

Article

A Method to Construct an Indoor Air Pollution Monitoring System Based on a Wireless Sensor Network

Ahra Nazih¹, Li Fu¹, Galloua Mohamed Elamine² and Lingling Wang^{1,*}

¹ School of Automation Science and Electrical Engineering, Beihang University, Beijing 100191, China; naz-aero@hotmail.com (A.N.); fuli@buaa.edu.cn (L.F.)

² School of Computer Science and Engineering, Beihang University, Beijing 100191, China; galloua.ma@hotmail.fr

* Correspondence: wangling0908@buaa.edu.cn; Tel.: +86-10-8231-7306

Received: 14 January 2019; Accepted: 17 February 2019; Published: 25 February 2019



Abstract: Wireless Sensors Networks (WSNs) are currently receiving much research interest due to their wide-ranging use in a number of different fields. In the current study, a system based on a WSN is proposed that can monitor indoor air pollution in several public spaces, such as subway stations, offices, schools, and hospitals. The proposed system uses integrated sensors in mobile phones, moving from a stationary nodes model to a mobile nodes model. The main objective of building this system is to provide full coverage of the target area. To achieve this goal, the system is simulated by MATLAB and the following algorithms are applied: Particle Swarm Optimization (PSO) to maximize the coverage in the region of interest (RoI), Voronoi Diagram (VD) to detect holes in the coverage, and finally the Point in Polygon (PiP) algorithm to heal the holes in the coverage. The application of the algorithms mentioned above has been very effective as PSO has increased the coverage rate of the monitoring area to 100%. The VD allowed us to define the exact location of coverage holes while the Point in Polygon algorithm allowed us to heal the holes and find the remaining sensors in order to improve network coverage. This enabled us to achieve full coverage of the monitoring area.

Keywords: wireless sensor network; region of interest; voronoi diagram; point in polygon; coverage; coverage hole

1. Introduction

With the rapid development of communication technology, network and remote sensing technologies can help us to improve air pollution monitoring systems that are usually designed in a wireless mode. Currently, monitoring the physical infrastructure of data centers is very important so as to enable effective maintenance, the prevention of any downtime, accurate event information, as well as efficiently planning appropriate energy requirements. These technologies are usually designed in a wireless mode and create a lot of air pollution. Furthermore, physical electronic devices that are used alongside complex electronic devices create a lot of air pollution within a specific frequency spectrum [1–4]. Wireless air pollution monitoring systems operate in several modes. However, these modes are expensive to install and maintain due to their complexity. Although WSNs are initially used for military and industrial reasons [5], it has been developed for a growing number of industrial applications in recent years.

A study from 2009 [6] presented a network for indoor and outdoor air quality monitoring by installing nodes in different rooms that included tin dioxide sensor arrays connected to an acquisition and control system. The main findings of this paper were: The development of an air quality

monitoring system that uses a smart sensor in a wireless network where the nodes have a specific location, the embedding of neural network processing blocks distributing the processing charge between the embedded systems (web sensor) and the web browser installed in a personal computer, the development of PC software for sensing node TCP/IP remote control, advanced data processing, data storage, and web publishing software associated with air quality monitoring systems using JavaScript and LabView for a web publisher. The work in reference [7] presented a prototype WSN that could monitor the environment in a conference room in a smart building using stationary sending nodes. This prototype network facilitated the testing of four different architectures tasked with learning the relationship between CO₂ and occupant state. Four different learning models were compared: Gradient Boosting (GB), k-Nearest Neighbors (KNN), Linear Discriminant Analysis (LDA), and Random Forests (RF). The KNN model returned the best performance under preferred settings with an accuracy of 47.8% and RMSE (Root Mean Square Error) of 1.021.

The system in reference [8] aimed to detect the level of seven gases in three different settings, ozone (O₃), particulate matter, carbon monoxide (CO), nitrogen oxides (NO₂), sulfur dioxide (SO₂), volatile organic compound, and carbon dioxide (CO₂), on a real-time basis. In doing so, the system provided an overall air quality alert using one of two sensor nodes. In the sensing node, a smoothing algorithm was introduced to prevent temporary sensor errors, and an aggregation algorithm was used to reduce network traffic and power consumption. In the middleware, all data from the sensor cloud were calibrated with the characteristic tables of the cross sensitivities and temperature/humidity dependence. Experiments were conducted to validate and support the development of the system for real-time monitoring and alerts.

