefore, our localized algorithm for computing  $\mathcal{V}_{n_i}^k$  works in an expending ring manner, as shown in **Algorithm 2**.

---

**Algorithm 2:** Localized  $\mathcal{V}_{n_i}^k$  Computation

---

**Input:** For each  $n_i \in \mathcal{N}$ , initial ring radius  $\rho = 0$   
**Output:**  $\mathcal{V}_{n_i}^k$

- 1 **repeat**
- 2      $\rho \leftarrow \rho + \gamma$ ;    $out \leftarrow \text{true}$
- 3      $\mathcal{N}(n_i, \rho) \leftarrow \{n_j \mid \|u_j - u_i\|_2 < \rho\}$
- 4     Construct a local coordinate system using  $\mathcal{N}(n_i, \rho)$
- 5     **foreach**  $v \in \mathcal{A}$  s.t.  $\|v - n_i\|_2 = \rho/2$  **do**
- 6          $\hat{\mathcal{S}}_{n_i}^k(v) \leftarrow \{n_j \in \mathcal{N}(n_i, \rho) \mid \|u_j - v\|_2 < \|u_i - v\|_2, j \neq i\}$
- 7         **if**  $|\hat{\mathcal{S}}_{n_i}^k(v)| < k$  **then**  $out \leftarrow \text{false}$ ; **break**
- 8     **end**
- 9 **until**  $out = \text{true}$ ;
- 10 Compute  $\mathcal{V}_{n_i}^k$  based on  $\mathcal{N}(n_i, \rho)$

---

Basically, we expand the search ring  $\rho$  with a granularity of the transmission range  $\gamma$  (line 2). As expending  $\rho$  beyond  $\gamma$  will need multi-hop communication and the hop number is always an integer, it makes no sense to apply a smaller granularity. We use the embedding algorithm proposed in [28] to construct a local coordinate system (line 4). If the location information is available, this step is not necessary.

Under the constructed coordinate system, we check whether the circle centered at  $u_i$  with a radius  $\rho/2$  is not dominated by  $n_i$  anymore (lines 5 to 8, based on **Proposition 1**). The ring expending terminates if the answer becomes true. Finally, we compute  $\mathcal{V}_{n_i}^k$  using only nodes falling into the current search ring (line 10).

In order to show that  $\mathcal{V}_{n_i}^k$  computed by **Algorithm 2** is indeed the one that would be computed in a centralized manner using global information, we need the follow lemma.

**Lemma 1.** *If the dominating region of  $n_i$  is enclosed by a circle centered at  $u_i$  with a radius of  $\rho/2$ , then it is fully determined by all the nodes located within another circle centered at  $u_i$  with a radius of  $\rho$ .*

*Proof:* For a disk  $\odot(u_i, \rho/2)$  centered at  $u_i$  with radius  $\rho/2$ , if  $\mathcal{V}_{n_i}^k \subset \odot(u_i, \rho/2)$ , the boundary of  $\mathcal{V}_{n_i}^k$  also belongs to  $\odot(u_i, \rho/2)$ . According to the definition of Voronoi cells, the cell boundary consists of bisectors, each of which is determined by two generators. For  $\mathcal{V}_{n_i}^k$ , one generator is  $n_i$ , and all other generators can be obtained by going through each segment (or bisectors) on the boundary of  $\mathcal{V}_{n_i}^k$  and identifying another generator that determines this bisector along with  $n_i$ . Since the boundary of  $\mathcal{V}_{n_i}^k$  belongs to  $\odot(u_i, \rho/2)$ , all generators of  $\mathcal{V}_{n_i}^k$  belong to  $\odot(u_i, \rho)$ . ■

The correctness of our algorithm is immediate from this lemma: as the algorithm terminates when  $n_i$  is not dominating the circle centered at  $u_i$  with a radius  $\rho/2$  anymore, the nodes falling into  $\odot(u_i, \rho)$  are sufficient to compute  $\mathcal{V}_{n_i}^k$ . In Figure 2, we demonstrate this sufficiency using  $k$ -order dominating region ( $k = 1$  to 12) in a regularly deployed WSN. The regular deployment is chosen to facilitate exposition, our algorithm works for any arbitrary deployments.

