Postprint: k-Coverage Research in WSNs Based on Static and Mobile Sensors

Authors: He Chunlin, Zhao Haijun, Yihong Chen

Date: 2020-09-28T00:00:00+00:00

Abstract

This study addresses the k-coverage problem in wireless sensor networks. First, an over-provisioning factor characterizing network coverage efficiency is defined, based on which the k-coverage problem is analyzed for static sensor networks and fully mobile sensor networks, yielding the over-provisioning factors for both scenarios and the maximum movement distance of mobile sensors in fully mobile sensor networks. Subsequently, a hybrid network structure comprising static sensors and a small number of mobile sensors is proposed, and under this structure, network-size-independent k-coverage is achieved along with a distributed mobility scheduling algorithm for coordinating mobile sensor movements to realize effective coverage. Simulation results indicate that the proposed hybrid network structure not only achieves precise k-coverage but also attains higher coverage compared with other k-coverage algorithms.

Full Text

Preamble

Vol. 38 No. 3
Application Research of Computers

Research on k-Coverage Based on Static and Mobile Sensors in WSNs

He Chunlin[1,2], Zhao Haijun[1,2], Chen Yihong[1,2]
(1. School of Computer, China-West Normal University, Nanchong 637009, China;
2. Nanchong Key Laboratory of Internet of Things Perception & Big Data Analysis, Nanchong 637009, China)

Abstract: This paper studies the k-coverage problem for wireless sensor networks. First, it defines an over-provisioning factor to characterize network coverage efficiency. Based on this, it analyzes the k-coverage problems in static sensor networks and all-mobile sensor networks, deriving the over-provisioning factors for both cases and the maximum movement distance of mobile sensors in all-mobile networks. Furthermore, it proposes a hybrid network structure composed of static sensors and a small number of mobile sensors, achieving k-coverage independent of network size in this architecture and developing a distributed mobile scheduling algorithm to coordinate mobile sensor movements for effective coverage. Simulation results demonstrate that the proposed hybrid network structure not only achieves accurate k-coverage but also provides higher coverage compared to other k-coverage algorithms.

Keywords: wireless sensor networks; k-coverage; mobility; grid points; distributed scheduling; coverage

---

0 Introduction

Wireless sensor networks (WSNs) consist of numerous simple, low-cost sensors that perform tasks such as environmental monitoring, target tracking, or infrastructure surveillance in a self-organized manner [?, ?]. A fundamental research issue in WSNs is the coverage problem. The sensing region of an individual sensor is typically abstracted as a disk with radius r. A region is said to be k-covered when every point in the region lies within the sensing range of at least k sensors [?, ?].

Network deployment is critical to coverage quality. Due to factors such as network size and terrain complexity, achieving optimal deterministic deployment is often impractical. The common approach is to scatter sensors randomly across the region of interest. While this simplifies deployment, it cannot guarantee effective coverage. Mobile sensors can improve network coverage efficiency by repositioning themselves to recover coverage holes, thereby compensating for deployment randomness. Mobile sensors are frequently used for coverage enhancement [?], load balancing [?], and network lifetime extension [?]. However, most studies neither consider movement distance constraints for mobile sensors nor assume they can be recharged repeatedly. Other research attempts to minimize total movement distance or total number of movements across all sensors, which is insufficient because energy cannot be transferred between mobile sensors. Therefore, it is preferable to limit each mobile sensor' s maximum movement distance by moving only a few sensors over short distances, such as through cascaded movement [?]. Reference [?] proposed a mobile scheduling algorithm based on flipping sensors to maximize network coverage, while [?] further formulated this as an optimization problem to minimize sensor displacement across different regions. However, neither [?] nor [?] provided partial coverage area or maximum movement distance analysis. Reference [?] presented a multi-objective k-coverage preservation algorithm that improves network lifetime and coverage by establishing affiliation relationships between sensor nodes and target nodes and enabling energy transfer between different nodes. Reference [?] proposed a three-dimensional surface k-coverage multi-connectivity deployment method that freely selects grid sizes to partition the target area, establishes multi-connectivity relationships between grids, constructs k-coverage sets within grids using a directional gradient probabilistic sensing model, and employs a minimum spanning tree algorithm to construct connected graphs for identifying critical nodes to form bi-connected graphs. Simulation results show this method achieves complete coverage and connectivity while ensuring network robustness. Reference [?] introduced a coordinate-free k-coverage hole detection algorithm that first detects 1-coverage holes and then extends to k-coverage scenarios by finding independently covering node subsets within already-covered target regions and putting those nodes to sleep, reducing network coverage degree by 1. Repeating the 1-coverage hole detection algorithm $k-1$ times reveals all k-coverage hole boundary segments and circumferences. Simulation results demonstrate that this algorithm improves hole detection compared to coordinate-based approaches.

