Wireless Pers Commun (2008) 45:231–263
DOI 10.1007/s11277-007-9411-z

Efficient Sensor Deployment Control Schemes and Performance Evaluation for Obstacle and Unknown Environments

Hsu-Yang Kung · Chung-Ming Huang · Hao-Hsaing Ku

Published online: 7 November 2007

Abstract Deployment is a fundamental issue for wireless sensor networks (WSNs). A well-designed deployment control method not only directly influences the number of deployed sensors, but also influences on data accuracy and network topology. Three widely discussed deployment methods are random deployment, deterministic deployment and deployment by graphic theory. Most related works have focused on the maximal deployment area problem, but few studies have considered efficient methods to solve the k-coverage problem. Moreover, such methods have high time complexity, making them unsuitable for k-covered sensor deployment. To achieve scalable and efficient deployment, this study presents two new topology deployment methods, namely the slow-start method (SSM) and square-encircled method (SEM). The proposed deployment methods can yield k-covered scenarios with minimal overlapping areas, by three different coverage sensors. SSM and SEM are without needing to pre-analyze unknown or unsafe environments when deploying a k-coverage area. Deploying and satisfying each layer until k layers are obtained requires guaranteeing k coverage. The proposed methods have time complexities of $O(n^2)$, making them suitable for WSNs. Moreover, this study first presents nine Construct Performance Evaluation (CPE) factors to evaluate the total costs of a WSN. Finally, this study evaluates the total deployment costs through CPE factors, and analyzes their performance. The simulation results clearly indicate the efficiency and effectiveness of the proposed deployment methods.

H.-Y. Kung (✉)
Department of Management Information Systems, National Pingtung University of Science and Technology, Pingtung, Taiwan, ROC
e-mail: kung@mail.npust.edu.tw

C.-M. Huang · H.-H. Ku
Department of Computer Science, National Cheng Kung University, Tainan City, Taiwan, ROC

C.-M. Huang
e-mail: huangcm@locust.csie.ncku.edu.tw

H.-H. Ku
e-mail: kuhh@locust.csie.ncku.edu.tw

Keywords Wireless sensor networks · Deployment · Slow-start · Square-encircled · Evaluate factors

1 Introduction

The fast development of embedded systems, and of wireless and mobile networks, has enhanced the convenience of daily life. Wireless sensor network (WSN) applications have attracted particular interest. WSNs can perform monitoring, sensing or various sensor-enabled applications based on different terrain features, such as natural disaster detection and prevention, pervasive healthcare systems and military applications [5,13,16,22,26]. The related studies of sensor networks cover a wide variety of topics, including the deployment problem, coverage problem, topology setting, routing problem, power consumptions, data stored and Quality of Service (QoS) [1,8,13,15–17,21,22,27,28].

Most previous studies on the deployment problem have considered random deployment in unknown environments by adjusting the locations of deployed sensors to obtain k coverage [15,27]. The k coverage problem involves determine whether an area with sensors deployed is sufficiently k-covered, such that each point in the target area is covered by at least k sensors, where k denotes a given parameter [11,12]. Although k coverage can be obtained, the deployment process requires many calculations or highly complex computation methods. A few studies have presented well-designed k coverage deployment control methods that are efficient and suitable for WSNs. Hence, random deployment cannot ensure the scalability and minimum overlapping of sensors deployed [6]. To reduce the time complexity of deployment, this study deploys three different coverage sensors in an unknown environment. An unknown environment means an area within which the number of obstacles and the locations of obstacles are unknown for sensor deployment. The unknown environment is near the real sensor deployment environment. Related studies indicate that the coverage problem is formulated as a decision problem [11,12].

Performance evaluation factors that must be considered when studying sensor network deployment include the maximum deployment area, the deployment reliability and the deployment method. The aims of sensor deployment are minimum deployment costs, minimum power consumption, highest reliability, deployment accuracy and data accuracy, and lowest communication cost, flexibility and efficiency [2,9,15,20,23,24]. Additionally, the deployment method must consider the relationship among coverage areas, power consumptions and scalability, since the network sensors have limited power. However, previous studies generally do not balance the relevant factors and goals. Moreover, many deterministic methods only consider the un-deployed parts of effective areas, or the areas that need to increase the reliability. Most deployment methods are too complex to consider all performance factors in deployment of the sensors in a network [4,7,19,25]. Thus, obtaining k-coverage is an NP-complete problem [2,14]. The solution to the high-density deployment problem lies in high-complex-computation and high-time-complexity algorithms.

This study presents two low-time-complexity methods to solve the k-covered deployment problem based on the concept of divide-and-conquer. This means transforming three-dimensional space down to two-dimensional spaces, where a space is composed of k planes. Effective areas are then deployed to obtain k-coverage in a time of Non-NP-complete complexity, enabling low-cost deployment by the proposed Slow-start method (SSM) and square-encircled method (SEM).

Since many sensors are available in the market, with different covering range for each type, this study chooses three sensors with different coverage to fully deploy an unknown space with obstacles. In the deployment process, the effective areas are first detected, and the sensor with the biggest coverage is deployed to lengthen the service life of the power supply. The medium and small sensors are then deployed by the proposed methods SSM and SEM.

The rest of this study is organized as follows. Section 2 presents related works describing three different deployment methods. Section 3 describes the proposed performance evaluation factors for WSNs. Section 4 describes the proposed slow-start and square-encircled methods. Section 5 discusses the testing and analysis of SSM and SEM. Section 6 describes a related application with k coverage scenario. Finally, Sect. 7 draws conclusions.

2 Related Works

The k-covered deployment is generally performed by a deterministic method, which requires low power consumption and time complexity [2,3,9,10,29]. Wireless sensor network (WSN) deployment methods can be roughly divided into the following three types: (1) random deployment, (2) deterministic deployment and (3) deployment by graphic theory. Related works are described as follows.

2.1 Random Deployment

Random deployment means deployment of sensors in an unknown space by random distribution. Various adjusting mechanisms, including calculating the moving distance or sensing cost, can be adopted to achieve k-coverage [11,12,14,18,30]. Huang and Tseng presented an evaluation mechanism that detects the covered area of each sensor to minimize the number of covered sensors. The lowest guaranteed coverage of the whole area is calculated as the intersections of the numbers [11,12].

Kumar et al. [14] presented a theory of guaranteed k-coverage by grid, uniform and Poisson distributions. The proposed control operations are as follows. A unit matrix of $\sqrt{n} \times \sqrt{n}$ is deployed by these three distribution methods to produce a critical value of $np\pi r^2/log(np) = 1$, where n denotes the number of deployment points in random distribution. In the Poisson distribution, n denotes the deployment density. If $n > 1$, then the area is covered, while if $n < 1$, then the area is uncovered.

Zou and Chakrabarty [30] presented a Virtual Force Algorithm (VFA). VFA adopts force different values, namely total attractive forces, total repulsive forces, magnitude of range and orientation θ, to adjust the locations of related sensors. VFA based on the concept of triangle theorem and moving sensors location to regulate and manage a randomly deployed area to obtain a maximum and optimized coverage. The deployment methods of VFA can achieve a maximum deployed area by adjusting sensors, but cannot easily achieve a k-coverage scenario. When deploying a k-coverage area, VFA must increase the number of sensors and adjust all sensors until k-coverage is achieved.

Ma and Aylor [18] built a power saving structure by choosing different types of sensors and adopting the hub-spoke network structure and Resource Oriented Protocol (ROP). The proposed control method determines the states of the mobile and isolated sensors that consume the fewest resources before integrating them into the whole network, and therefore achieves the lowest resource consumption in medium and small sensors.

In general, although the random deployment method can easily deploy sensors in the effective area, its adjustment cost is too high to guarantee k-coverage.

2.2 Deterministic Deployment

Deterministic deployment recognizes unknown areas before deploying sensors in advance. The locations of the sensors are adjusted according to the pre-calculated results to obtain the ideal deployment effects. The deterministic deployment procedure can be divided into the following two modes.

(1) *Grid deployment*: The fundamental control principle of grid deployment is to divide an area into many sub-areas. The sensors are responsible for detection and moving of locations until the sensing range meets certain conditions, e.g., achieving the largest coverage area. The deployment is performed until its completion to deploy the whole area.
(2) *Deterministic deployment*: Wang et al. [25] presented the standard deterministic deployment method, which advanced the proxy-based sensor network deployment protocol by the computation method or direct environmental detection before deployment. In this study, the proxy server calculates the lowest moving costs and the biggest coverage. The deterministic deployment method then moves sensors to targets to reduce the deployment costs until finding out the suitable target location. However, the study fails to consider how to move in an area full of obstacles or in an unknown area. Additionally, the system has a long computation time and high computation costs.

2.3 Deployment by Graphic Theory

Voronoi graphic theory is the most commonly adopted computation method to ensure full deployment in an effective area by precise graphic theory. Bejerano [3] adopted the Polynomial time Approximation Algorithm with integration of cluster graph and divide-and-conquer concepts. Prof. Bejerano presented a method of clustering and disjoining sensors that are already deployed, and then adjusting sensors among clusters according to terrain features. The proposed algorithm can achieve the minimum time complexity in polynomial time under the specific and strict conditions.

Most studies calculate the location of each sensor by Voronoi graphic theory [5,12,19] to detect the largest area with the smallest possible number of sensors. Graphic theory can be adopted in different terrains, and can deploy many sensors in areas with high reliability. Hence, many previous studies have adopted graphic theory to study deployment. However, the effective area of deployment must be calculated by using graphic theory. Although obstacles can be avoided, minimizing the overlap involves costly and complex computations.

With regards to effective area deployment, most previous studies do not specifically discuss deployment problem in WSNs, but instead consider sensor network applications. Hence, this study performs an in-depth evaluation of sensor network deployment. This study discusses two modes of sensor network deployment, namely the slow-start method (SSM) and the square-encircled method (SEM). Both modes can achieve the effectiveness of the least overlapping and largest reliable coverage without a previously known deployed area or number of obstacles. Additionally, reliable k coverage can be obtained with the time complexity reduced to that in the deterministic deployment method.

3 Construct Performance Factors Analysis

This section presents nine factors to evaluate activities in WSNs, e.g., deployment, topology setting and communications. Previous relevant studies did not consider all factors that

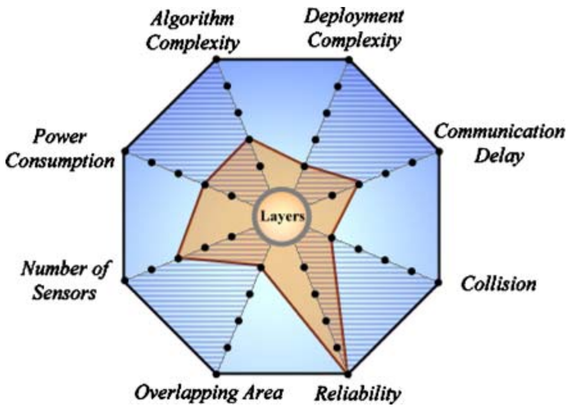

Fig. 1 Relationships among the nine CPE factors

influence deployment. For instance, Zou and Chakrabarty et al. [30] found that the location of deployment influence coverage, communication cost and resource management, and have discussed the relationships among these factors in detail. Additionally, Wang et al. [25] took repeated steps to detect hostile and obstacle-filled environment together with inner communications among sensor networks, before moving the sensors to the uncovered area to obtain a large coverage. They also discussed the relationship between communication cost and coverage. Subramanian and Fekri [23] solved adopted a sleep scheduler to solve communication problems, especially package collision, avoidance and package transportation cost reduction.