The authors in [9] addressed the concept of a real-time WSN capable of monitoring the concentration of indoor carbon dioxide and dangerous situations inside three different areas (two medium size and one by size) using stationary sensor nodes for data aggregation and mobile nodes for data calibration. They followed a cognitive networking technique with an opportunistic routing protocol. An information processing framework was proposed to detect outliers, form data packets, and calibrate the sensors. The system managed to report CO₂ concentration in real-time, while the information processing framework minimized outliers. In [10], indoor air quality was improved by controlling particulate matter (PM) consisting of components such as house dust and bacteria, using an atmospheric pressure plasma. In this study, two types of electrode were used for particle collection and formaldehyde removal. Both electrodes were designed to shorten the gap distance and to minimize the discharge voltage used for indoor air treatment. Formaldehyde control was performed in a 1 m³ chamber. A high gas flow was used with the electrode system to simulate the large flow rate in a room. These electrode systems had lower operating voltages than those reported for conventional ESPs.

In the field of technology, there are many indoor air pollution monitoring systems based on WSN. The network of those systems was built with stationary sensor nodes and developed using different architectures, modules, networking protocols, and expensive elements such as: IoT platform and Libelium collections [11], AVR ATmega64 micro controller, Raspberry, and LoRaWAN wireless communication [12], XBee module and Arduino Mega system [13], and ZigBee networking protocol [14,15], GSM modem [16], and ARIMA prediction models [17]. However, the data collected, and the analyzed results, are not accessible to the general public.

For those major reasons, in this project we will realize a low-cost system to monitor indoor air pollution and switch from stationary nodes in a specific area to mobile nodes in the whole city. We will achieve this by using the mobile phones of the general public to compose a WSN and then transmit the values of the parameters collected by using WIFI or a hot spot to the base station where the level of air pollution will be evaluated and then communicated to the general public. Deploying sensor nodes is a critical phase that dramatically affects the performance of the network and the quality of service. Additionally, the performance of the WSN is highly influenced by the method used to deploy the sensor nodes.

A number of different issues have been raised in regard to the deployment of sensor nodes in a WSN. Several studies based on optimization algorithms have considered different methods to achieve total coverage of the monitoring area such as: Biogeography based optimization and deferential evaluation [18], virtual forces algorithm [19], genetic algorithm with particle swarm optimization [20], particle swarm optimization with Voronoi Diagram (VD) [21], and differentiated deployment algorithm based on image processing and 3D modeling [22]. These studies mainly concern stationary and mobile cases.

The main objective of this project is to ensure the best quality of service, and this begins with full coverage of the monitoring area. The process of deploying the sensors has an impact on the performance of the WSN because it affects the network coverage, communication cost, and management resources, so the deployment strategy of a WSN is a major problem.

After we had conducted our review of the relevant literature, we then began to choose the algorithms that would suit us in our work to allow us overcome the aforementioned problems. Therefore, for the first part of coverage optimization and deployment strategy, the Particle Swarm Optimization (PSO) will be applied to fulfill the coverage over the region of interest (RoI) in Section 2. The second part of our application will be divided into two sections: A VD to detect the location of the coverage holes, and to then use the Point in Polygon (PiP) for coverage hole healing, which will be expanded in Section 3. Finally, we will use MATLAB language to simulate our system in Section 4, and give the conclusion in Section 5.

2. WSN Coverage Optimization Based on PSO

2.1. Network Node Set Coverage

Assume that the target area (Region of Interest) A is a two-dimensional plane, which is digitized into $l \times w$ grids where the size of each grid is equal to 1. The WSN with n total number of sensors is deployed on the target area with two specifications:

- (1). All sensors have the same sensing range r and communication range $R_c = 2r$
- (2). The sensor nodes with the same parameters are placed, and the coordinates of each node are (x_i, y_i) , where $i = 1, 2, \dots, n$.