For a node  $n_i$  on the boundary, the search ring will never stop expanding, as the arc that is out of the network coverage will always need the domination of  $n_i$ . To cope with this issue, we first employ a boundary detection service (e.g., [29]) to indicate which nodes are on or close to the boundary. Based on the boundary awareness, the boundary node executes the circle checking procedure (lines 5 to 8) only for the arc that lies within the network coverage area, as shown in Figure 3. Finally, the boundary node calculates its dominating region, using  $\mathcal{N}(n_i, \rho)$  as well as the searching ring to determine the boundary of the dominating region. For

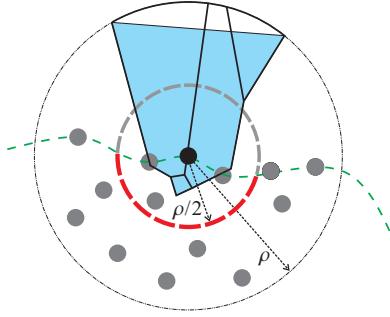


Figure 3. Recognizing the network boundary (the green dashed curve), a boundary node (dark) determines a searching ring (the outer circle) and check the half-radius arc within the network coverage area (the inner red arc). It then calculates its dominating region (the blue area) with  $\mathcal{N}(n_i, \rho)$ , while the searching ring helps to determine part of the boundary.

an initial random deployment in which nodes only occupy a small fraction of  $\mathcal{A}$ , this procedure has the effect of “pushing” boundary nodes outwards, hence expanding the network coverage to the whole  $\mathcal{A}$ . In fact, such a constrained checking procedure should always be used by nodes on the boundary of  $\mathcal{A}$ , the difference is that  $\mathcal{A}$ ’s boundary serves as a natural boundary for a dominating region.

2) *Convergence Analysis*: Showing the convergence of **Algorithm 1** appears to be highly non-trivial, as many  $k$ -order Voronoi cells are concerning a certain node, and the dominating region of a node is mostly probably a non-convex region. Fortunately, the convergence can be shown by focusing on the boundary of a dominating region.

**Proposition 4.** *Algorithm 1* is convergent.

*Proof:* As shown in Figure 4,  $u_i^l$  and  $\mathcal{V}_{n_i}^{k,l}$  are the location and dominating region of node  $n_i$  at the beginning of the  $l$ -th iteration, respectively. We also denote by  $c_i^l$  and  $R_i^l$  the Chebyshev center and the circumradius, respectively, of  $\mathcal{V}_{n_i}^{k,l}$  computed by  $n_i$  during the  $l$ -iteration. Let  $\hat{R}_i^l = \max_{u \in \mathcal{V}_{n_i}^{k,l}} \|u - u_i^l\|_2$  be the farthest distance from  $u_i^l$  in  $\mathcal{V}_{n_i}^{k,l}$ . Finally, we define  $R^l = \max_i \{R_i^l\}$  and  $\hat{R}^l = \max_i \{\hat{R}_i^l\}$ .

We first prove the convergence for  $\alpha = 1$  by contradiction. For each  $n_i$ , we put a disk  $\odot(c_i^l, R^l)$  centered at  $c_i^l$  with radius  $R^l$ . Obviously,  $\bigcup_{i=1}^N \odot(c_i^l, R^l)$  form a  $k$ -coverage over the targeted area  $\mathcal{A}$ . The convergence would be naturally

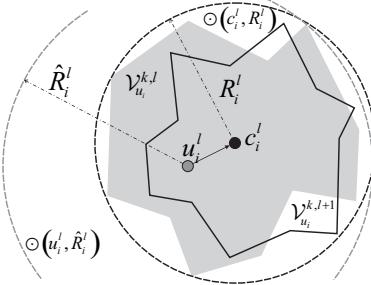


Figure 4. Notations in the proof for Proposition 4.

justified if we can prove that  $\mathcal{V}_{n_i}^{k,l+1}$  is inside  $\odot(c_i^l, R^l)$  after  $u_i^l$  is updated to  $c_i^l$  (i.e.,  $u_i^{l+1} = c_i^l$ ). For each point  $q$  on the boundary of  $\mathcal{V}_{n_i}^{k,l+1}$ , it is straightforward to see that  $c_i^l$  is the location of the  $k$ -th nearest nodes. Assume that  $q$  is outside  $\odot(c_i^l, R^l)$ , there would be only  $k-1$  disks covering  $q$ , which contradicts the fact that  $\bigcup_{i=1}^N \odot(c_i^l, R^l)$  constitute a  $k$ -coverage over  $\mathcal{A}$ . This completes the proof for  $\alpha = 1$ .