3.1 Definition of Construct Performance Evaluation Factors, CPE Factors

This study considered nine Construct Performance Evaluation (CPE) factors in order to provide a complete picture of cost evaluation of the wireless sensors network setup. The nine CPE factors are: (1) algorithm complexity, (2) deployment complexity, (3) communication delay, (4) collision, (5) reliability, (6) overlapping area, (7) number of sensors, (8) power consumptions and (9) number of layers. The first eight factors were the periphery performance factors with the number of layers defined in the kernel construct performance evaluation factor. Figure 1 illustrates the relationships among the nine CPE factors, which is an example to illustrate the scenario of weight distributions for deployment problem using SSM. Appendix 1 lists all parameters adopted in this study. The nine CPE factors are described as follows:

(1) *Algorithm complexity* (E_{AC}): the complexity of the whole system algorithm including deployment, roundabouts, routing, data transmission and data collection.
(2) *Deployment complexity* (E_{DC}): time complexity of sensor deployment until k-coverage achieved. This value rises with more precise sensor deployment for higher complexity.
(3) *Communication delay* (E_{CD}): the packet transmission delay, which rises as the number of the routed sensors rises.
(4) *Collision* (E_{CO}): the number of collisions, which rises with rising numbers of packets transported on the network.
(5) *Reliability* (E_{RE}): the reliability of network sensing and monitoring, which rises with rising numbers of working sensors.
(6) *Overlapping area* (E_{OA}): a smaller overlapping area saves power resources when the sensors are deployed, while a larger overlapping area improves the reliability.

Table 1 The definitions of CPE Factors

CPE factors	Definition
E_{AC}	Algorithm complexity of whole system
E_{DC}	Deployment complexity for deployment of sensors in a area
E_{CD}	Communication delay for communication of sensors
E_{CO}	Collision for WSNs
E_{RE}	Reliability for detected area
E_{OA}	Overlapping area of sensor coverage areas
E_{SE}	Total number of sensors
E_{PC}	Power consumptions for total sensors
E_{LA}	Number of layers for k-covered scenario

(7) *Number of sensors* (E_{SE}): to reduce the deployment and coverage cost, E_{SE} should be reduced. A larger number of sensors leads to a higher coverage cost, and hence to a greater overlap.
(8) *Power consumptions* (E_{PC}): power consumption is a significant constraint in WSNs, since each sensor has limited computation power under limited power supply and memory. Thus, lower re-transmission, lower collision, and lower overlapping area are required for power saving.
(9) *Number of Layers* (E_{LA}): the determining factor in k-covered deployment, where k denotes the number of layers. The number of layers has a positive correlation with all other periphery performance factors.

Table 1 shows the nine CPE factors, which are discussed in three categories, namely periphery performance factors, kernel construct performance evaluation factor and neighbor performance factors.

3.2 Periphery Construct Performance Evaluation Factors

The periphery performance factors are the eight factors in the periphery of the octagon in Fig. 1. Factors opposite to each other have positive correlation. For instance, E_{PC} is in direct proportion to E_{CO}. The relationships between each pair of factors are defined as follows:

$$\textit{Periphery CPE factor } i = weight_{ij} \times \textit{ the opposite periphery CPE factor } j \qquad (1)$$

Increasing the number of collisions increases the number of retransmissions and the power consumption, according to Eq. 1. Hence, the relationship between E_{PC} and $\mathrm{E_{CO}}$ can be described as $E_{PC} = weight \times E_{CO}$.

3.3 Kernel Construct Performance Evaluation Factor

The kernel CPE factor at the center of the octagon in Fig. 1, i.e. the number of layers factor (E_{LA}), correlates positively with the peripheral CPE factors. Since all peripheral CPE factors are increasingly influenced by the kernel CPE factor as the number of layers rises, the cost rises as more layers are deployed. The number of layers should be considered in topology deployment, communications and wireless sensors network configuration.

Equation 2 describes the relationship between the kernel CPE factor and the other peripheral CPE factors:

$$Periphery\ CPE\ factor\ V_i = W_i \times number\ of\ layers(E_{LA}) \tag{2}$$

where V_i denotes the periphery performance factor i, and W_i denotes the weighing value of performance factor i.

According to Eq. 2, the computation complexity and sensors network reliability increase with increasing numbers of deployment layers. However, the costs of all eight factors, including the total power consumption, also rise. For instance, the relationship between E_{PC} and the kernel CPE factor can be described as $E_{PC} = weight \times E_{LA}$, and similar equations apply to other peripheral CPE factors.

3.4 Neighbor Construct Performance Evaluation Factors

Figure 1 illustrates eight neighboring performance factors covering a triangle containing the kernel performance evaluation factor. The area can be divided into two parts, namely triangles and shadowed triangles.

(1) *Triangles*: the neighboring factors positively correlate with each other according to a definition equation similar to Eq. 1, as indicated in the triangle formed by E_{AC}, E_{DC} and E_{LA}. The others are (E_{PC}, E_{SE}, E_{LA}), (E_{OA}, E_{RE}, E_{LA}) and (E_{CO}, E_{CD}, E_{LA}).
(2) *Shadowed triangles*: the neighboring factors have negative correlation with each other with the same definition equation as Eq. 3, as in the triangle formed by E_{AC} and E_{PC}. The others are (E_{SE}, E_{OA}), (E_{RE}, E_{CO}) and (E_{CD}, E_{DC}).

$$V_i = \frac{W_i}{G_j} \tag{3}$$

In Eq. 3, G_j denotes the neighboring performance evaluation factor of the shadowed triangle V_i.

Equation 3 can illustrate that setting up a power saving mode based on the E_{AC} and E_{PC} factors raises the complexity of the algorithm, making the power impossible to minimize by simple random deployment. Hence, the deployment must be optimized deterministically, resulting in a highly complex algorithm, given by $E_{AC} = W_i/E_{PC}$. Moreover, if E_{PC} is given by $f(n) = nlogn + c$, then the time complexity of E_{AC} is $O(n^2)$. If $W_i = 2$, then the equation is $O(n^2) = 2/f(n)$. The time complexity of E_{AC} varies with E_{PC}.

This study concludes the nine CPE factors with related correlations. Table 2 shows the relationships between the nine CPE factors, labeled "+" for positive correlation, and "−" for negative correlation. Table 2 shows the relations among CPE factors, for example the relation between E_{AC} and other seven factors. As E_{AC} increases, the reliability, deployment complexity and layer also rise, while communication delay, collision, overlapping area, number of sensors and power consumption fall.

Thus, the cost of the wireless sensors network is defined in Eq. 4:

$$\mathrm{Cost(Activities)} = \sum_{i=1}^{9} \mathrm{W_i} \times \mathrm{Value(CPE_i)} \tag{4}$$

Equation 4 indicates the area formed by some activities for a WSN, e.g. deployment and routing. The octagon determined by the nine CPE factors shows the estimated cost of setting up an activity in a WSN. For instance, Fig. 1 shows the total costs of SSM. If each neighbor performance factor is graded into five levels, then the algorithm complexity is 2, the network

Table 2 Relationship among the nine CPE factors

Effective factors	E_{AC}	E_{DC}	E_{CD}	E_{CO}	E_{RE}	E_{OA}	E_{SE}	E_{PC}	E_{LA}
Algorithm complexity (E_{AC})	N	+	−	−	+	−	−	−	+
Deployment complexity(E_{DC})	+	N	−	−	+	+	−	−	+
Communication delay (E_{CD})	−	+	N	+	−	+	+	+	+
Collision (E_{CO})	−	−	+	N	−	+	+	+	+
Reliability(E_{RE})	+	+	−	−	N	+	+	+	+
Overlapping area (E_{OA})	−	+	+	+	+	N	−	+	+
Number of sensors(E_{SE})	−	−	+	+	+	−	N	+	+
Power consumptions(E_{PC})	−	−	+	+	+	+	+	N	+
Number of layers(E_{LA})	+	+	+	+	+	+	+	+	N

reliability is 5, the communication delay is 2, the collision is 1, the deployment complexity is 1, the number of sensors is 3 and the power consumption is 2. Hence, the single layer cost is the area formed with the kernel performance evaluation factor 3. The total cost is the volume of the area formed by all factors. The formula, Cost(Deployment) $= \sum_{\mathrm{i}=1}^{9} \mathrm{W_i} \times \mathrm{Value}(\mathrm{CPE_i})$, focuses on the deployment scheme.

The nine CPE factors are defined to evaluate all problems of WSNs, e.g. deployment problem, routing problem, power saving problem. The formulae are defined as Cost (Deployment) $= \sum_{\mathrm{i}=1}^{9} \mathrm{W_i} \times \mathrm{Value}(\mathrm{CPE_i})$, $Cost(Routing) = \sum_{i=1}^{9} W_i \times Value(CPE_i)$ and $Cost(Power) = \sum_{i=1}^{9} W_i \times Value(CPE_i)$.

4 Topology Deployment

This section adopts the layering concept to guarantee k-coverage. Figure 2a shows the k-covered deployment problem is solved by the traditional method, e.g. random deployment. The whole space must be deployed, and any point in the space must guarantee to be k-covered. However, Fig. 2b shows the full layering coverage based on the concept of divide-and-conquer described herein. To reduce the deployment complexity, the required minimum value of k is set to the number of layers. It means that each layer is deployed individually, then sum of k layers, i.e. $k\text{-}covered = \sum_{i=1}^{k} layer_i$, achieve k-covered as same as that of the traditional method. It is because each layer is deployed sensors from 1st layer to kth layers.

This study presents two coverage-guaranteeing methods for deploying various terrains, namely slow-start method (SSM) and square-encircled method (SEM). No pre-analysis or heavy computation is required to avoid obstacles and guarantee k-coverage. The three different sensors discussed herein, which were deployed in an unknown area with obstacles to achieve minimum overlapping sensing area, can reduce the deployment cost. Figure 3 shows the sensing ranges of the three sensors. The small sensor covers one case; the medium sensor covers nine cases, and the large sensor covers 25 cases. The square within the round area is taken as the effective detection area of each sensor.

Previous studies generally consider one of two deployment methods: complex algorithm [6, 12, 19, 30] and random deployment [14, 18]. Random deployment adopts random distributions to deploy sensors in the whole area, as in Fig. 4a. Figure 4b shows a network where the

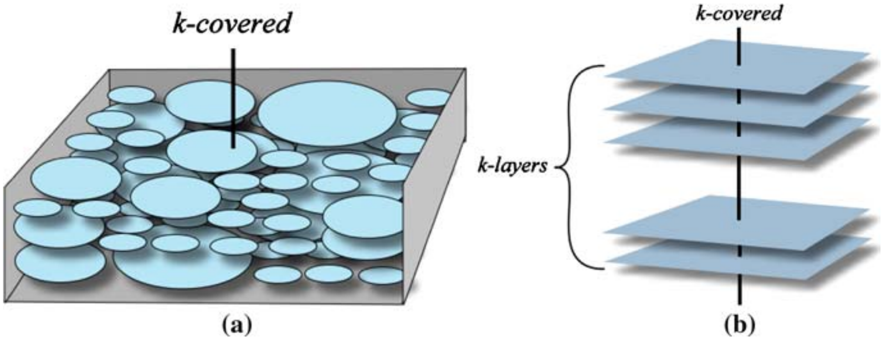

Fig. 2 (**a**) The traditional method of guaranteeing k-coverage. (**b**) Layer-based method of guaranteeing k-coverage

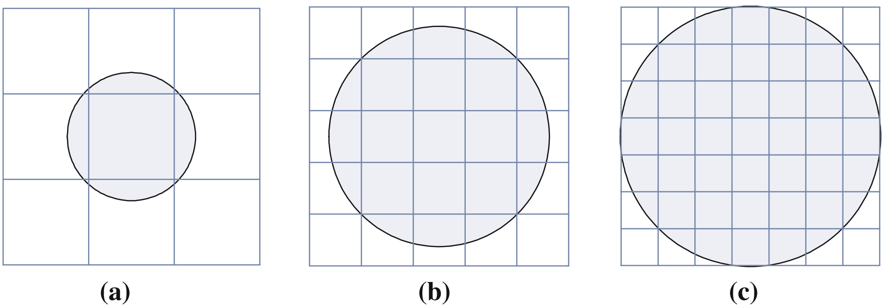

Fig. 3 Sensing areas of (a) small, (b) medium and (c) big sensors

redundant sensors are turned off. The costs of k-coverage are high when adopting random deployment. Although these two methods can achieve k-coverage, the deployment costs are high due to the continuous computation control. Continuous adjustment is initially very power-consuming, and therefore is not appropriate for large-scale wireless sensor networks, but can achieve fairly precise sensor deployment with feedback messages controlling the deployment effectiveness and coverage. However, this method can also lead to feedback implosion, significantly increasing the time complexity of guaranteed k-coverage.