The monitoring field of the i sensor node is centered on sensor position coordinate (x_i, y_i) and r is the circle of the monitoring radius. The node of each sensor node set is denoted by $C = \{c_1, c_2, \dots, c_n\}$, where $c_i = \{x_i, y_i, r\}$. Assuming that the monitoring area A is digitally discredited into $l \times w$ pixels where the coordinates of each pixel are denoted by (x_j, y_j) and $j = 1, 2, \dots, l \times w$. The Euclidean distance [23] between the j pixel and the i sensor node is:

$$d(c_i, p_i) = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2} \quad (1)$$

The probability of the event which the pixel (x_j, y_j) is covered by the sensor node $c_i = \{x_i, y_i, r\}$ is:

$$P_{cov}(x_j, y_j, c_i) = \begin{cases} 1, & \text{if } d(c_i, p_i) < r \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

Due to the interference of environment, noise, and other factors in sensing model, the sensor nodes should distribute in a certain probability [24],

$$P_{cov}(x_j, y_j, c_i) = \begin{cases} 1, & \text{if } d(c_i, p_i) \leq r - re \\ e^{\frac{-\alpha_1 \lambda_1 \beta_1}{\lambda_2 \beta_2 + \alpha_2}}, & \text{if } r - re < d(c_i, p_i) < r + re \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

where re is the reliability parameter of the measuring of sensor nodes with $re(0 < re < r)$, $\alpha_1, \beta_1, \alpha_2, \beta_2 > 0$ are the measurement parameters about the characteristics of the sensor node, where

$$\lambda_1 = re - r + d(c_i, p_i) \text{ and } \lambda_2 = re + r - d(c_i, p_i) \quad (4)$$

Considering that the target pixel is simultaneously covered by multiple sensor nodes [24], the joint monitoring probability is

$$P_{cov}(Cov_j) = 1 - \prod_{c_i \in Cov_j} (1 - P_{cov}(x_j, y_j, c_i)) \quad i = 1, 2, \dots, n; j = 1, 2, \dots, l \times w \quad (5)$$

where Cov_j is the set of sensor nodes that measure the target pixel j .

2.2. Optimization Model of WSN Coverage

In this work, we use the joint monitoring probability of the node set to measure whether each pixel of the target area is covered. P_{th} is the expected coverage threshold, then:

$$P_{cov}(Cov_j) = \begin{cases} 1, & \text{if } P_{cov}(Cov_j) < P_{th} \\ 0, & \text{if } P_{cov}(Cov_j) \geq P_{th} \end{cases} \quad (6)$$

where $P_{cov}(Cov_j) = 1$ indicates that the target pixel j is overwritten; otherwise, the target pixel j is not overwritten [23].

The effective coverage points are calculated by the previous formula, and the number of sampling points of the whole target area is $l \times w$, so the coverage rate ϕ of the monitoring area A is calculated with the following formula:

$$\phi = \frac{\sum_j P_{cov}(Cov_j)}{l \times w} \quad (7)$$

The objective function in this work is the monitoring area coverage rate ϕ , the node in the monitoring area establishes the wireless sensor network coverage optimization model according to the previous formula.

$$\text{maximize } \phi = \frac{\sum_j P_{cov}(Cov_j)}{l \times w} \quad (8)$$

$$\text{where } \begin{cases} 0 \leq x_i \leq l \\ 0 \leq y_i \leq w \end{cases}.$$

2.3. Network Coverage Optimization Based on PSO

The PSO is inscribed in the family of evolutionary algorithms and is one of the most popular meta-heuristic optimization algorithms inspired by nature. This technique is used for any problem to explore a search in space to find the set parameters that maximize/minimize a particular goal [25].

The population in the PSO algorithm is called the swarm, and each individual in the group is called the particle. The displacement of any particle (as indicated above) is governed by very specific rules and conditions and is influenced by the movement of other particles in the neighborhood.