Most existing research on WSNs coverage or k-coverage focuses on static sensor deployment or ignores mobile sensor costs, which neither optimally utilizes sensor resources nor satisfies dynamic requirements in practical WSN monitoring. This paper addresses the more general k-coverage problem in WSNs with coexisting static and mobile sensors.

1.1 System Model

Consider a square sensing region with side length l and area $L = l \times l$ (referred to as network size). Assume $N = \lambda L$ static sensors are uniformly and independently scattered throughout the network. When N is large, the number of static sensors in a region of area A (denoted n_A) follows a Poisson distribution with mean λA [?]:

$$P\{n_A = i\} = \frac{(\lambda A)^i}{i!} e^{-\lambda A}$$

where λ is the static sensor density. Moreover, sensor counts in disjoint regions are independent. Therefore, for sufficiently large networks, the deployment can be approximated as a stationary Poisson point process. In subsequent derivations, we directly adopt the Poisson point process assumption. Each static sensor covers a disk region of radius $r = \frac{1}{\sqrt{\pi}}$, meaning each sensor covers a unit area disk. A region is k-covered when every point in the disk is covered by at least k sensors. To ensure connectivity under full coverage, we assume the sensor communication range exceeds $2r$. Mobile sensors are uniformly and independently distributed in the network with total quantity $M = \Lambda L$, where Λ is the mobile sensor density. Mobile sensors have the same coverage range as static sensors. Due to energy and cost constraints, each mobile sensor moves only once over a limited distance to compensate for coverage holes. We assume mobile sensors have sufficient energy to maintain at least the same sensing and communication duration as static sensors after relocation. The objective is to ensure the entire region is k-covered, where k is predetermined by network operators before deployment.

1.2 Over-Provisioning Factor

We define a new metric called the over-provisioning factor $\eta = (\lambda + \Lambda)/k$, which represents the ratio of total sensor density (static λ plus mobile Λ) to network coverage requirements. Clearly, smaller η indicates more efficient network deployment for k-coverage.

In random deployment, many small regions may have more than k covering sensors while a few regions are covered by exactly k sensors. For large networks, random deployment yields a large over-provisioning factor and low efficiency. For deterministic deployment, the upper bound of η can be obtained by placing sensors on a regular grid. For example, placing sensors on a square grid with side length $d_s = \sqrt{2}r$ provides 1-coverage. For k-coverage, k sensors can be placed at each grid point. Thus, the over-provisioning factor has a lower bound of 1 for any deployment, as the total area of all sensor sensing disks must exceed k times the sensing region size. Therefore, the optimal over-provisioning factor for deterministic sensor networks is $O(1)$.

1.3 k-Coverage with Static Sensors

We now examine the over-provisioning factor for randomly deployed static sensor networks with density λ. According to random coverage process theory, the expected total uncovered area is $Le^{-\lambda}$.