Hence, this study presents the slow-start method (SSM) and square-encircled method (SEM) to guarantee k-coverage in deployment. Slow-start method can quickly find and deploy individual effective space, and the square-encircled method can avoid obstacles in the effective region. The desired k-coverage can thus be achieved quickly. An effective area is defined as an area without obstacles in an unknown environment. Both methods deploy sensors in all effective areas.

4.1 Slow-start method (SSM)

The core mechanism of SSM is fast detection with quick withdrawal for searching new locations when obstacles are detected. Figure 5a shows this process. The first detected location is the center of the whole area. Detection continues rightwards and upwards direction until obstacles are encountered. Thus, the individual effective area rises from 1 to 3. Figure 5b indicates that obstacles are detected at the start. The detecting point then moves down to

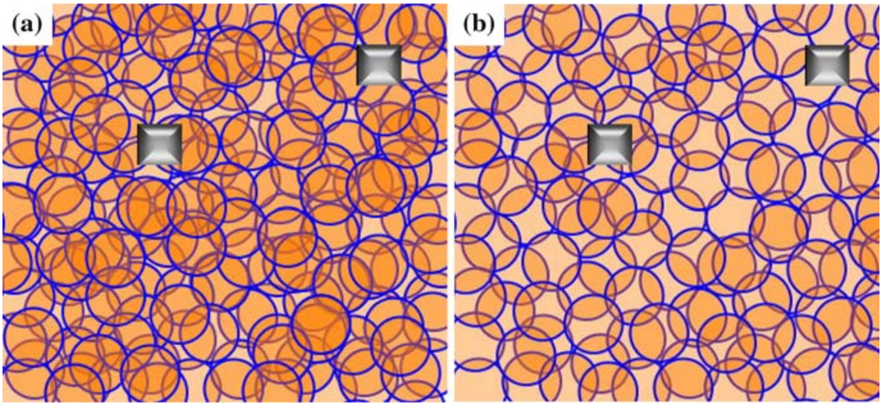

Fig. 4 (**a**) The random deployment scenario. (**b**) 1-Covered scenario after turn off redundant sensors

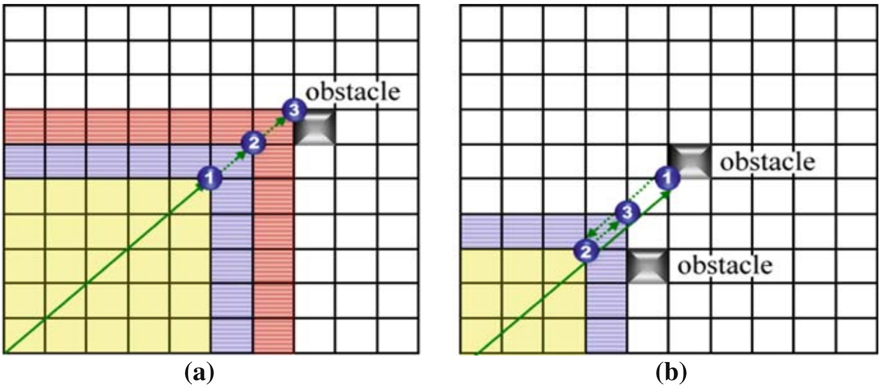

Fig. 5 SSM deployment. (**a**) obstacles met after start, (**b**) obstacles met before start

the point $(x_t', y_t') = \left(\left\lceil \frac{x_t}{2} \right\rceil, \left\lceil \frac{y_t}{2} \right\rceil\right)$, then slowly moves rightwards and upwards until it discovers the largest individual effective area. Figures 6 and 7 show a complete deployment of 1-covered and 2-covered scenarios, respectively, using SSM.

Table 3 shows the execution procedure of the SSM algorithm. The execution steps are explained as follows.

Step 1: Determine whether the current layer is the first layer, in which case the large sensors are deployed.
Step 2: Establish start and finish points of detection.
Step 3: Detect obstacles.
Step 4: Move the target point to $(x_t' + 1, y_t' + 1)$ when no obstacle is encountered. Otherwise, the location of any obstacle is the target point.
Step 5: When deploying an individual effective area with sensors, first deploy the medium sensors, then the small sensors, to lengthen the data transmission time.

Table 3 SSM algorithm

Slow-start method

Parameters notification
    The size of total area is $x \times y$
        x is the length of area
        y is the width of area
    $P_o(0, 0)$ is initial point
    $P(x'_o, y'_o)$ is start point
    $P(x, y)$ is end point
    $P(x'_t, y'_t)$ is temporary end point
    Layer is number of layers
    a, temp are counters
    S_L, S_M, S_S are Large sensors, medium sensors, and small sensors

Define-effective-area($P(x'_o, y'_o), P(x'_t, y'_t)$)
  loop (detect from $P(x'_o, y'_o)$ to $P(x'_t, y'_t)$)
    if (effect area is with obstacles) **then**
      $$(x'_t, y'_t) = \left(\left\lceil \frac{x'_t}{2} \right\rceil, \left\lceil \frac{y'_t}{2} \right\rceil\right)$$
    End if
  End loop
  loop (the increased area without obstacles)
    $(x'_t, y'_t) = (x'_t + 1, y'_t + 1)$
  End loop
  Return (x'_t, y'_t)

Slow-start method (*from Layer 1 to Layer k*)
  if (*Layer=1*) **then**
    set start point $P(x'_o, y'_o) = P_0(0, 0)$
    loop (!sockets are full of $\mathrm{S_L}$ sensors or $\mathrm{S_L}$ sensors use up)
      set S_L to all sockets
      Mark all covered range of S_L
    End loop
  End if
  Set start point $P(x'_o, y'_o) = P_0\left(\left\lfloor \frac{Layer}{x} \right\rfloor, mod\left(\frac{Layer}{y}\right)\right)$
  Put $\mathrm{S_S}$ sensors from $P_o(0, 0)$ to $P(x'_o, y'_o) - 1$
  Set temporary end point $P(x'_t, y'_t) = P\left(\left\lceil \frac{x}{2} \right\rceil, \left\lceil \frac{y}{2} \right\rceil\right)$
  loop (from start point $P(x'_o, y'_o)$ to end point $P(x, y)$)
    Define-effective-area ($P(x'_o, y'_o), P(x'_t, y'_t)$)
    // set sensors into effective area
    loop (from start point $P(x'_o, y'_o)$ to end point $P(x'_t, y'_t)$)
      set S_s or S_M into the effective_area
      Mark effective_area
    End loop
    // to count next start point and end point
    loop (a is from 0 to x)
      $P(x'_o, y'_o) = P(a, (y_t + 1) mod\ y)$
      if ($P(x'_o, y'_o)$ is marked space and $(y'_o + 1)$ is not equal to y) **then**
        $P(x'_o, y'_o) = P(x'_o, (y'_o + 1) mod\ y)$
      else
        $P(x'_o, y'_o) = P((x'_o + 1) mod\ x, (y'_o + 1) mod\ y)$
      End if
    End loop
    temp=minimum $((x - x'_t), (y - y'_t))$
    $P(x'_t, y'_t) = (x'_t + temp, y'_t + temp)$
  End loop
  L*ayer=Layer+1*

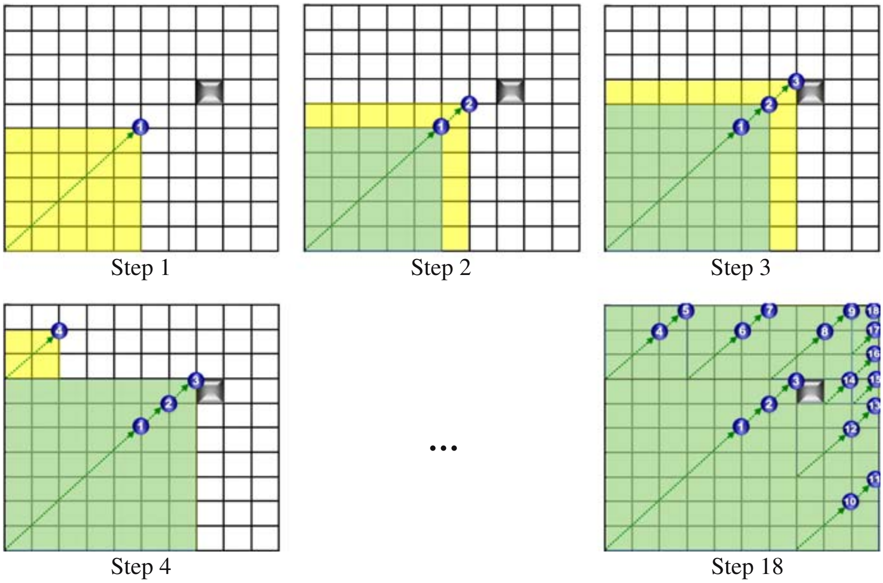

Fig. 6 A 1-covered scenario in 1st plane using SSM

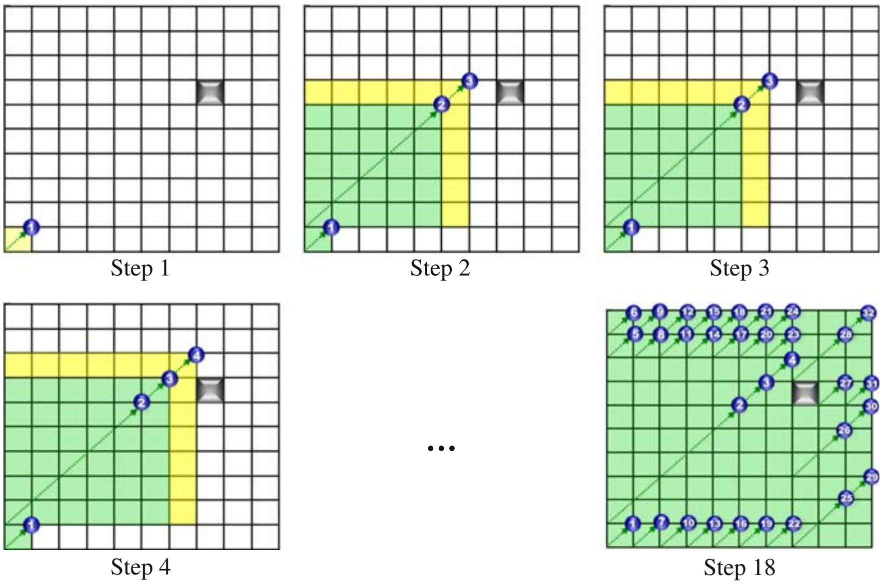

Fig. 7 A 2-covered scenario in 2nd plane using SSM

Step 6: When the obstacle is encountered, move down to the point $(x_t', y_t') = \left(\left\lceil \frac{x_t}{2} \right\rceil, \left\lceil \frac{y_{ti}}{2} \right\rceil\right)$, then slowly move rightwards and upwards.