The algorithm is guided by personal experience (Pbest), overall experience (Gbest), and the present movement of the particles to decide their next positions in the search space. In D dimensional search space, we have M particles at certain speed flight. To find the optimal solution, PSO algorithm initializes a group of random particles through the iterative. With the speeds of each generation of particles, the location updates the objective function as follows [26],

$$v_{ij}(n+1) = w \times v_{ij}(n) + c_1 \times rand_1 \times (pbest_{ij} - x_{ij}) + c_2 \times rand_2 \times (gbest - x_{ij}) \quad (9)$$

- j : Refers to the dimension of the question searched by particles
- i : Refers to the number of particles
- n : Refers to iterations
- c_1 and c_2 : The accelerating constants in the algorithm
- $rand_1, rand_2$: Are two independent uniformly distributed random variables between (0,1)
- v_{ij} : Is usually valued in a certain range
- w : Is the inertia coefficient where:

$$w = wmax - \frac{n}{\maxnumber} \times (wmax - wmin) \quad (10)$$

The appreciation of the quality of its position is stopped by the value of the “objective” function at this point. It is essential that this particle can memorize the better position by which it has already passed, formulated as follows:

$$Pbest_{ij} = (pbest_{i1}, pbest_{i2}, \dots, pbest_{iD})$$

The best position reached by the particles of the swarm is formulated as follows:

$$Gbest = (gbest_1, gbest_2, \dots, gbest_D)$$

The notion of $Gbest$ (global best), is modeled on the PSO Global version (PSOG) where all the particles of the swarm are derived from the particle i . At the beginning of the algorithm, the particles of the swarm are initialized randomly/regularly in the search space. Subsequently at each iteration, the particles move, merging the three components:

- $v_{ij}(n)$: Is the speed of the particle in last iterative;
- $c_1 \times rand_1 \times (pbest_{ij} - x_{ij})$: The “cognitive” part which is the distance between the current position with best position of the particle, which indicates that particles can learn on itself;
- $c_2 \times rand_2 \times (gbest - x_{ij})$: The “social” part which is the distance between the current position with the global the best position, that represents the collaboration between the particles.

At the end of the displacement of the particles in a given iteration, the new positions are evaluated and the two vectors $Pbest_i$ and $Gbest$ are reindexed. According to Equation (11) in the case of a minimization of an objective function then Equation (12) in the PSOG can be achieved:

$$Pbest_i(t+1) = \begin{cases} Pbest_i(t), & \text{si } f(x_i(t+1)) \geq Pbest_i(t) \\ x_i(t+1), & \text{otherwise} \end{cases} \quad (11)$$

$$Gbest(t+1) = \operatorname{argmin}_{Pbest_i} f(Pbest_i(t+1)), 1 \leq i \leq N \quad (12)$$

This procedure is represented in the PSO algorithm, for N represents the number of particles of the swarm:

Step 1: Randomly initialize N particles: Position and speed.

Step 2: Evaluate particle positions.

Step 3: For each particle i , $Pbest_i = x_i$

Step 4: Calculate $Gbest$ according to (13)

Step 5: As if the stopping criterion is not satisfied, move the particles according to (9) and (10), assess particle positions and update $Pbest_i$ and $Gbest$ according to (12) and (13).

3. Coverage Hole Detection and Healing

The previous phase of work allowed us to identify the positions of the sensors used to build the air pollution monitoring wireless network and ensure full coverage of the RoI. After a period of time (one hour), it is necessary to ascertain if any of the sensors change its position or leave the RoI.

Updating the sensor locations allow to check the positions of the deployed sensors. If they remain their positions we keep the same deployed sensor for RoI coverage otherwise we move to the next phase of the work which is the holes detection and healing process. The second phase of project will be divided into two parts: The first is holes detection and coverage based on the VD, and the next is to repair the holes using the PiP algorithm.

3.1. Coverage Hole Detection Based on the VD

As the sensors used in this work are embedded in public mobile phones, so these nodes move randomly, which is why the network is able to crack at any time (mobile phone is turned off, leaves the RoI, is disconnected, etc.). To define the coverage hole introduced by the off-grid sensor and its exact location, we will introduce the VD algorithm to provide each sensor with their own cell. In this way we can define with accuracy the coverage hole location.