By choosing sufficiently large λ, the percentage of uncovered area $e^{-\lambda}$ can be made arbitrarily small. However, as network size $L \to \infty$, the probability of connected coverage holes larger than unit area approaches 1 for networks with constant sensor density λ. The reason is that when a point in the network has no sensor within distance $2r$, the disk of radius $\frac{1}{\sqrt{\pi}}$ around that point remains uncovered, forming a coverage hole of at least unit area. Such points always exist when the network is not fully covered by sensors with expanded $2r$ range. According to random coverage process theory, with probability approaching 1, a network of infinite size cannot be fully covered by sensors with constant density and range $2r$. Therefore, as network size increases, constant sensor density λ cannot guarantee the absence of large holes, even though most of the network may be covered.

To achieve k-coverage in large networks, static sensor density must increase with network size: $\lambda = \log L + (k+2)\log\log(L)$ [?]. Consequently, the over-provisioning factor for randomly deployed static sensor networks is:

$$\eta_s = \frac{\log L + (k+2)\log\log L}{k}$$

For fixed k, η_s is $O(\log L)$, indicating that coverage efficiency degrades as network size increases in static sensor deployments.

1.4 k-Coverage with All-Mobile Sensors

Now consider network coverage where all sensors are mobile and randomly deployed. These mobile sensors can reposition themselves to provide k-coverage. The key question is the maximum distance each sensor must travel to reach optimal positions.

Theorem 1 gives the maximum movement distance for all-mobile networks.

Theorem 1. Consider an all-mobile sensor network uniformly and independently distributed in a square region of area L. The network can provide k-coverage with over-provisioning factor $\eta_m = \pi/2$, and the maximum distance any mobile sensor moves is $O(\log^{3/4}(kL))$.

Proof. For deterministic deployment, k sensors are placed at identical grid points with side length $d_s = \sqrt{2}r$. We partition the sensing region into a square grid with side length $d_a = \sqrt{2/k}r$, as shown in Figure 1 [Figure 1: see original paper]. The grid point density is $k/2r^2 = \eta_m k$.

First, we prove that placing one mobile sensor at each grid point achieves k-coverage, then bound the maximum movement distance for uniformly distributed mobile sensors to realize this regular grid deployment.

Assume mobile sensors have been repositioned such that exactly one mobile sensor occupies each grid point. Consider interior network regions more than distance r from boundaries. A point is k-covered if it lies within distance r of at least k grid points. According to a lower bound on grid points covered by a circle [?], for a circle of radius r centered at any point, there exist at least $W(k)$ grid points of side length d_a covered, where:

$$W(k) \geq \frac{\pi r^2}{d_a^2} - \frac{2\pi r}{d_a} = \pi k - \sqrt{2\pi k}$$

$W(k)/k$ is monotonically increasing for $k \geq 1$, and $W(k) > k$ when $k \geq 25$. It can be verified that the network is at least k-covered for $1 \leq k < 25$. Thus, placing one sensor at each grid point ensures complete k-coverage of the network interior. To cover points near boundaries, we slightly expand the deployment region to an $(l+2r)\times(l+2r)$ square, which only increases density by a negligible $O(r/l)$ fraction for large networks.

After random deployment, mobile sensors must be repositioned so that each grid point contains exactly one sensor. This is essentially a matching problem between mobile sensors and grid points. The maximum movement distance can be obtained using min-max grid matching theory [**?**]: For an $l \times l$ square region with unit-side grid squares, if $L = l^2$ points are uniformly and independently scattered, there exists a perfect matching between L random points and L grid points with maximum distance $O(\log^{3/4} L)$ [**?**]. In our network, the total grid points are $O(kL)$ times more, and since grid size is $\sqrt{2/k}$ instead of 1, the maximum movement distance becomes $O(\log^{3/4}(kL))$.

In all-mobile networks, mobile sensors compensate for randomness differently than static approaches. While static methods require higher density (proportional to $O(\log L)$) to compensate for network size, mobile sensor networks maintain constant sensor density $\pi/2$.