Step 7: Repeat steps 3–7 until the end of detection and deployment.

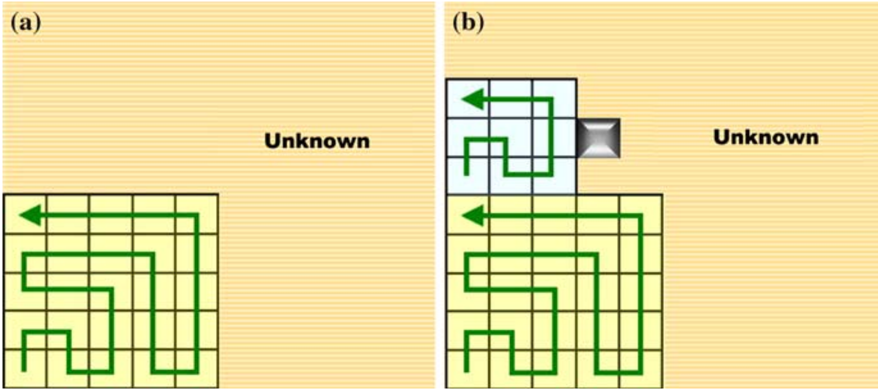

Fig. 8 The deployment of (a) 25 cases (big sensors), (b) nine cases (medium sensors) using Square-encircled method (SEM)

Table 4 SEM circling design

Deployment of square-encircled method
Push effective_area_stack P$(x_c, y_c + 1)$
Push effective_area_stack P$((1 + 2b) + x_c, y_c - (1 + 2b))$
Push effective_area_stack P$(x_c + 1, y_c)$
Push effective_area_stack P$((2 + 2b) + y_c, x_c - (2 + 2b))$

Step 8: Move up one layer, and set a new start point at $P(x'_o, y'_o) = P_0\left(\left\lfloor \frac{Layer}{x} \right\rfloor, mod\left(\frac{Layer}{y}\right)\right)$, with small sensors deployed before the start points.
Step 9: Repeat steps 3–9 until the procedure has run k times, which guarantees k-coverage.

4.2 Square-encircled method (SEM)

SEM is a modification of the traditional Zigzag detection method in which all the effective areas are configured as individual effective areas, which are deployed with big, medium or small sensors. SEM is showed in Fig. 8. Fig. 8a shows the deployment of 25 cases (big sensors), and Fig. 8b shows the deployment of nine cases (medium sensors). The detecting directions follow the sequence of up, right, down and left. The advantage of SEM is that it can deploy an environment without knowing the related information. SEM can efficiently auto-detect the locations of obstacles in unknown environments.

Table 4 shows the detailed design. The target is pushed into the stack, and pops out when it encounters obstacles, until one of the three type sensors is satisfied. Table 5 shows the complete SEM algorithm. Figures 9 and 10 show the complete deployment of 1-covered and 2-covered scenarios, respectively, by SEM.

Step 1: Determine whether the current layer is the first layer, in which case large sensors are deployed to guarantee fundamental transmission deployment.
Step 2: Start to detect obstacles by SEM.
Step 3: If no obstacles are found, then increment the value of the stack when moving. If an obstacle is met, then reset the stack counter. If 25 moves are achieved without

Table 5 SEM algorithm

Square-encircled method

```
Parameters notification
    The size of total area is x × y
        x is the length of area
        y is the width of area
    P_o(0, 0) is initial point
    P(x'_o, y'_o) is start point
    P(x, y) is end point
    P(x_c, y_c) is check point
    Layer is number of layers
    a, b is a counter
    S_L, S_M, S_S are Large sensors, medium sensors, and small sensors
Square-encircled method (from Layer 1 to Layer k)
  if (Layer=1) then
      set start point P(x'_o, y'_o) = P_0(0, 0)
      loop (!sockets are full of S_L sensors or S_L sensors use up)
          set S_L to all sockets
          Mark all covered range of S_L
      End loop
  End if
  Set start point P(x'_o, y'_o) = P(⌊Layer/x⌋, mod(Layer/y))
  Put S_s sensors from P_o(0, 0) to P(x'_o, y'_o) − 1
  Set check point P(x_c, y_c)
  loop (from start point P(x'_o, y'_o) to end point P(x, y))
      loop ((encircled area is satisfied with S_M or S_s)
          if (!obstacles)then
              Push effective_area_stack P(x_c, y_c + 1)
              Push effective_area_stack P((1 + 2b) + x_c, y_c − (1 + 2b))
              Push effective_area_stack P(x_c + 1, y_c)
              Push effective_area_stack P((2 + 2b) + y_c, x_c − (2 + 2b))
          Else
              Pop (from top to 1)
          End if
      End loop
      loop(from top to empty)
          Mark effective_area_stack()
          Pop effective_area_stack()
      End loop
      loop (a is from 0 to x)
          P(x'_o, y'_o) = P(a, (y_t + 1)mod y)
          if (P(x'_o, y'_o) is marked space and (y'_o + 1) is not equal to y) then
                  P(x'_o, y'_o) = P(x'_o, (y'_o + 1)mod y)
          else
                  P(x'_o, y'_o) = P((x'_o + 1)mod x, (y'_o + 1)mod y)
          End if
      End loop
  End loop
  Layer=Layer+1
```

encountering obstacles, then a big sensor is set up. A medium sensor is set up after nine moves, and a small sensor is set up after one move.

Step 4: Move the start point upwards when the obstacle is encountered.

Step 5: Repeat steps 3–4 until the end of detection and deployment.

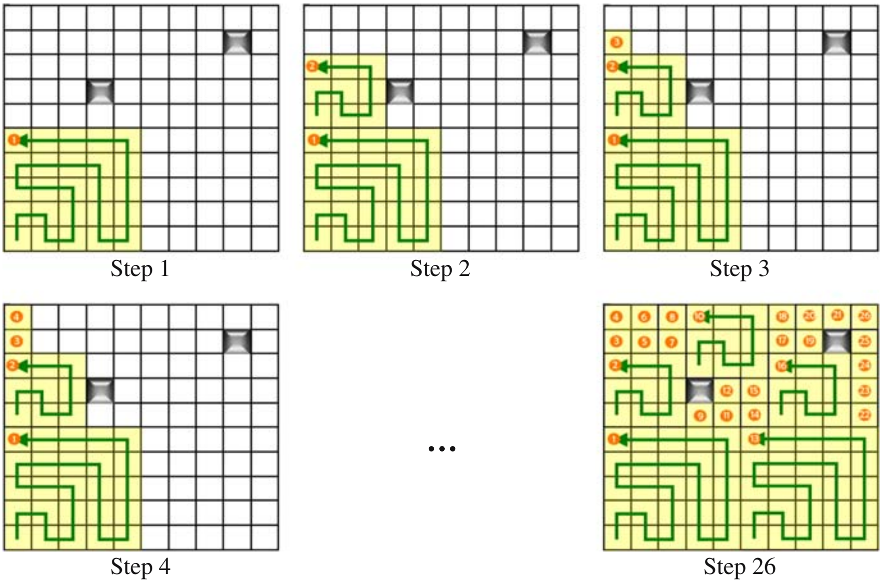

Fig. 9 A 1-covered scenario in 1st plane using SEM

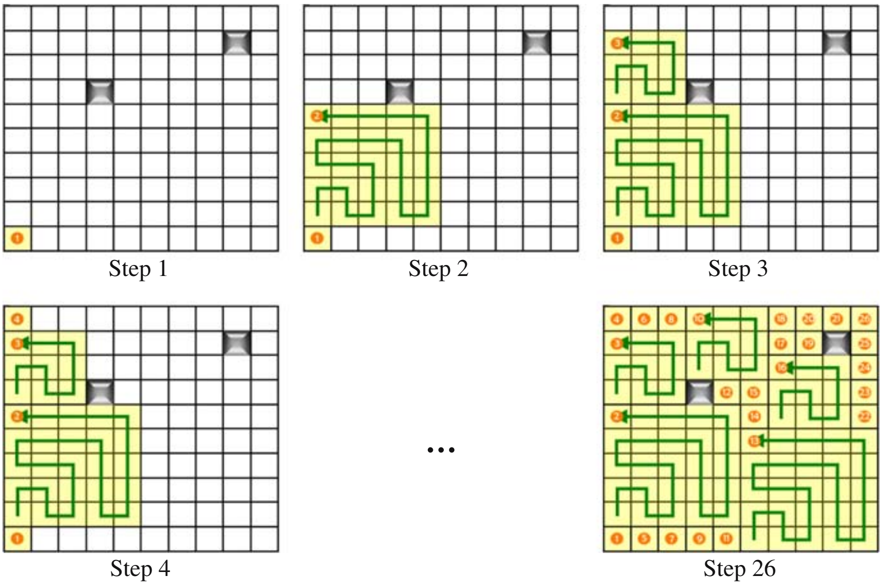

Fig. 10 A 2-covered scenario in 2nd plane using SEM

Step 6: Move up the upper layer to set up a new start at point $P(x_o', y_o') = P_0\left(\left\lfloor \frac{Layer}{x} \right\rfloor, mod\left(\frac{Layer}{y}\right)\right)$,with small sensors deployed before the start.

Step 7: Repeat steps 3–6 until the procedure has been run k times to guarantee k-coverage.

4.3 Analysis the Complexities of SSM and SEM

The k coverage problem is primarily a decision problem. This study indicates that the SSM and SEM can manipulate three different coverage sensors to obtain a k-covered scenario. The time complexity of the proposed methods was analyzed as follows. Appendix 2 presents the detailed analysis.

4.3.1 SSM

The time of executing SSM is $(t_i \times n) = t_1 n + t_2 n + t_3 n + t_4 n(t_5 n + t_6 n) = (t_4 t_5 + t_5 t_6)n^2 + (t_1 + t_2 + t_3)n$, where t_i denotes the execution time for evaluating the algorithm. The time complexity of SSM with k-coverage scenario is as follows.

$$O(k((t_4 t_5 + t_5 t_6)n^2 + (t_1 + t_2 + t_3)n)) = O(k(n^2)) = O(kn^2) = O(n^2).$$

4.3.2 SEM

The time of executing SEM is $t_1 n + t_2 n(t_3 n + t_4 n + t_5 n) = (t_2 t_3 + t_2 t_4 + t_2 t_5)n^2 + t_1 n$. The time complexity of SSM with k-coverage scenario is as follows:

$$O(k((t_2 t_3 + t_2 t_4 + t_2 t_5)n^2 + t_1 n)) = O(k(n^2)) = O(kn^2) = O(n^2)$$

The time complexities of the proposed methods are $O(n^2)$. Both SSM and SEM can efficiently solve the k-coverage problem without pre-computing the whole area. Both methods are thus suitable for WSNs.

5 Deployment Analysis and Minimization

This section analyzes the correctness and guaranteed k-coverage of the proposed method. The proof is achieved by induction and the minimum value is obtained by differentiation.