A VD is a partition of a space into small regions called the polygon of Voronoi based on closeness to points in a specific subset of the plane as each region contains one, and only one, site (point) and each point belongs to its region (cell) [27]. In this application, the VD devised the RoI into n cells where each sensor is the center of its own cell. To detect the coverage hole, we use the following theorems shown in Figure 1.

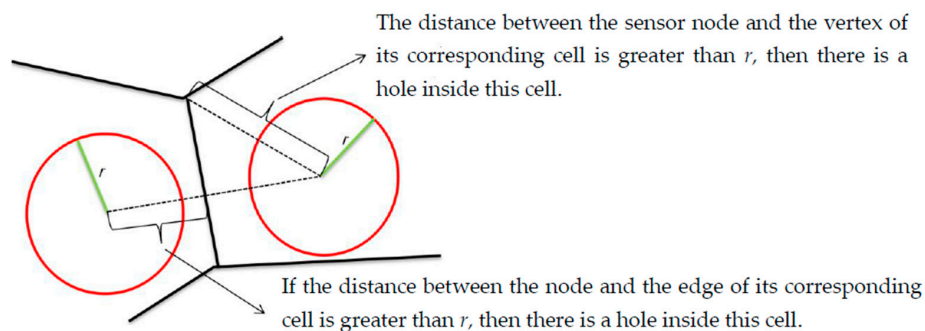


Figure 1. Coverage holes detection method.

3.2. Coverage Hole Healing Based on PiP Algorithm

After defining the location of the cell that contains a coverage hole using the VD, the next step is to find other sensors inside this cell in order to reapply the coverage algorithm and select the suitable sensors to deliver full coverage in the RoI. To complete this stage, we will use the PiP algorithm.

The PiP algorithm allows us to check, by way of a program, whether a particular point is inside a polygon or outside of it. A common way to approach the problem is to count how many times a line drawn from the point (in any direction) crosses the edge of the polygon. If the line and the polygon intersect an even number of times (or not at all), the point is outside. If they cross an odd number of times, the point is inside. This is true even for complex shapes that have a lot of coordinates and a very precise edge [28]. This process can be seen in Figure 2.

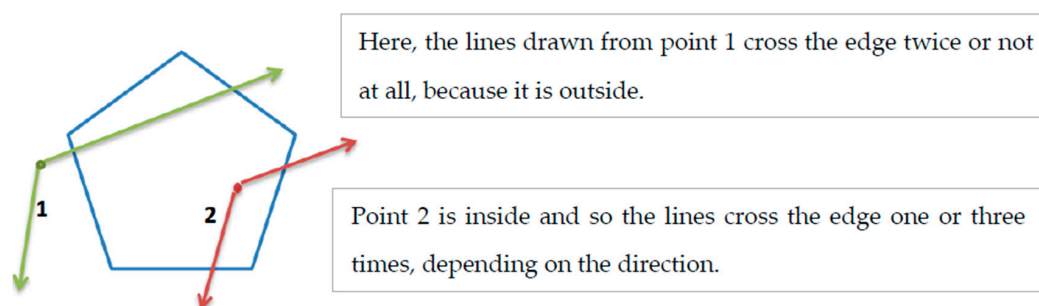


Figure 2. The principle of the point in polygon algorithm.

After having selected the sensors which are inside the cell of the coverage hole, we will apply to these sensors the algorithm of coverage already applied in the previous phase of this project (from function 3 to 8).

4. Simulation and Results Discussion

In this study, assuming that target area (RoI) A is two-dimensional plane (100 m^2), it is digitized into 100×100 grids where each grid size is equal to 1. In the simulation, the WSN with 100 total numbers of sensors is deployed on the target area with two assuming specifications:

- (1). All sensors have the same sensing range $r = 20 \text{ m}$ and communication range $r_c = 2 r$
- (2). Each sensor knows its location by a certain mechanism and the location information can be sent to the base station.

This project is defined in order to find the best distribution of the deployed sensors to obtain full coverage of the monitoring area. To achieve our objective, the system implementation will be passing through three basic phases:

- Coverage optimization of the monitoring area using PSO algorithm to find an optimal number of the deployed sensors to totally cover the RoI.
- Coverage hole detection based on the VD.
- Coverage hole healing using the algorithm of PiP to find sensors inside the hole then to use them to revalue the coverage of the hole using the coverage algorithm used in the first phase of work.