We then prove the convergence for  $0 < \alpha < 1$ . According to line 5 of **Algorithm 1**, during the  $l$ -th iteration,  $u_i^{l+1} = u_i^l + \alpha(c_i^l - u_i^l)$ . We put disks  $\odot(u_i^l, \hat{R}^l)$  and  $\odot(c_i^l, \hat{R}^l)$  centered at  $u_i^l$  and  $c_i^l$ , respectively. Obviously,  $\mathcal{V}_{n_i}^{k,l}$  is inside  $\odot(u_i^l, \hat{R}^l) \cap \odot(c_i^l, \hat{R}^l)$ , which implies  $\mathcal{V}_{n_i}^{k,l} \subset \odot(u_i^{l+1}, \hat{R}^l)$ . Hence,  $\bigcup_{i=1}^N \odot(u_i^{l+1}, \hat{R}^l)$  constitute a  $k$ -coverage over  $\mathcal{A}$ . Following a similar argument as for  $\alpha = 1$ , we have  $\mathcal{V}_{n_i}^{k,l+1}$  is involved in  $\odot(u_i^{l+1}, \hat{R}^l)$ , which completes the proof. ■

In summary, our LAACAD algorithm converges for any  $\alpha \in (0, 1]$ . Usually, smaller  $\alpha$  leads to slower convergence but smoother motion trace. As a byproduct of the proof, we also conclude that  $\hat{R}$  is non-increasing iteratively and finally equivalent to  $R$ . Especially, when  $\alpha = 1$ ,  $R$  is also non-increasing in the iterative process. Therefore,

**Corollary 1.** *Algorithm 1* converges to a local minimum of  $k$ -CSDP.

It is important to note that  $R^l$  and  $\hat{R}^l$  are introduced only for our proof. During the algorithm execution, each node  $n_i$  can only compute its own  $R_i^l$  and  $\hat{R}_i^l$ . According to our earlier discussion in Sec. IV-B1 (see Figure 3 also), the evolution of the node positions often takes two phases: an *expanding* phase and a *converging* phase. The expanding phase exists if the initial node distribution is non-uniform, our LAACAD algorithm will force the node to spread out during this phase, as discussed at the end of Sec. IV-B1. During this phase, both  $R^l$  and  $\hat{R}^l$  are most probably achieved by boundary nodes. The expanding phase ends when all the boundary nodes are at the boundary of  $\mathcal{A}$ , this is when the converging phase starts.

### C. Discussions

In this section, we show the relations between the output of our LAACAD algorithm and other optimization objectives

often considered for area coverage problems in WSNs, as well as some desirable properties of LAACAD.

*Min-Node  $k$ -Coverage:* One type of problem that is commonly tackled in the research community is to achieve  $k$ -coverage using a minimum number of nodes (e.g., [3], [15]). As this problem often assumes that all nodes have a fixed and identical sensing range  $r_s$ , it appears that LAACAD may not suggest a direct solution to it. However, we can transform our algorithm to deliver a good approximation to this min-node  $k$ -coverage problem as follows. LAACAD is called iteratively<sup>3</sup> and  $R^*$  is compared with  $r_s$  at the end of each iteration. Nodes are added (resp. reduced) if  $R^* > r_s$  (resp.  $R^* < r_s$ ), until  $R^* \leq r_s$  but adding one more node would make  $R^* > r_s$ . Although the solution may not be optimal, it yields very good approximation to the optimal solution, as we will demonstrate in Sec. V. In fact, as the up-to-date algorithms are all approximations for  $k > 2$  and they are not autonomous (e.g., [15]), our algorithm is also the first autonomous deployment for approximate min-node  $k$ -coverage with an arbitrary  $k$ .

*Maximum  $k$ -Coverage:* Another type of problem aims at maximizing coverage under fixed sensing ranges, but existing proposals only focus on 1-coverage [9], [10]. A natural definition of the general maximum  $k$  coverage problem is to maximize the area that is  $k$ -covered under a fixed sensing range. The major difference between  $k = 1$  and  $k > 1$  is that the former achieves maximum coverage if nodes are far apart from each other whereas the same principle does not apply to the latter. An obvious example is the following: assume only 3 nodes are used to 3-cover an area, the maximum coverage is achieved only if all three nodes are put at the same location. Consequently, the heuristic of bounding the minimum separation among nodes [22] does not work. Intuitively, the output of LAACAD may also serve as a good approximation to the maximum  $k$ -coverage problem, as, for example, LAACAD does converge to the optimal solution for the aforementioned 3-node case.

*Connectivity:* Although maintaining network connectivity is not our concern in designing LAACAD, it appears to be a natural outcome of achieving  $k$ -coverage for  $k \geq 2$ . Under  $k$ -coverage, it is easy to see that there should be at least  $k$  nodes in the sensing range  $r_i$  of a node  $n_i$  (including  $n_i$ ), otherwise  $u_i$  is not  $k$ -covered. In reality, as shown in Figure 2, there are at least 7 nodes in a certain sensing range for  $k \geq 2$ . Because a common (realistic) assumption in the literature is  $\gamma \geq r_i$  (e.g., [5], [9], [10]), each node in a WSN has at least a degree of 6, which is sufficient to guarantee connectivity in the WSN.