---

2 Hybrid Network Structure

While all-mobile networks can achieve deterministic deployment with small sensor movements, mobile sensors are significantly more expensive than static ones. To reduce network costs, it is preferable to use a small number of mobile sensors to enhance network performance. This section studies hybrid network coverage, where many static sensors and a small fraction of mobile sensors are deployed with over-provisioning factor $O(1)$, requiring fewer than $\pi/2$ mobile sensors. We further show that for this specific deployment, any mobile sensor' s maximum required movement distance is $O(\log^{3/4} L)$.

2.1 Mobile Sensor Density

We fix static sensor density at $\lambda = 2\pi k$ and partition the network into square cells of side length $d_h = r/\sqrt{k}$. Since the sensing range is r, any sensor within a cell can fully cover that cell. The average number of static sensors per cell is $2\pi k d_h^2 = 2\pi$.

If all cells contain at least k sensors, the network will be k-covered. Due to deployment randomness, some cells may contain fewer than k static sensors. If cell i contains $n_i < k$ static sensors, we say cell i has $v_i = k - n_i$ holes. By Poisson approximation, n_i are independent and identically distributed random variables with mean 2π. The distribution of holes $v_i = [k - n_i]^+$ (where $[x]^+$ means $\max\{x, 0\}$) is:

$$P\{v_i = j\} = \begin{cases} \sum_{m=0}^{k} \frac{(2\pi)^m}{m!} e^{-2\pi} & j = 0 \\ \frac{(2\pi)^{k-j}}{(k-j)!} e^{-2\pi} & 0 < j \leq k \end{cases}$$

The expected number of holes in a cell is:

$$E\{v_i\} = \sum_{j=1}^{k} j \cdot \frac{(2\pi)^{k-j}}{(k-j)!} e^{-2\pi} = e^{-2\pi} \sum_{l=0}^{k-1} (k-l) \frac{(2\pi)^l}{l!}$$

Using Stirling' s approximation $k! \approx \sqrt{2\pi k}(k/e)^k$, and since the error order is $O(e^{1/12k})$, as $k \to \infty$:

$$E\{v_i\} \to \frac{2\pi}{k}$$

When static sensor density is $\lambda = 2\pi k$, the ratio of mobile to static sensor density is $\Lambda/\lambda \approx 1/k$. As k increases, only a small fraction of mobile sensors is needed to fill holes because the Poisson distribution of static sensors in a cell becomes more concentrated around the mean k.

For any large network and integer k, the over-provisioning factor is $\eta_h = (\lambda + \Lambda)/k \leq 2\pi + \frac{2\pi}{k}$.

2.2 Mobile Sensor Movement Distance

In hybrid networks, mobile sensors must fill holes in each cell, requiring one-to-one matching between mobile sensors and holes. The matching construction proceeds in two steps: (a) match mobile sensors to grid points of side length $d_s = \sqrt{2/k}r$ with maximum distance $O(\log^{3/4}(kL)/\sqrt{\Lambda})$, which decreases with Λ; (b) match holes to grid points of side length $d_h = r/\sqrt{k}$. Theorem 2 gives the maximum matching distance.

Theorem 2. Consider a square network of area L where ΛL holes are independently and identically distributed in cells of side length $d_h = r/\sqrt{k}$ according to equation (5). There exists a matching between holes and grid points of cells with side length d_h having maximum distance $O(\log^{3/4} L)$.

Based on the preceding analysis, Theorem 1, and space limitations, we omit the detailed proof of Theorem 2. Thus, we construct a one-to-one matching between mobile sensors and holes with maximum distance $O(\log^{3/4} L)$.

Table 1 compares sensor densities and movement distances for three network structures.

Table 1. K-covering sensor density and moving distance for three different network structures

Network Structure	Static Sensor Density	Mobile Sensor Density	Maximum Moving Distance
Static Network	$O(k \log L)$	0	0
All-Mobile Network	0	$\pi k/2$	$O(\log^{3/4}(kL))$
Hybrid Network	$2\pi k$	2π	$O(\log^{3/4} L)$

2.3 Distributed Mobile Scheduling Algorithm

Simple greedy scheduling that moves mobile sensors to the nearest holes cannot fill all holes with short-distance movements. Since the matching problem is a special network flow problem, we can solve mobile scheduling distributively using a network flow structure.