The effectiveness of the network coverage optimization strategy will be determined through a simulation experiment. Using a computer with a 1.10 GHZ processor, the coverage optimization algorithm of the WSN is carried out in MATLAB (R2013a) environment.

4.1. WSN Coverage Optimization

4.1.1. Application Method

In the simulated two-dimensional target area, assuming that all the deployed sensors in a big space have a large sensing range $r = 20 \text{ m}$, and each sensor knows its location by a certain mechanism and sends it to the base station. The coverage optimization based on the PSO algorithm should calculate the coverage rate of each pixel node for each sensor node according to Equation (4), calculate the joint coverage of each pixel node for each sensor node according to Equation (5), and calculate the coverage rate of the RoI using the objective function according to Equation (8), in the particle swarm algorithm also known as fitness.

In this work, we assume that there are M particles (the total number of the sensors in RoI, $M = 100$). The condition for the simulation experiment is described in the Table 1. The PSO algorithm is as follows:

- i. Randomly generate the position and velocity of each particle in the monitoring area;
- ii. Update the position and velocity of each particle according to Equations (9–11);
- iii. Calculate the fitness of each particle according to the optimization objective function;
- iv. Compare the particle fitness and the fitness of its own best position, if better, and then set the new *pbest*
- v. Compare the fitness of each particle and the fitness of the best position in the population, if better, set the new *gbest*
- vi. Until the maximum number of iterations is reached, the algorithm stops, otherwise proceed to step two
- vii. Output the optimal fitness and the corresponding particle position.

Table 1. The condition for simulation experiment.

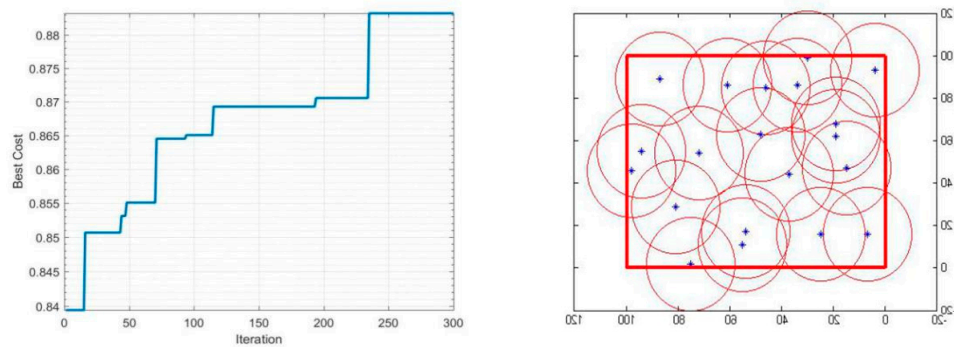
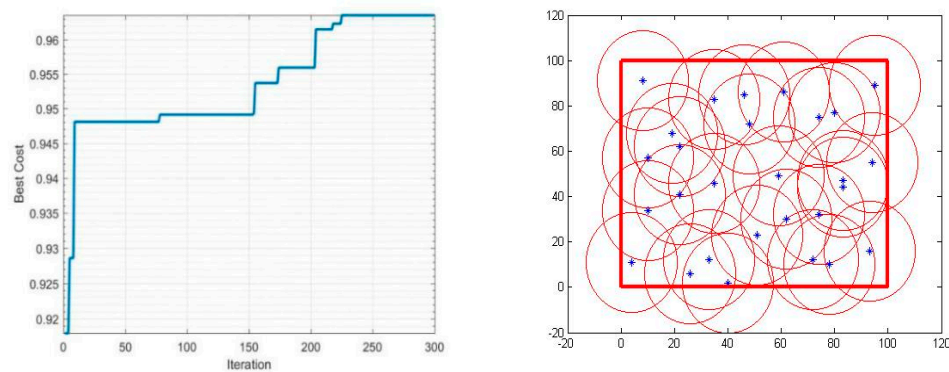
Population M	Deployed Sensors Number N	Swarm Size $D=2 \times N$	c_1, c_2	α_1, α_2 β_1, β_2	$re = \frac{r}{2}$	Maximum Iterations
100	20	40	2	1,0,1, 0.5	10	300
100	30	60	2	1,0,1, 0.5	10	300
100	35	70	2	1,0,1, 0.5	10	300
100	40	80	2	1,0,1, 0.5	10	300

4.1.2. Results and Discussion

In the simulation condition mentioned above, the detection radius r will be constant and equal to 20 m, and the number of sensors used will vary between 20 and 40 to determine its impact on the performance of the coverage. The iterations-coverage rate and the coverage effect are shown in Table 2, and Figures 3–6.