*Min-Max Fair:* While our  $k$ -CSCP only requires that the maximum sensing range is minimized and hence does not concern nodes with non-maximum sensing ranges, the

min-max fair utility (a Pareto optimal point) requires that a node  $n_i$  cannot further reduce its sensing ranges  $r_i$  without increasing the sensing range  $r_j$  ( $r_j \geq r_i$ ) of another node  $n_j$ . According to the property of  $k$ -order Voronoi diagram, the output of LAACAD is at least locally optimal with respect to the min-max fair utility, i.e., if we reduce  $r_i$ , another node  $n_j$  that shares dominating region boundary with  $n_i$  should increase  $r_j$  to maintain  $k$ -coverage, but we know  $r_j \geq r_i$  before  $r_i$  gets reduced. In fact, our simulation results in Sec. V show that, after LAACAD converges, the maximum and minimum sensing ranges are almost the same for  $k > 2$ .

## V. SIMULATIONS

In this section, we report our simulation results. We first present the convergence of LAACAD. Then, we analyze the sensing energy consumption. Finally, we evaluate the output of LAACAD in terms of minimizing the number of sensor nodes to achieve certain coverage, followed by the adaptability to network irregularities.

### A. Convergence

As convergence results we obtain from our extensive experiments are all similar, we only present one case to demonstrate the convergence of our algorithm. We consider a targeted area of  $1 \text{ km}^2$ , and initially deploy 100 sensor nodes at the bottom-left corner, as shown by Figure 5(a). In the following four sub-figures, it is obvious that our algorithm leads to an “even” node distribution in the sense of multiple coverage. Specifically, in the multiple coverage cases with  $k = 2, 3, 4$ , nodes tend to cluster in groups of size  $k$ , in contrast to the pure even distribution for  $k = 1$ . This is not a surprise as such an “even clustering” distribution yields more overlaps of the dominating regions among every cluster, which in turn reduces the required sensing range. Interestingly, this appears to also meet the needs of maximum  $k$ -coverage. As we discussed in Sec. IV-C, LAACAD leads to a co-location deployment for the extreme example of using three nodes to achieve 3-coverage.

We also show the convergence process of LAACAD in Figure 6. Since a sensor node finally reaches the Chebyshev center of its dominating region and the sensing range is equivalent to the circumradius of the dominating region, we illustrate the relationship between execution rounds (of length  $\tau$  ms each) and the maximum/minimum circumradii. As the nodes are deployed at the corner of the targeted area initially, the maximum circumcircle usually appears on the network boundary, which is mainly determined by the searching ring (Figure 3). Consequently, the maximum circumradii for  $k = 1, 2, 3, 4$  are almost the same at the beginning. Corresponding to our proof of the convergence, the maximum circumradius is monotonically decreasing with the execution rounds of LAACAD, while the minimum radius is increasing in general. In the end, the maximum and minimum radii are very close to each other, especially

<sup>3</sup>If an application only requires a one-time (rather than autonomous) deployment, we may use LACAAD in a centralized fashion.

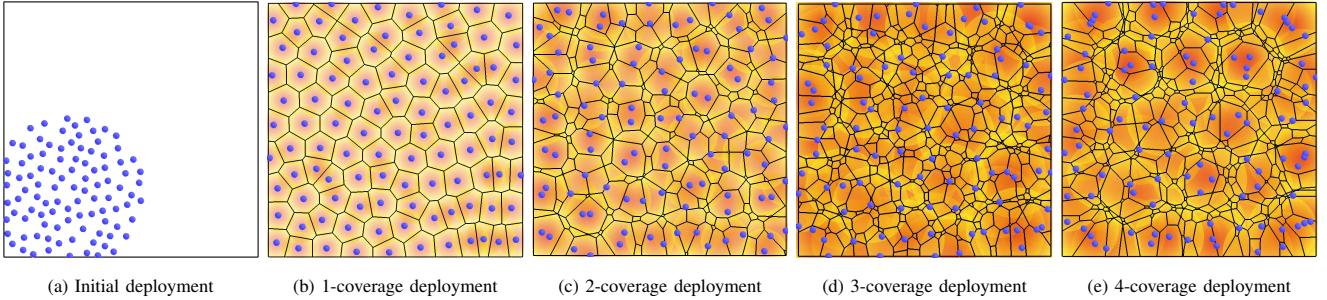


Figure 5. Initial deployment and  $k$ -coverage ( $k = 1, 2, 3, 4$ ) deployments as the output of LAACAD.

for larger  $k$ . While the monotonic decreasing in maximum circumradius shows the convergence of LAACAD, the fact that the minimum circumradius coincides with the maximum one further reflects the balance of sensing load in the WSN.