5.1 Deployment Analysis

Guaranteeing k-coverage is a decision problem. Therefore, k-coverage must be guaranteed to a given area $m \times n$. This study adopts the concept of layering to obtain k coverage, and also obtains k-coverage by deploying three sensors at each layer for k iterations. Consider an aggregate of effective areas S.

$$S = \bigcup_{j=1}^{k} S_j(X), \text{ where } S_j \text{ denotes an effective area}$$

The area of each layer is A; the effective area is given by $A - B(X)$, and $B(X)$ denotes the total area of obstacles, given by $B(X) = c$, where c denotes a constant. Thus, $S = A - B(X)$ can be set.

Suppose that $f(x, y)$ denotes the area covered by the large sensor; $g(x, y)$ denotes the area which can be covered by the medium sensor, and $h(x, y)$ denotes the area that can be covered by the small sensor. Applying the Cauchy–Schwarz inequality indicates that the layer can be deployed by any of the three sensors, and that k-coverage can be ensured by applying Induction. The Appendix 3 proves the three-tier Cauchy–schwarz inequality.

<proof>: Given a finite set S, and

$$\overset{k}{\underset{j=1}{U}}{}^{j} S \geq \overset{k}{\underset{j=1}{U}} A - B(x) \tag{5}$$

$$\begin{aligned} &\overset{k}{\underset{j=1}{U}}{}^{j}\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right) \\ &\quad \geq \overset{k}{\underset{j=1}{U}}{}^{j}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 - jc, k \geqq j, \{j,k\} \in \mathrm{N} \end{aligned} \tag{6}$$

where j denotes a counter to count the number of Layers, and k denotes the maximum E_{LA}.

(1) To prove that the layer-based method of guaranteeing k-coverage by induction is correct when $k = 1$, Eq. 6 can be rearranged as:

$$\begin{aligned} &\overset{1}{\underset{j=1}{U}}{}^{1}\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right) \\ &\quad \geq \overset{1}{\underset{j=1}{U}}{}^{1}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 - c \end{aligned} \tag{7}$$

Equation 7 is rearranged as

$$\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx \geq \left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 - c \tag{8}$$

(2) If Eq. 6 is correct when $k = n$, then Eq. 6 can be rearranged as:

$$\begin{aligned} &\overset{n}{\underset{j=1}{U}}{}^{j}\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right) \\ &\quad \geq \overset{n}{\underset{j=1}{U}}{}^{j}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 - nc \end{aligned} \tag{9}$$

Equation 8 can be expanded as:

$$\begin{aligned} &{}^{1}\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right) \\ &\cup^{2}\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right)\cdots \\ &\cup^{n}\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right) \\ &\geq{}^{1}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 \cup^{2}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2\cdots \\ &\cup^{n}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 - nc \end{aligned} \tag{10}$$

Equation 10 is simplified as

$$n\left(\int f^2(x,y)dx \cdot \int g^2(x,y)dx \cdot \int h^2(x,y)dx\right) \geq n\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2 - nc \tag{11}$$

(3) If $k = n + 1$ is put into Eq. 6, then:

$$\begin{aligned}
&\overset{n+1}{\underset{j=1}{U}}{}^{j}\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \\
&\quad \geq \overset{n+1}{\underset{j=1}{U}}{}^{j}\left(\int f(x, y)g(x, y)h(x, y)dx\right)^2 - (n+1)c \\
&\quad (n+1)\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \\
&\quad = n\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \\
&\quad +1\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) - (n+1)c
\end{aligned} \tag{12}$$

Put (8) and (11) into (12):

$$\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx \geq \left(\int f(x, y)g(x, y)h(x, y)dx\right)^2 - c \, (8)$$

$$\begin{aligned}
&n\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \\
&\quad \geq n\left(\int f(x, y)g(x, y)h(x, y)dx\right)^2 - nc
\end{aligned} \tag{11}$$

Then obtain (13)

$$\begin{aligned}
&(n+1)\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \\
&\quad \geq n\left(\int f(x, y)g(x, y)h(x, y)dx\right)^2 - nc + \left(\int f(x, y)g(x, y)h(x, y)dx\right)^2 - c \\
&(n+1)\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \\
&\quad \geq (n+1)\left(\int f(x, y)g(x, y)h(x, y)dx\right)^2 - (n+1)c
\end{aligned} \tag{13}$$

The final formula (13) proves that all unknown area with obstacles can be deployed by the proposed method. Thus, the proposed method can achieve k-coverage scenario.

5.2 Minimum Value of Deployment

This section demonstrates that the minimum value can be calculated by the differential coefficient. The basic method is to take the logarithm of Eq. 6, then calculate the minimum value of Eq. 6 with the differential. The procedure is as follows:

(1) Logarithm of Eq. (6)

$$\log\left[\overset{k}{\underset{j=1}{U}}{}^{j}\left(\int f^2(x,y)dx\cdot\int g^2(x,y)dx\cdot\int h^2(x,y)dx\right)\right]$$
$$\geq \log\left[\overset{k}{\underset{j=1}{U}}{}^{j}\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2-jc\right]$$
$$\overset{k}{\underset{j=1}{U}}{}^{j}\left(\log\int f^2(x,y)dx+\log\int g^2(x,y)dx+\log\int h^2(x,y)dx\right)$$
$$\geq \overset{k}{\underset{j=1}{U}}\log^j\left[\left(\int f(x,y)g(x,y)h(x,y)dx\right)^2-jc\right] \tag{14}$$

(2) Rearrange Eq. 14, and calculate the minimum value by the differential rule

$$\overset{kj}{\underset{j=1}{U}}\left(\log\int f^2(x,y)dx+\log\int g^2(x,y)dx+\log\int h^2(x,y)dx\right)$$
$$\geq \overset{k}{\underset{j=1}{U}}\log^j\left[\log\left(A-B(x)\right]\underset{\Rightarrow}{f(x)'}\ \overset{kj}{\underset{j=1}{U}}\left(\frac{2f^2(x,y)}{\int f^2(x,y)dx}\times\frac{df(x,y)}{dx}+\frac{2g^2(x,y)}{\int g^2(x,y)dx}\right.\right.$$
$$\left.\times\frac{dg(x,y)}{dx}+\frac{2h^2(x,y)}{\int h^2(x,y)dx}\times\frac{dh(x,y)}{dx}\right)\geq \overset{k}{\underset{j=1}{U}}\frac{\frac{dA}{dx}\times\frac{dB(x)}{dx}}{A-B(x)} \tag{15}$$

Equation 15 proves that the layering deployment method can effectively guarantee *k*-coverage, and can calculate the minimum value of the equation. Deployment can then be performed in accordance with the $A - B(X)$ area. The minimum value of the equation can then be calculated by deploying the three different sensors. A user can use the minimum value to deploy a sensor network with k-covered scenario, and to evaluate the number of different type sensors.

6 Cost-effectiveness Simulation and Analysis

This section compares the cost-effectiveness of SSM, SEM, random deployment, and VFA. The analysis and test are divided into four parts: (1) Effective deployment ratio (EDR), (2) Deployment rate (DR), (3) Deployment difficulty ratio (DDR) and (4) number of sensors. The test results of various items are explained below:

6.1 Effective Deployment Ratio, EDR

The deployment ratio in the effective area was tested by the three methods. The equation for EDR is defined in Eq. 16

$$EDR=\frac{Effective\ deployment\ area}{All\ area}\times 100\% \tag{16}$$

The EDR was adopted to calculate the percentage of the deployment area that was effective when deploying the same number of sensors. The total area was set to 100 × 100, and the number of obstacles was set to 100. This experiment considered the relationship between EDR and the number of layers. Figure 11 shows the experimental results, which indicate

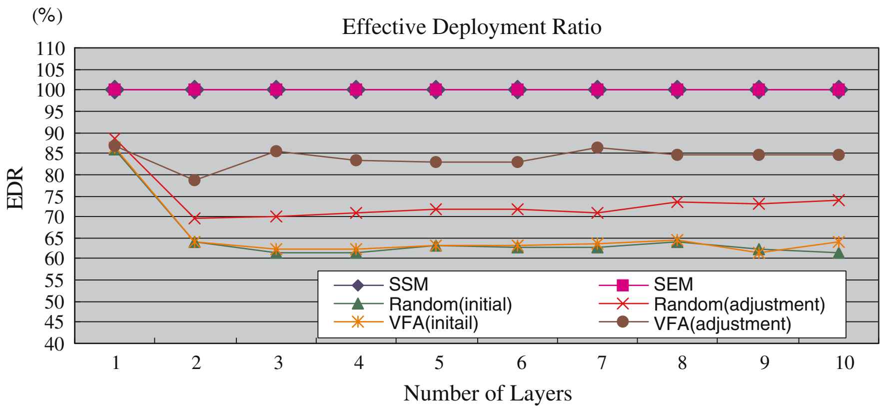

Fig. 11 10-Covered EDR test

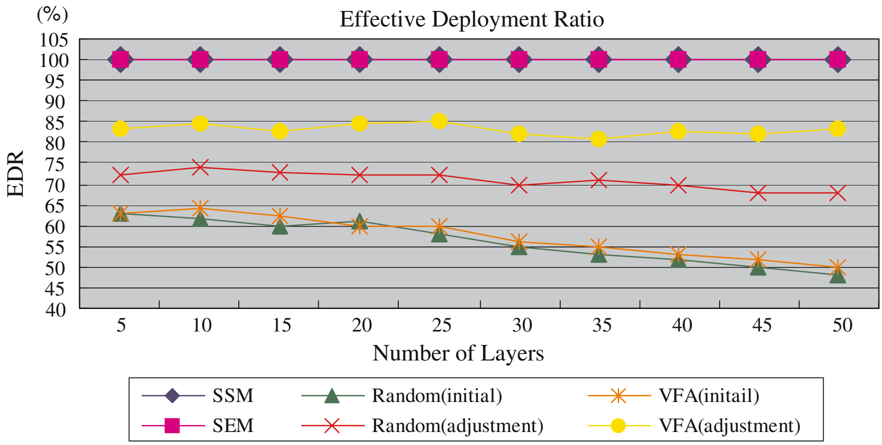

Fig. 12 50-Covered EDR test

the testing of 10-layer EDR. As indicated in Fig. 11, the proposed SSM and SEM deployment schemes were more effective than the traditional random deployment method and VFA. Hence, 10-coverage was guaranteed. Both random deployment and VFA were required to adjust some sensors. The initial effective coverage rate was only 61.2% before adjustment, and rose to 74% after adjustment of random deployment. The initial effective coverage rate was only 64% before adjustment, and rose to 84% after adjustment of VFA. Figure 12 shows the test result of 50 layers. The initial effective coverage rate with random deployment was 48%, rising to 68.58% after adjustment. The initial effective coverage rate with VFA was 50%, and rose to 83.54% after adjustment. Hence, the effective coverage after adjustment was 32.42% and 17.46% less than the EDR of SSM and SEM, respectively, revealing that the random deployment method produced overlapping detection rates of 32.42% and 17.46%.

6.2 Deployment Rate, DR

Deployment rate is mainly adopted to test the layer deployment rate. All methods require guaranteed coverage. Thus, post-test adjustment should be performed by the random deployment method, as in Eq. 17.