2.3.1 Mobile Scheduling Problem Formulation We formulate mobile scheduling as follows: For each cell i, let n_i be static sensors and m_i be mobile sensors. The number of holes in cell i is $v_i = [k - n_i]^+$. The network flow problem is constructed as shown in Figure 2 [Figure 2: see original paper]. Build a graph $G(V, E)$ where each cell is a vertex. When the center distance between cells i and j is less than $D = O(\log^{3/4} L)$ (the maximum movement distance), add a directed edge (i, j). Thus, mobile sensors can move between cells when their distance is less than D. Let x_{ij} denote the number of mobile sensors moving from cell i to cell j. The mobile scheduling problem becomes:

$$\begin{aligned} \text{Minimize} \quad & \sum_{i,j} c_{ij} x_{ij} \\ \text{subject to} \quad & \sum_j x_{ji} - \sum_j x_{ij} \geq v_i - m_i, \quad \forall i \\ & \sum_j x_{ij} \leq m_i, \quad \forall i \\ & x_{ij} \geq 0 \end{aligned}$$

where c_{ij} is the movement cost. Equation (9) is the flow conservation condition, requiring that the net inflow to cell i (mobile sensors moving in minus moving out) exceeds its required mobile sensors (holes minus initial mobile sensors). This ensures the final number of mobile sensors in cell i after movement will exceed hole count. Equation (10) ensures the total number of mobile sensors moving out of cell i does not exceed its initial count. If all $c_{ij} = 1$, the optimal solution yields minimum movement count. If c_{ij} is the inter-cell distance, we obtain minimum total movement distance.

Since v_i and m_i are integers, the optimal solution x^*_{ij} is integer-valued, meaning moving exactly x^*_{ij} mobile sensors from cell i to cell j ensures each cell has at least k sensors.

2.3.2 Distributed Mobile Algorithm We propose a distributed mobile algorithm to implement the scheduling. First, we present a mobile algorithm to find a feasible schedule that fills all holes without minimizing total cost, then a push-relabel algorithm to obtain minimum-cost flow.

Assume each mobile or static sensor knows its location and which cell it belongs to. After deployment, sensors within the same cell i communicate to compute v_i and m_i. Each cell selects one mobile or static sensor as an agent to maintain necessary cell information during algorithm execution. The agent of cell i also communicates and exchanges information with neighbors. If an empty cell contains no sensors, we randomly assign a mobile sensor from an adjacent cell as its agent. Since empty cells can be filled within maximum distance D, at least one mobile sensor exists within distance D. If an empty cell disconnects the network, the nearest mobile sensor moves to it before algorithm execution.

The push-relabel algorithm iteratively pushes excess flow from a vertex to lower-height neighbors or relabels the vertex when pushing is impossible. Repeating this until all cells have no excess flow constitutes the distributed algorithm. Pseudocode for the mobile algorithm and push-relabel algorithm are shown in Algorithms 1 and 2.

Algorithm 1. Mobile Algorithm for Cell i 1. Collect cell information v_i and m_i 2. Set heights $h(i_{in})$ and $h(i_{out})$ of vertices i_{in} and i_{out} to 0 3. Set excess $e(i_{in}) = 0$, excess $e(i_{out}) = m_i - v_i$ 4. While excess exists at any vertex 5. Call push-relabel(i_{in}) 6. Call push-relabel(i_{out}) 7. Update heights of neighboring cells within distance D 8. End while 9. Send mobile sensors to neighboring cell j according to flow on arc (j_{out}, i_{in})

Algorithm 2. Push-Relabel Algorithm (for vertex u**)** 1. If $e(u) > 0$ 2. While $e(u) > 0$ and exists arc (u, w) with $h(u) = h(w) + 1$ and residual capacity $cap(u, w) > 0$ 3. Push $y = \min\{e(u), cap(u, w)\}$ to cell associated with w via arc (u, w) 4. $e(u) = e(u) - y$ 5. $e(w) = e(w) + y$ 6. Update $cap(u, w)$ 7. End while 8. If $e(u) > 0$ 9. Relabel $h(u) = h(u) + 1$ 10. Broadcast $h(u)$ to neighboring cells within distance D 11. End if 12. End if