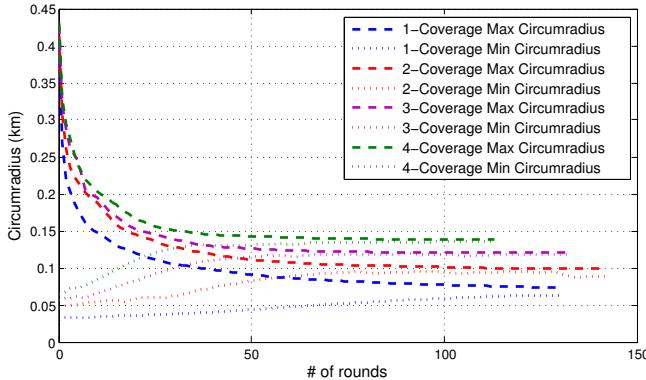


Figure 6. The convergence of LAACAD.

### B. Energy Consumption

In this section, we show the sensing energy consumption. We again consider a targeted area of  $1 \text{ km}^2$ , while scaling the network size (i.e., the number of sensor nodes) from 20 to 180. As the sensing range is modeled as a disk centered at  $u_i$  with radius  $r_i$ , we naturally define the energy consumption function as  $E(r_i) = \pi r_i^2$ : an increasing function of  $r_i$ .

We illustrate the sensing energy consumption in terms of maximum load  $\max\{E(r_i)\}_{i=1,\dots,N}$  and total load  $\sum_{i=1}^N E(r_i)$  in Figure 7. As we deploy more sensor nodes to a given targeted area, each node takes care of less area when achieving a certain coverage. The maximum sensing load is thus decreasing with the increasing number of nodes. Given a certain number of nodes, to achieve higher coverage degree, each sensor node is supposed to cover larger area thereby enhancing the maximum sensing load. We also observe that for  $k_1$ -coverage and  $k_2$ -coverage, the ratio of maximum loads between them is roughly  $k_1/k_2$ , which can be explained as follows. Since LAACAD makes the minimum sensing range very close to the maximum one,

each sensor node roughly covers the same area  $k|\mathcal{A}|/N$ , i.e.,  $E(r_i) = k|\mathcal{A}|/N$  where  $|\mathcal{A}|$  is the area of the targeted region. Nevertheless, increasing the number of nodes does decrease the total sensing load of the WSN, shown by Figure 7(b). Because using a less number of nodes leads to each sensor node taking charge of a larger area, which implies a larger sensing disk. This in turn yields more overlap between sensing ranges (i.e., a larger sensing redundancy), thus a higher total load.

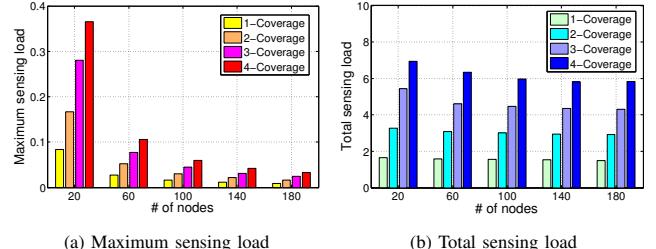


Figure 7. Energy consumption in the final deployments using 100 nodes

### C. Comparisons with Min-Node $k$ -Coverage

As mentioned in Sec. IV-C, our LAACAD algorithm results in a good approximation to min-node  $k$ -coverage problem (where all nodes have the same sensing range and the objective is to minimize the number of nodes used to achieve  $k$ -coverage). In this section, we compare our algorithm with the deployment strategies proposed in [3] and [15], in terms of the required number of nodes guaranteeing  $k$ -coverage ( $k \geq 2$ ). As we can anyway increase the minimum sensing range to the maximum one in the output of LAACAD without compromising the coverage (these two values are almost the same according to Sec. V-A), we assign an identical sensing range to every node as the maximum sensing range  $R^*$  in our case.

In [3], without considering boundary effect, Bai *et al.* have proven that the optimal **congruent** deployment density<sup>4</sup> for 2-coverage is  $4\pi/3\sqrt{3}$ , where sensor nodes have the same

<sup>4</sup>Deployment density is defined as a ratio of the area of sensing disks to the area of Voronoi polygons generated by sensor nodes [3].