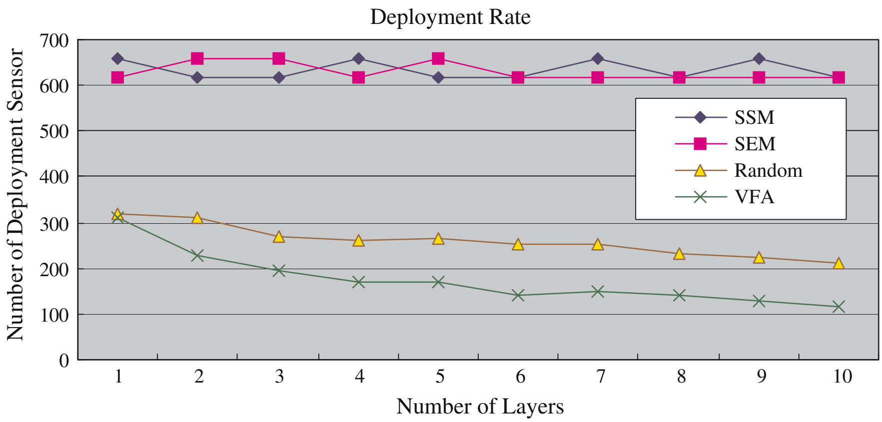

Fig. 13 DR test

$$Deployment\ rate = \frac{Effective\ deployment\ area}{Time} \tag{17}$$

The total area was set to 100 × 100, and the number of obstacles is set to 100. The relationship between the number of obstacles and deployment rate was tested with these values. Figure 13 shows the DR test for 10-covered deployment. SSM and SEM had an approximate deployment rate of 650 sensors per second. However, the deployment rates of random deployment and VFA fall with increasing number of layers. The deployment rate (DR) indicates that 213 and 115 sensors were deployed per second at the 10th layer. SSM and SEM are stable methods, and are thus suitable for deploying WSNs.

6.3 Deployment Difficulty Ratio, DDR

The DDR is mainly adopted to calculate the relationship between the numbers of obstacles and deployment rate. The effectiveness and response of three different methods were tested with rising numbers of obstacles. The DDR equation is defined as Eq. 18

$$\text{Deployment difficulty ratio} = \frac{\dfrac{\text{Area} - \text{obstacles}}{\text{Deployment rate}} - \text{fiducial value}}{\text{Fiducial value}} \times 100\% \tag{18}$$

The relationship between the number of obstacles and deployment rate was tested under 10 layers. The area of each layer was 100 × 100. The fiducial value was set by SEM, when 50 obstacles were solved. Figure 14 indicates that the random deployment method and VFA produced a higher DDR than SEM and SSM. The DDR values of VFA rose when the number of obstacle rose to 100. The DDR vales were 221.61% and 211.56% of the random deployment method and VFA, respectively. The difference of VFA and SSM was about 521% when the number of obstacles rose to 250. Figure 14 indicates that both SSM and SEM could guarantee stable *k*-covered deployment. In particular, SEM by circling can secure the specific locations of sensors. However, the random deployment method and VFA became increasingly unstable as the number of obstacles rose.

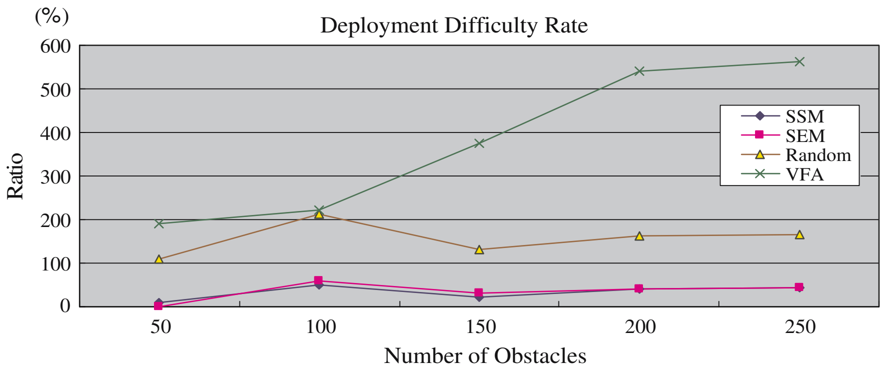

Fig. 14 DDR test

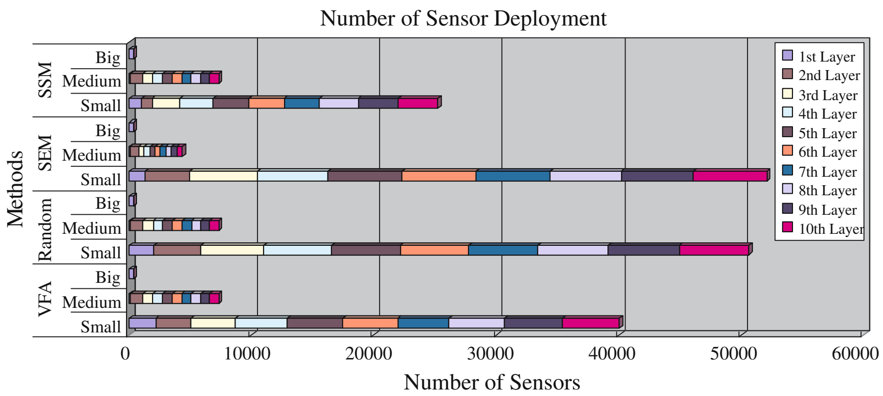

Fig. 15 Number of sensors used

6.4 Number of Deployed Sensors

This section discusses the number of sensors with four different deployment methods, as shown in Fig. 15 and Table 6. The X-axis in Fig. 15 denotes the number of sensors, and the Y-axis denotes the types of sensors adopted with four deployment methods. The general sum of 10-covered deployment sensors was also analyzed. All methods were run with 330 large sensors.

SSM mostly adopts medium sensors, as indicated in Fig. 15. Thus, data transmission can be performed without small sensors, thus reducing the battery cost and increasing the network survival rate. The deployment cost rises significantly if each layer has many small sensors when adopting the random deployment method.

Additionally, the number of overlapping areas reached 12,027 when the 5th layer was deployed by the random deployment method. The number of overlapping areas rose to 24,590 using random deployment, and 13,870 using VFA in 10th layer deployment. Hence, the overlap rate was high in random deployment and VFA.

Table 6 Number of sensors used

Number of layers	SSM		SEM		Random		Overlapping area	VFA		Overlapping area
	Medium	Small	Medium	Small	Medium	Small	Total	Medium	Small	Total
1	77	957	43	1,263	77	2,025	968	77	2,159	1,102
2	1,002	882	696	3,636	1,002	3,774	3,760	1,002	2,867	2,987
3	846	2,286	482	5,562	864	5,172	6,708	846	3,632	4,233
4	799	2,709	456	5,796	799	5,527	9,426	799	4,272	5,696
5	774	2,934	428	6,048	774	5,635	12,027	774	4,525	7,187
6	779	2,889	428	6,048	779	5,593	14,631	779	4,475	8,673
7	778	2,898	432	6,012	778	5,689	17,322	778	4,178	9,853
8	752	3,132	454	5,814	752	5,688	19,778	752	4,545	11,166
9	733	3,303	439	5,949	733	5,819	22,194	733	4,755	12,518
10	745	3,195	437	5,967	745	5,691	24,590	745	4,647	13,870

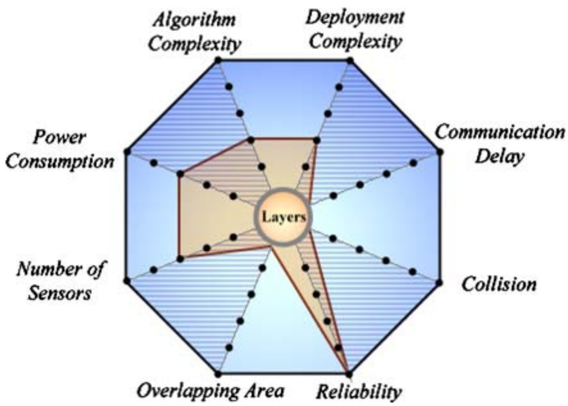

Fig. 16 The deployment costs analysis of SSM

6.5 Cost Analysis

CPE factors were analyzed in SSM, random deployment and VFA. Figures 16–18 show the total deployment costs of SSM, random deployment, and VFA. The figures indicate that all of these methods are with the same reliability, i.e., k coverage. Other CPE factors are discussed as follows. Since the deployment problem does not need to consider the communication delay (E_{CD}) and Collision (E_{co}), the values of E_{CD} and E_{CO} are set to zero. Table 7 compares the performance of the proposed methods.

E_{DC} is calculated from the complexity of the deployment algorithm. Random deployment is easy to implement. The time complexity of random deployment is $O(n)$. The E_{AC} of SSM, random deployment and VFA are $O(n^2)$, $O(n)$ and $O(n^3)$, but the k-coverage scenario, E_{DC}, of SSM is also $O(n^2)$. The E_{DC} of VFA is higher than that of SSM. The E_{DC} proportions are as follows:

$$\text{SSM: Random deployment: VFA} = 2 : 3 : 4. \tag{19}$$

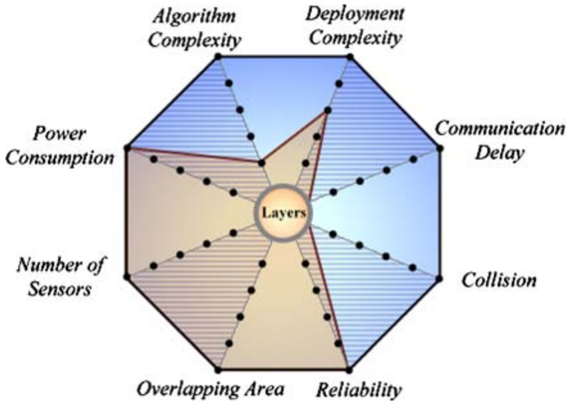

Fig. 17 The deployment costs analysis of random deployment

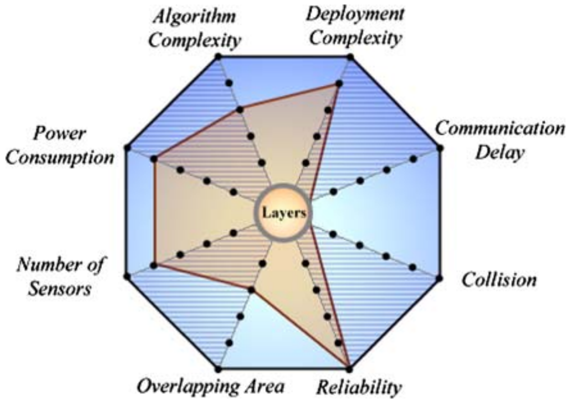

Fig. 18 The deployment costs analysis of VFA

Table 7 Comparison related algorithms

Effective factors	SSM/SEM	Random deployment	VFA
Algorithm complexity (E_{AC})	$O(n^2)$	$O(n)$	$O(n^3)$
Deployment complexity (E_{DC})	$O(n^2)$	$O(n^3)$	$O(n^4)$
Reliability (E_{RE})	High (k coverage)	High (k coverage)	High (k coverage)
Overlapping area (E_{OA})	Low	High	Medium
Number of sensors (E_{SE})	Low	High	Medium
Power consumptions (E_{PC})	Medium	High	Medium
Number of layers (E_{LA})	k layers	k layers	k layers

Although random deployment has a lower algorithm complexity (E_{AC}), some CPE factors are higher than others. Evaluation results indicate that the effective coverage after adjustment was 13.46% lower than the effective deployment ratio (EDR) of SSM. Therefore, about 13.46% detection was overlapping detection with VFA. The E_{OA} factor in random distribution was worse than those of SSM and VFA. The 26% area covered and overlapped at least one sensor. The E_{OA} proportions are as follows:

$$\text{SSM: Random deployment: VFA} = 0 : 5 : 2. \tag{20}$$

Test results of deployment rate (DR) indicate that SSM and SEM could deploy 650 sensors per second. The random deployment deployed 213 sensors per second when the 10th layer was deployed. The VFA only deployed 115 sensors per second when the 10th layer was deployed. For the testing of DDR for k-covered deployment, random deployment was about 122.46% higher than SSM, and VFA was about 563.33% higher than SSM. Hence, the deployment complexities (E_{DC}) of random deployment and VFA were higher than that of SSM.