The algorithm limits mobile sensors moving out of a cell by constraining total flow capacity through the cell. Each cell i is split into two vertices i_{in} and i_{out} representing input and output vertices. The input vertex connects to the output vertex via a unidirectional arc (i_{in}, i_{out}) with zero cost and capacity m_i, which is the upper bound of mobile sensors that can leave cell i. The output vertex i_{out} connects to input vertices j_{in} of neighboring cells via unidirectional arcs (i_{out}, j_{in}). In the push-relabel algorithm, each cell maintains two vertices and only needs to know heights of neighboring vertices within distance D to perform push or relabel operations. The push process between i_{in} and i_{out} within the same cell is similar to inter-cell pushes but requires no message passing. Each cell sends mobile sensors to neighboring cells based on incoming flow from its input vertices.

3.1 All-Mobile Networks

We first consider networks where all sensors are mobile. In simulations, we uniformly randomly deploy $M = \Lambda L$ mobile sensors in a network of area L, with Λ fixed at $\pi/2$. Mobile sensors are matched to M grid points on a grid of side length $d_s = \sqrt{2/k}r$ to provide full coverage. Network size varies from 10×10 to 50×50 grids. Repeating 10^5 randomly generated topologies yields the probability of infeasible matching for different maximum movement distances D.

Figure 3 [Figure 3: see original paper] shows this probability for different network sizes, with D normalized by grid size d_s. The probability drops rapidly from 1 to 0 as movement distance increases from $1.5d_s$ to $3.5d_s$, demonstrating a critical threshold phenomenon in random geometric graphs. Larger maximum movement distances facilitate feasible matching. Notably, larger networks require smaller maximum movement distances relative to network size. For example, in a $l = 50d_s$ network, mobile sensors need only move at most $3.5d_s$ to form a regular grid deployment—less than one-tenth of the network size. The difference is more pronounced for larger networks because movement distance scales as $O(\log^{3/4} L)$.

3.2 Hybrid Networks

In hybrid network simulations, we partition the region into cells of side length $d_h = r/\sqrt{k}$. We uniformly deploy $N = \lambda L$ static sensors and $M = \Lambda L$ mobile sensors with $\lambda = 2\pi k$ and select M to exactly fill all holes. Mobile sensors in a cell can move to cells within distance D.

Figure 4 [Figure 4: see original paper] shows the probability of infeasible mobile scheduling to fill all holes for different network sizes, with D normalized by d_h. The maximum movement distance grows slowly with network size. Since $d_h = 0.5d_s$, the actual movement distance in hybrid networks is comparable to the all-mobile case when $k = 1$. Figure 5 [Figure 5: see original paper] shows results for different k values in a network of 900 cells. The required maximum movement distance decreases slightly as k increases, with curves for $k = 50$ and

$k = 100$ nearly overlapping. When k is small, maximum movement distance is affected by both matching distances (mobile sensors to grid points and grid points to holes). As k increases, the first matching distance diminishes with increasing mobile density, leaving the second matching distance (holes to grid points) as the limiting factor, which becomes independent of k.

Figure 6 [Figure 6: see original paper] compares coverage rates for k-coverage implementation using our hybrid network versus algorithms from [?], [?], and [?] when $k = 10$. Our hybrid network k-coverage is optimal, improving coverage by approximately 21%, 26%, and 32% compared to [?], [?], and [?] respectively. This improvement stems from the hybrid structure leveraging static sensors' fixed coverage advantages while precisely matching mobile sensors to coverage holes through fine-grained grid partitioning and optimal movement distances.