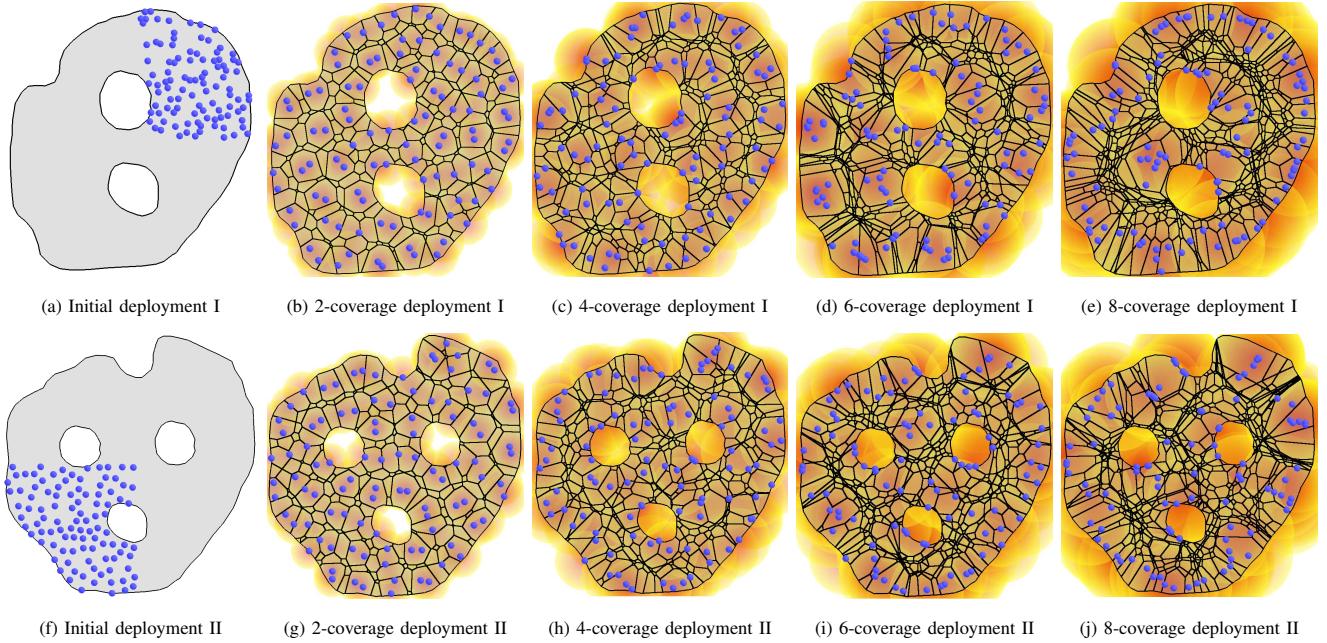


Figure 8. Adaptability of LAACAD to arbitrarily shaped areas and obstacles as well.

sensing range  $r$ . Given a targeted area  $\mathcal{A}$ , we compute the minimum number of sensor nodes  $N_{k=2}^*$  for 2-coverage as  $N_{k=2}^* = \frac{|\mathcal{A}| \frac{4\pi}{3\sqrt{3}}}{\pi r^2} = \frac{4|\mathcal{A}|}{3\sqrt{3}R^{*2}}$ , here we use  $|\mathcal{A}|$  to replace the area of Voronoi polygons generated by sensor nodes, which leads to an under-estimation of  $N_{k=2}^*$  due to the boundary effect. We simulate large-scale WSNs with size ranging from 1000 to 1600 in a  $1 \text{ km}^2$  targeted area. The result is shown in Table I. In general, the number of nodes deployed by LAACAD is about 15% higher than the minimum value, but the boundary effect is the main reason for this difference. As the boundary effect is not taken into account in [3], extra nodes are needed to cover the vacancies on the boundary due to the mismatch between the congruent deployment and an arbitrarily shaped targeted area. Therefore, we can conclude that LAACAD actually leads to a very good approximation of the min-node 2-coverage problem.