Additionally, the number of overlapping areas reached 12,027 when the 5th layer was deployed in the random deployment method, and rose to 24,590 in deployment of the 10th layer. VFA had fewer overlapping areas than random deployment. The number of overlapping areas reached 7,187 when the 5th layer was deployed, and rose to 13,870 in deployment of the 10th layer. These findings indicate that the number of sensors (E_{NS}) was higher in random deployment and VFA than in the proposed methods. Figure 14 shows the total costs using the proposed SSM, SEM, random deployment and VFA control schemes. As indicated in Fig. 14, the total deployment costs of the random scheme were higher than those of the proposed SSM/SEM scheme. The E_{NS} proportion is as follows:

$$\text{SSM: Random deployment: VFA} = 3 : 5 : 4. \tag{21}$$

The results indicate that SSM has the following advantages: (1) it can deploy a k-coverage scenario easily; (2) it can deploy a hostile environment; (3) it requires fewer sensors than VFA and random deployment; (4) it has a small overlapping area, and (5) it has a high scalability of heterogeneous sensors.

7 Application

The k-coverage is a basic and important issue for prevention of nature disasters, especially on land-slope disasters. Many feasible mechanisms and systems have been presented in the past several years to predict the occurrence of land-slope disasters. Sensors can be deployed on the dangerous area. However, 1-coverage scenario is not enough. A WSN is a small network, in which all sensors have limited power supply, limited capabilities and sensing small area. These sensors can easily miss detected data, and run out of power, causing unsafe situations. The minimum acceptable coverage is 2-coverage. One layer is basic, and the other is duplicated for data detection and collection. The accurate probability of occurrence of land-slope disasters is based on the collected data and effective prediction models. Therefore, the accuracy of collected data is a very important issue. The value of k influences data accuracy. Figure 19 shows a relief map for land-slope disasters. The region needs k-coverage sensing. Figure 20 shows a k-coverage scenario based on an application of land-slope disaster.

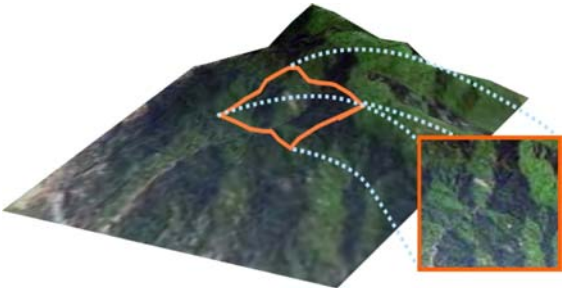

Fig. 19 A relief map for land-slope disasters

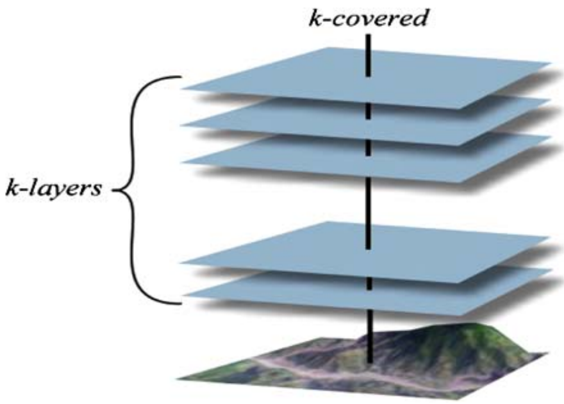

Fig. 20 A k coverage application based on land-slope disaster

8 Conclusion

This study presents two novel topology deployment methods, and performs a comprehensive performance analysis and evaluation of the cost-effectiveness of setting up wireless sensor networks. The proposed slow-start method and square-encircled method are adopted to perform deployment analysis for deploying an unknown obstacle-filled area while reducing the overlap rate. SSM and SEM simplify the originally complex problem of guaranteeing k-coverage by employing the concept of divide and conquer. SSM can be deployed quickly the first time, and SEM can solve the problem of deploying unknown areas. The correctness of both methods is mathematically proven. Moreover, this study first defines nine CPE factors in evaluating the performance of WSN's topology deployment. Finally, as indicated in the cost-effectiveness simulation results, the deployment effectiveness rate, difficulty rate and number of sensors are more favorable when using SSM and SEM than the traditional random deployment method and VFA.

Acknowledgments The authors would like to thank the National Science Council of the Republic of China, Taiwan for financially supporting this research under Contract No. NSC 96-2221-E-020-034-MY2 supports the research.

Appendix 1

Appendix 1 lists all the parameters into Table 8.

Table 8 The list of all parameters

Parameters	Meaning
E_{AC}	Algorithm complexity.
E_{DC}	Deployment complexity.
E_{CD}	Communication delay.
E_{CO}	Collision.
E_{RE}	Reliability.
E_{OA}	Overlapping area.
E_{SE}	Number of sensors.
E_{PC}	Power consumptions.
E_{LA}	Number of layers.
$weight_{ij}$	The related CPE factors weight between factor i and j.
V_i	The periphery performance factor i.
W_i	The weighting value of performance factor i.
G_i	The neighboring performance evaluation factor of the shadowed triangle Vi.
S_L,	Sets of large sensors.
S_M,	Sets of medium sensors.
S_s	Sets of small sensors.
k	k layers.
X	The length of an area.
Y	The width of an area.
$P(x'_o, y'_o)$	Start point.
$P(X, Y)$	End point.
$P(x'_i, y'_j)$	Temporary end point.
$P(x_c, y_c)$	Check point.
Layer	Number of layers.
S	Total effective area.
$S_j(x)$	The effective area of j^{th} layer.
t_i	t_i is the execution time for evaluate algorithm.
n	The input data number.
A	The area of each layer.
$B(X)$	The areas of obstacles.
$f(x, y)$	The covered area by the large sensor.
$g(x, y)$	The covered area by the medium sensor.
$h(x, y)$	The covered area by the smaill sensor.
U	U is union.
EDR	Effective deployment ratio.
DR	Deployment ratio.
DDR	Deployment difficulty ratio.

Appendix 2

Appendix 2 is the analysis the time complexities of proposed algorithms, SSM and SEM, in Tables 9 and 10.

Table 9 The analysis of SSM algorithm

Slow-start method	Time	Times
Parameters notification		
The size of total area is $x \times y$		
x is the length of area		
y is the width of area		
$p_o(0,0)$ is initial point		
$P(x'_o, y'_o)$ is start point		
$p(x, y)$ is end point		
$P(x'_t, y'_t)$ is temporary end point		
Layer is number of layers		
a, temp are counters		
S_L, S_M, S_S are Large sensors, medium sensors, and small sensors		
Define-effective-area$(P(x'_o, y'_o), P(x'_t, y'_t))$		
loop (detect from $P(x'_o, y'_o)$ to $P(x'_t, y'_t)$)	t_1	n
if (effect area is with obstacles) **then**		
$(x'_t, y'_t) = \left(\left\lceil \frac{x'_t}{2} \right\rceil, \left\lceil \frac{y'_t}{2} \right\rceil\right)$		
End if		
End loop		
loop (the increased area without obstacles)	t_2	n
$(x'_t, y'_t) = (x'_t + 1, y'_t + 1)$		
End loop		
Return (x'_t, y'_t)		
Slow-start method (*from Layer 1 to Layer k*)		
if (*Layer* = *1*) **then**		
set start point $P(x'_o, y'_o) = P_0(0,0)$		
loop (!sockets are full of S_L sensors or S_L sensors use up)	t_3	n
set S_L to all sockets		
Mark all covered range of S_L		
End loop		
End if		
Set start point $P(x'_o, y'_o) = P_0\left(\left\lfloor \frac{Layer}{x} \right\rfloor, mod\left(\frac{Layer}{y}\right)\right)$		
Put S_s sensors from $P_o(0,0)$ to $P(x'_o, y'_o) - 1$		
Set temporary end point $P(x'_t, y'_t) = P\left(\left\lceil \frac{x}{2} \right\rceil, \left\lceil \frac{y}{2} \right\rceil\right)$		
loop (from start point $P(x'_o, y'_o)$ to end point $P(x, y)$)	t_4	n
Define-effective-area$(P(x'_o, y'_o), P(x'_t, y'_t))$		
// set sensors into effective area		
loop (from start point $P(x'_o, y'_o)$ to end point $P(x'_t, y'_t)$)	t_5	n
set S_s or S_M into the effective_area		
Mark effective_area		
End loop		
// to count next start point and end point		
loop (a is from 0 to x)	t_6	n
$P(x'_o, y'_o) = P(a, (y_t + 1) mod y)$		
if ($P(x'_o, y'_o)$ is marked space and $(y'_o + 1)$ is not equal to y) **then**		
$P(x'_o, y'_o) = P(x'_o, (y'_o + 1) mod y)$		
else		
$P(x'_o, y'_o) = P((x'_o + 1) mod\ x, (y'_o + 1) mod\ y)$		
End if		

Table 9 continued

Slow-start method	Time	Times
End loop		
temp=minimum($(x - x'_t), (y - y'_t)$)		
$P(x'_t, y'_t) = (x'_t + temp, y'_t + temp)$		
End loop		
Layer=*Layer*+1		

Table 10 The analysis of SEM algorithm

Square-encircled method	Time	Times
Parameters notification		
The size of total area is $x \times y$		
x is the length of area		
y is the width of area		
$P_o(0, 0)$ is initial point		
$P(x'_o, y'_o)$ is start point		
$P(x, y)$ is end point		
$P(x_c, y_c)$ is check point		
Layer is number of layers		
a, b is a counter		
S_L, S_M, S_S are Large sensors, medium sensors, and small sensors		
Square-encircled method (*from Layer 1 to Layer k*)		
if (*Layer=1*) **then**		
set start point $P(x'_o, y'_o) = P_0(0, 0)$		
loop (!sockets are full of S_L sensors or S_L sensors use up)	t_1	n
set S_L to all sockets		
Mark all covered range of S_L		
End loop		
End if		
Set start point $P(x'_o, y'_o) = P\left(\left\lfloor \frac{Layer}{x} \right\rfloor, mod\left(\frac{Layer}{y}\right)\right)$		
Put S_s sensors from $P_o(0, 0)$ to $P(x'_o, y'_o) - 1$		
Set check point $P(x_c, y_c)$		
loop (from start point $P(X'_o, Y'_o)$ to end point $P(x, y)$)	t_2	n
loop ((encircled area is satisfied with S_M or S_s)	t_3	n
if (!obstacles)**then**		
Push effective_area_stack P($x_c, y_c + 1$)		
Push effective_area_stack P($(1 + 2b) + x_c, y_c - (1 + 2b)$)		
Push effective_area_stack P($x_c + 1, y_c$)		
Push effective_area_stack P($(2 + 2b) + y_c, x_c - (2 + 2b)$)		
Else		
Pop (from top to 1)		
End if		
End loop		
loop(from top to empty)	t_4	n
Mark effective_area_stack()		
Pop effective_area_stack()		
End loop		
loop (a is from 0 to x)	t_5	n

Table 10 continued

Square-encircled method	Time	Times
$P(x'_o, y'_o) = P(a, (y_t + 1) mod\ y)$		
if ($P(x'_o, y'_o)$ is marked space and $(y'_o + 1)$ is not equal to y) **then**		
$P(x'_o, y'_o) = P(x'_o, (y'_o + 1) mod\ y)$		
else		
$P(x'_o, y'_o) = P((x'_o + 1) mod\ x, (y'_o + 1) mod\ y)$		
End if		
End loop		
End loop		
Layer=*Layer*+1		

Appendix 3

Appendix 3 proves the three-tier Cauchy–Schwarz inequality. Most of people know and are used to use the general formula of Cauchy–Schwarz inequality which is $(a^2+b^2)(c^2+d^2) \geqq (ac+bd)^2$, where a, b, c, d are any number. In this study, the formula of Cauchy–Schwarz inequality is used by integral type as $(\int f(x)g(x)h(x)dx)^2 \leq \int f^2(x)\,dx \int g^2(x)dx \int h^2(x)dx$. In this appendix, we will introduce the processes of proving this formula.