---

4 Conclusion

This paper conducted an in-depth study of the k-coverage problem in sensor networks. We first defined an over-provisioning factor to indicate WSN deployment efficiency. Building upon analyses of static and all-mobile sensor networks for k-coverage, we proposed a hybrid network structure combining static sensors with a small number of mobile sensors, along with a distributed scheduling algorithm for mobile sensor coordination. Simulation results demonstrate that the proposed hybrid network structure not only achieves precise k-coverage but also provides higher coverage rates compared to other k-coverage algorithms.

References

[1] Fang Deliang. Mission Planning of Multisensor Collaborative Surveillance in Sensor Networks [D]. Zhengzhou: PLA Information Engineering University, 2017.

[2] Liao W H, Kuai S C, Lin M S. An Energy-Efficient Sensor Deployment Scheme for Wireless Sensor Networks Using Ant Colony Optimization Algorithm [J]. Wireless Personal Communications, 2015, 82(4): 2135-2150.

[3] De P, Sarac K, Chandrasekaran R. Heuristics for 2-coverage multi point relay problem in wireless ad hoc and sensor networks [C]//Proc of 2016 14th International Symposium on Modeling and Optimization in Mobile, Ad-Hoc and Wireless Networks. Tempe, USA: IEEE Press, 2016: 1-8.

[4] Shan A X, Xu X H, Cheng Z M. Target Coverage in Wireless Sensor Networks with Probabilistic Sensors [J]. Sensors, 2016, 16(9): 1372-1385.

[5] Fan Xinggang, Yang Jingjing, Wang Heng. Algorithm for Enhancing Probabilistic Coverage in Wireless Sensor Network [J]. Journal of Software, 2016,

27(2): 418-431.

[6] Gao Jianming, Zhu Xiaohua. A load balancing-based multi-task scheduling scheme in wireless sensor networks [J]. Computer Applications and Software, 2016, 33(8): 157-162, 166.

[7] Wang Yuanhao. Modeling and Analysis For The Lifetime of Wireless Sensor Networks [D]. Xian: XIDIAN University, 2016.

[8] Saha D, Das N. Self-organized area coverage in wireless sensor networks by limited node mobility [J]. Innovations in Systems and Software Engineering, 2016, 12(3): 227-238.

[9] Ates E, Kalayci T E, UgUr A. Area-priority-based sensor deployment optimisation with priority estimation using K-means [J]. IET Wireless Sensor Systems, 2017, 7(6): 198-207.

[10] Dash D, Dasgupta A. Distributed restoring of barrier coverage in wireless sensor networks using limited mobility sensors [J]. IET Wireless Sensor Systems, 2017, 7(6): 198-207.

[11] Gao Chunling, Li Zi, Sun Zeyu, et al. Multi-target k-Coverage Preservation Algorithm in Wireless Sensor Network [J]. Computer Engineering, 2016, 42(7): 59-64, 71.

[12] Wang Dandan, Xu Tingrong. k-coverage and multi connectivity method for 3D surface wireless sensor network [J]. Application Research of Computers, 2018, 35(7): 2110-2113.

[13] Ma W Y, Yan F, Zuo X Z, et al. Coordinate-free k-coverage hole detection algorithm in wireless sensor networks [J]. Journal of Southeast University (English Edition), 2019, 35(1): 8-15.

[14] Lambrou Theofanis P. Optimized Cooperative Dynamic Coverage in Mixed Sensor Networks [J]. ACM Transactions on Sensor Networks, 2015, 11(3): 1-35.

[15] Jayakar S, Strichartz R. Average number of lattice points in a disk [J]. Communications on Pure and Applied Analysis, 2017, 15(1): 1-8.

[16] Eppstein D, Van Kreveld M, Speckmann B, et al. Improved Grid Map Layout by Point Set Matching [J]. International Journal of Computational Geometry & Applications, 2015, 25(0): 101-122.

[17] Cai G Q. Generation of correlated random variables and stochastic processes [J]. Probabilistic Engineering Mechanics, 2018, 52: 40-46.

Note: Figure translations are in progress. See original paper for figures.