Table I  
THE MINIMUM NUMBER OF SENSOR NODES TO ACHIEVE 2-COVERAGE

$N$	1000	1200	1400	1600
$R^*$ (m)	3.035	2.712	2.523	2.357
$N_{k=2}^*$	836	1047	1210	1386

In [15], Ammari *et al.* propose to decompose a targeted area into adjacent *Reuleaux triangles*, and nodes are deployed in the intersection areas between these triangles (these intersections are termed lens in [15]). According to their derivation,  $\frac{6k|\mathcal{A}|}{(4\pi - 3\sqrt{3})r^2}$  nodes are required to  $k$ -cover  $\mathcal{A}$  where  $k \geq 3$  and  $r$  is the sensing range. Here we compare this feasible deployment with LAACAD. We deploy 180 nodes in a  $1 \text{ km}^2$  area and let all nodes have the same

sensing range  $R_k^*$ . We also compute the number of nodes that deployed according to the strategy proposed in [15] as  $N_k^* = \frac{6k}{(4\pi - 3\sqrt{3})R_k^{*2}}$ .

From the result shown in Table II, it is very clear that LAACAD can use much less nodes to achieve the same level of coverage compared with [15].

Table II  
THE NUMBER OF SENSOR NODES TO ACHIEVE  $k$ -COVERAGE WITH THE STRATEGY PROPOSED IN [15] FOR  $k = 3, 4, \dots, 8$

$k$	3	4	5	6	7	8
$R_k^*$ (m)	8.77	10.21	11.24	12.36	13.39	14.32
$N_k^*$	318	313	323	320	318	318

#### D. Adaptability to Obstacles

Finally, we demonstrate the adaptability of LAACAD to arbitrarily shaped targeted area (with obstacles inside) in Figure 8. The “holes” represent obstacles that mobile sensor nodes cannot move upon. It is clear that LAACAD adapts well to these irregularities and again achieves the even clustering distribution as if the area were regular.

## VI. CONCLUSION

In this paper, we have focused on minimizing the maximum sensing range to achieve load balancing  $k$ -coverage through autonomous deployments (i.e., relying on mobile sensors nodes and the wireless communications among them). We have innovated in applying the  $k$ -order Voronoi diagram in a localized manner, and we have proposed LAACAD (Load bAlancing  $k$ -Area Coverage through Autonomous Deployment) to solve the optimization problem

through a distributed and localized procedure. Our approach significantly extends the existing mobility control methods for solving sensor coverage problem, as we are the first to tackle the problem of  $k$ -coverage autonomous deployment. We have proven the convergence of LAACAD as well as the (local) optimality of its output. We have also explained the close relations between the output of LAACAD and other commonly used optimization objectives, which provides a better understanding of optimal  $k$ -coverage deployments whose theoretical characterizations are hard to obtain under general settings. Finally, our simulation results strongly confirm all the theoretical claims we have made, and they also demonstrate the adaptability of LAACAD to the irregularities of the targeted sensing region.