Let λ is a polynomial of real number

f(x) and g(x) are the functions of x

$$\int(\lambda f(x)+g(x))^2 dx \geq 0 \tag{22}$$

$$\int(\lambda^2 f^2(x) + 2\lambda f(x)g(x) + g^2(x))dx \geq 0 \tag{23}$$

$$(\int f^2(x)dx)\lambda^2 + 2(\int f(x)g(x)dx)\lambda + \int g^2(x)dx \geq 0 \tag{24}$$

Discriminant $\triangle \leq= 0$

$$(2\int f(x)g(x)dx)^2 - 4\int f^2(x)dx \int g^2(x)dx \leq 0 \tag{25}$$

$$(\int f(x)g(x)dx)^2 \leq \int f^2(x)dx \int g^2(x)dx \tag{26}$$

Three-tier Cauchy–Schwarz inequality

let $H(x)=g(x)h(x)$

$$\int(\lambda f(x)+H(x))^2 dx \geq 0 \tag{27}$$

As the steps from formula (22)–(26)

$$(\int f(x)H(x)dx)^2 \leq \int f^2(x)dx \int H^2(x)dx \tag{28}$$

$$\int H^2(x)dx = \int(g(x)h(x))^2 dx \leq \int g^2(x)dx \int h^2(x)dx \tag{29}$$

To substitute to formula (28) and get

$$(\int f(x)g(x)h(x)dx)^2 \leq \int f^2(x)dx \int g^2(x)dx \int h^2(x)dx$$

Q.E.D

Suppose that $f(x, y)$ denotes the area covered by the large sensor; $g(x, y)$ denotes the area which can be covered by the medium sensor, and $h(x, y)$ denotes the area that can be covered by the small sensor. We can get the following formula.

$$\left(\int f^2(x, y)dx \cdot \int g^2(x, y)dx \cdot \int h^2(x, y)dx\right) \geq \left(\int f(x, y)g(x, y)h(x, y)dx\right)^2$$

The formula is set up. Q.E.D.

References

1. AboElFotoh, H. M. F., Iyengar, S. S., & Chakrabarty, K. (2005). Computing reliability and message delay for cooperative wireless distributed sensor networks subject to random failures. *IEEE Transactions on Reliability, 54*(1), 145–155.
2. Abrams, Z., Goel, A., & Plotkin, S. (2004). Set k-cover algorithms for energy efficient monitoring in wireless sensor networks. In *Proceedings of Third Int. Symposium on Information Processing in Sensor Networks (IPSN 2004)*, pp. 424–432.
3. Bejerano, Y. (2004). Efficient integration of multihop wireless and wired networks with QoS constraints. *IEEE/ACM Transactions on Networking, 12*(6), 1064–1078.
4. Beutel, J., Dyer, M., Meier, L., & Thiele, L. (2005). Scalable topology control for deployment-support networks. In *Proceedings of Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 359–363.
5. Bulusu, N., Heidemann, J., Estrin, D., & Tran, T. (2004). Self-configuring localization systems: design and experimental evaluation. *ACM Transactions on Embedded Computing Systems (TECS), 3*(1), 24–60.
6. Cărbunar, B., Grama, A., Vitek, J., & Cărbunar, O. (2006). Redundancy and coverage detection in sensor networks. *ACM Transactions on Sensor Networks, 2*(1), 94–128.
7. Cerpa, A., & Estrin, D. (2004). ASCENT: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing, 3*(3), 272–285.
8. Felemban, E., Lee, C. G., & Ekici, E. (2006). MMSPEED: Multipath Multi-SPEED protocol for QoS guarantee of reliability and timeliness in wireless sensor networks. *IEEE Transactions on Mobile Computing, 5*(6), 738–754.
9. Gupta, H., Zhou, Z. H., Das, S. R., & Gu, Q. (2006). Connected sensor cover: Self-organization of sensor networks for efficient query execution. *IEEE/ACM Transactions on Networking, 14* (1), 55–67.
10. Hou, Y. T., Shi, Y., Sherali, H. D., & Midkiff, S. F. (2005). On energy provisioning and relay node placement for wireless sensor networks. *IEEE Transactions on Wireless Communications, 4* (5), 2579–2590.
11. Huang, C. F., & Tseng, Y. C. (2003). Routing, coverage, and topology control: The coverage problem in a wireless sensor network. In *Proceedings of 2nd ACM International Conference on Wireless Sensor Networks and Applications*, pp. 115–121.
12. Huang, C. F., & Tseng, Y. C. (2005). A survey of solutions to the coverage problems in wireless sensor networks. *Journal of Internet Technology, 6*(1), 1–8.
13. Kawahara, Y., Kawanishi, N., Morikawa, H., & Aoyama, T. (2004). Top-down approach toward building ubiquitous sensor network applications. In *Proceedings of 11th Asia-Pacific Software Engineering Conference*, pp. 695–702.
14. Kumar, S., Lai, T. H., & Balogh, J. (2004). On k-coverage in a mostly sleeping sensor network. In *Proceedings of 10th Annual International Conference on Mobile Computing and Networking (MobiCom 2004)*, pp. 144–158.
15. Lee, J. J., Krishnamachari, B., & Kuo, C. C. J. (2004). Impact of heterogeneous deployment on lifetime sensing coverage in sensor networks. In *Proceedings of First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004)*, pp. 367–376.
16. Li, W., & Cassandras, C. G. (2005). A minimum-power wireless sensor network self-deployment scheme. In *Proceedings of IEEE Wireless Communications and Networking Conference, 3*, 1897–1902.
17. Lin, C. Y., Peng, W. C., & Tseng, Y. C. (2006). Efficient in-network moving object tracking in wireless sensor networks. *IEEE Transactions on Mobile Computing, 5*(8), 1044–1056.
18. Ma, Y., & Aylor, J. H. (2004). System lifetime optimization for heterogeneous sensor networks with a hub-spoke technology. *IEEE Transactions on Mobile Computing, 3*(3), 286–294.
19. Megerian, S., Koushanfar, F., Potkonjak, M., & Srivastava, M. B. (2005). Worst and best-case coverage in sensor networks. *IEEE Transactions on Mobile Computing, 4*(1), 84–92.

20. Mhatre, V. P., Rosenberg, C., Kofman, D., Mazumdar, R., & Shroff, N. (2005). A minimum cost heterogeneous sensor network with a lifetime constraint. *IEEE Transactions on Mobile Computing, 4*(1), 4–15.
21. Passos, R. M., Coelho, C. J. N., Loureiro, A. A. F., & Mini, R. A. F. (2005). Dynamic power management in wireless sensor networks: An application-driven approach. In *Proceedings of Second Annual Conference on Wireless On-demand Network Systems and Services (2005 WONS)*, pp. 109–118.
22. Rachlin, Y., Negi, R., & Khosla, P. (2005). Sensing capacity for discrete sensor network applications. In *Proceedings of Fourth International Symposium on Information Processing in Sensor Networks (IPSN 2005)*, pp. 126–132.
23. Subramanian, R., & Fekri, F. (2006). Sleep scheduling and lifetime maximization in sensor networks: Fundamental limits and optimal solutions. In *Proceedings of the Fifth International Conference on Information Processing in Sensor Networks (IPSN 2006)*, pp. 218–225.
24. Wang, G., Cao, G., & Porta, T. L. (2004). Proxy-based sensor deployment for mobile sensor networks. In *Proceedings of 2004 IEEE International Conference on Mobile Ad-hoc and Sensor Systems*, pp. 493–502.
25. Wang, Y. C., Hu, C. C., & Tseng, Y. C. (2005). Efficient deployment algorithms for ensuring coverage and connectivity of wireless sensor networks. In *Proceedings of First International Conference on Wireless Internet*, pp. 114–121.
26. Werner-Allen, G., Lorincz, K., Ruiz, M., Marcillo, O., Johnson, J., Lees, J., & Welsh, M. (2006). Deploying a wireless sensor network on an active volcano. *IEEE Internet Computing, 10*(2), 18–25.
27. Wu, J., & Yang, S. (2005). SMART: A scan-based movement-assisted sensor deployment method in wireless sensor networks. In *Proceedings of IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005), 4*, 2313–2324.
28. Xing, L., & Shrestha, A. (2006). QoS reliability of hierarchical clustered wireless sensor networks. In *Proceedings of 25th IEEE International on Performance, Computing, and Communications Conference (IPCCC 2006)*, pp. 641–646.
29. Zhou, Z., Das, S., & Gupta, H. (2004). Variable radii connected sensor cover in sensor networks. In *Proceedings of 2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks (IEEE SECON 2004)*, pp. 387–396.
30. Zou, Y., & Chakrabarty, K. (2004). Sensor deployment and target localization in distributed sensor networks. *ACM Transactions on Embedded Computing Systems, 3*(1), 61–91.

Author Biographies

Hsu-Yang Kung received his PhD Degree in Computer Science and Information Engineering from National Cheng-Kung University, Taiwan. He is currently a Professor and Chairman of the Department of Management Information Systems, National Pingtung University of Science and Technology, Taiwan. His research interests include distributed multimedia systems, wireless and mobile communications, and embedded multimedia applications.

Chung-Ming Huang received the B.S. Degree in Electrical Engineering from National Taiwan University, and the M.S. and Ph.D. Degrees in Computer and Information Science from The Ohio State University. He is a distinguished professor in Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, R.O.C. Currently, he is Chairman of Dept. of Computer Science and Information Engineering, National Cheng Kung University, Taiwan, R.O.C. and the director of The Promotion Center for Network Applications and Services Education, National Innovative Communication Education Program, Ministry of Education, Taiwan, R.O.C. He has published more than 100 referred journal and conference papers in wireless and mobile interactive multimedia systems, audio and video streaming, and formal modeling of communication protocols. His research interests include media streaming protocols and applications, wireless and mobile interactive multimedia systems, wireless and mobile communication protocols and software, and broadband/mobile Internet applications and service systems.

Hao-Hsaing Ku received the B.S. Degree in the Department of Management Information Systems from Chung-Hua University on 2001/6, and the M.S. degree in the Department of Management Information Systems from National Pingtung University of Science and Technology on 2003/6, Taiwan, R.O.C. He is currently working for his Ph.D. Degree in the Department of Computer Science and Information Engineering, National Cheng Kung University. His research interests are sensor networks, pervasive computing and embedded multimedia applications.