## REFERENCES

- [1] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A Survey on Sensor Networks,” *IEEE Communication Mag.*, vol. 40, no. 8, pp. 104–112, 2002.
- [2] S. Kumar, T. La, and J. Balogh, “On  $k$ -coverage in a Mostly Sleeping Sensor Network,” in *Proc. of the 10th ACM MobiCom*, 2004, pp. 144–158.
- [3] X. Bai, Z. Yun, D. Xuan, B. Chen, and W. Zhao, “Optimal Multiple-Coverage of Sensor Networks,” in *Proc. of the 30th IEEE INFOCOM*, 2011, pp. 2498 – 2506.
- [4] M. Cardei, M. Thai, Y. Li, and W. Wu, “Energy-Efficient Target Coverage in Wireless Sensor Networks,” in *Proc. of the 24th IEEE INFOCOM*, 2005, pp. 1976–1984.
- [5] Q. Zhao and M. Gurusamy, “Lifetime Maximization for Connected Target Coverage in Wireless Sensor Networks,” *IEEE/ACM Trans. on Networking*, vol. 16, no. 6, pp. 1378 –1391, 2008.
- [6] K. Han, L. Xiang, J. Luo, and Y. Liu, “Minimum-Energy Connected Coverage in Wireless Sensor Networks with Omni-Directional and Directional Features,” in *Proc. of the 13th ACM MobiHoc*, 2012, pp. 1 – 10.
- [7] Z. Zhou, S. Das, and H. Gupta, “Variable Radii Connected Sensor Cover in Sensor Networks,” *ACM Trans. Senor Networks*, vol. 5, no. 1, pp. 8:1–8:36, 2009.
- [8] S. Tang, X. Li, X. Shen, J. Zhang, G. Dai, and S. Das, “Cool: On Coverage with Solar-Powered Sensors,” in *Proc. of the 31st IEEE ICDCS*, 2011, pp. 488 – 496.
- [9] G. Wang, G. Cao, and T. L. Porta, “Movement-Assisted Sensor Deployment,” *IEEE Trans. on Mobile Computing*, vol. 5, no. 6, pp. 640–652, 2006.
- [10] N. Bartolini, T. Calamoneri, T. L. Porta, and S. Silvestri, “Autonomous Deployment of Heterogeneous Mobile Sensors,” *IEEE Trans. on Mobile Computing*, vol. 10, no. 6, pp. 753–766, 2011.
- [11] D. L. Hall and J. Llinas, *Handbook of Multisensor Data Fusion*. CRC Press, 2001.
- [12] J. Luo, L. Xiang, and C. Rosenberg, “Does Compressed Sensing Improve the Throughput of Wireless Sensor Networks?” in *Proc. of IEEE ICC*, 2010, pp. 1–6.
- [13] L. Xiang, J. Luo, and A. Vasilakos, “Compressed Data Aggregation for Energy Efficient Wireless Sensor Networks,” in *Proc. of the 8th IEEE SECON*, 2011, pp. 46–54.
- [14] M. Hefeeda and M. Bagheri, “Randomized  $k$ -Coverage Algorithms For Dense Sensor Networks,” in *Proc of the 26th IEEE INFOCOM*, 2007, pp. 2376–2380.
- [15] H. Ammari and S. Das, “Mission-Oriented  $k$ -Coverage in Mobile Wireless Sensor Networks,” in *Proc. of the 11th ICDCN (LNCS 5935)*, 2010, pp. 92–103.
- [16] K. Dantu, M. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G. Sukhatme, “Robomote: Enabling Mobility in Sensor Networks,” in *Proc. of the 4th ACM/IEEE IPSN*, 404-409.
- [17] B. Bash and P. Desnoyers, “Exact Distributed Voronoi Cell Computation in Sensor Networks,” in *Proc. of the 6th ACM/IEEE IPSN*, 2007, pp. 236–243.
- [18] D. Dong, Y. Liu, K. Liu, and X. Liao, “Distributed Coverage in Wireless Ad Hoc and Sensor Networks by Topological Graph Approaches,” in *Proc. of the 30th IEEE ICDCS*, 2010, pp. 106 –115.
- [19] R. Kershner, “The Number of Circles Covering A Set,” *American J. Math.*, vol. 61, no. 3, pp. 665–671, 1939.
- [20] K. Kar and S. Banerjee, “Node Placement for Connected Coverage in Sensor Networks,” in *Proc. of the 1st IEEE/ACM WiOpt*, 2003.
- [21] X. Bai, S. Kumar, D. Xuan, Z. Yun, and T. Lai, “Deploying Wireless Sensors to Achieve Both Coverage and Connecitvity,” in *Proc. of the 7th ACM MobiHoc*, 2006, pp. 131–142.
- [22] J.-E. Kim, J. Han, and C.-G. Lee, “Optimal 3-Coverage with Minimum Separation Requirements for Ubiquitous Computing Environments,” *Springer Mobile Netw. & Appl.*, vol. 14, no. 5, pp. 556–570, 2008.
- [23] Y.-C. Wang and Y.-C. Tseng, “Distributed Deployment Schemes for Mobile Wireless Sensor Networks to Ensure Multilevel Coverage,” *IEEE Trans. on Parallel and Distributed Systems*, vol. 19, no. 9, pp. 1280–1294, 2008.
- [24] X. Wang, X. Wang, and J. Zhao, “Impact of Mobility and Heterogeneity on Coverage and Energy Consumption in Wireless Sensor Networks,” in *Proc. of the 31st IEEE ICDCS*, 2011, pp. 477–487.
- [25] M. Shamos and D. Hoey, “Cloest-Point Problems,” in *Proc. of the 16th IEEE FOCS*, 1975, pp. 151–162.
- [26] E. Welzl, “Smallest Enclosing Disks (Balls and Ellipsoids),” *Results and New Trends in Computer Science (LNCS 555)*, pp. 359–370, 1991.
- [27] D. Lee, “On  $k$ -Nearest Neighbor Voronoi Diagrams in a Plane,” *IEEE Trans. on Computer*, vol. 31, no. 6, pp. 478–487, 1982.
- [28] Y. Shang and W. Ruml, “Improved MDS-Based Localization,” in *Proc. of the 23rd IEEE INFOCOM*, 2004, pp. 2640–2651.
- [29] F. Li, J. Luo, C. Zhang, S. Xin, and Y. He, “UNFOLD: UNiform Fast On-Line boundary Detection for Dynamic 3D Wireless Sensor Networks,” in *Proc. of the 12th ACM MobiHoc*, 2011, pp. 141–152.