Table 2. Coverage optimization results.

Area A network coverage optimization performance test				
Sensor node number N	20	30	35	40
Coverage rate without optimization	75%	88%	91%	96%
Optimal coverage (%)	88%	96%	98%	100%

**Figure 3.** Coverage of 20 sensors.**Figure 4.** Coverage of 30 sensors.

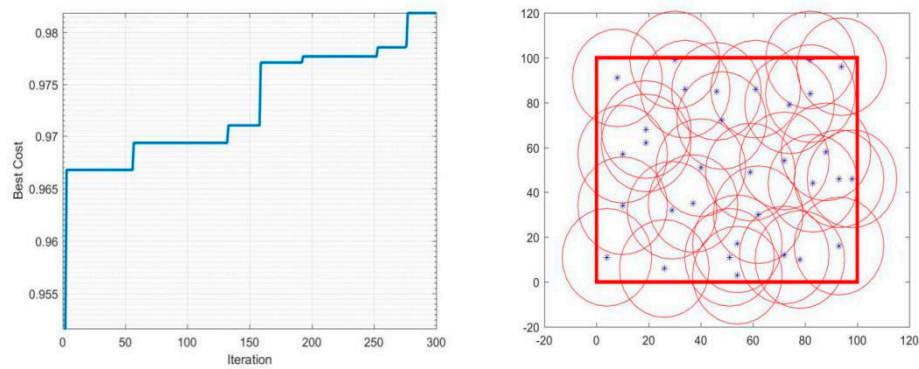


Figure 5. Coverage of 35 sensors.

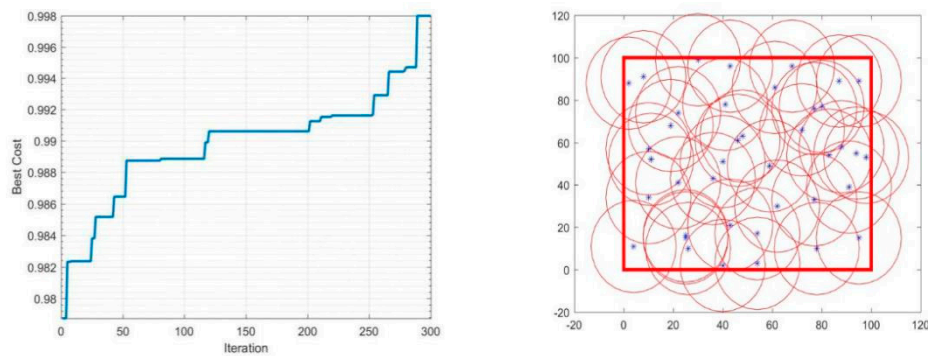


Figure 6. Coverage of 40 sensors.

During the optimization of coverage by using the PSO algorithm, the coverage optimization went through four stages, first the deployment of 20 sensors, then 30 sensors, followed by 35 sensors, and finally 40 sensors. The simulations results are as follows:

- The 20-sensor deployment result graph started with 84% coverage, after that the graph started to increase until it reached 87% at iteration 230 then jumped directly to the optimal result 88% and stabilized.
- The 30-sensor deployment result graph started with a 92% coverage rate and reached almost 95% after few iterations and remained practically stable until 150 iterations. After that, the algorithm started looking for a better result until iteration 225 where the final optimal coverage rate was 96%.
- The 35-sensor deployment result graph started with a 95% coverage rate, and reached almost 97% just after few iterations. After that, the algorithm started looking for a better result until iteration 280, where the final optimal coverage rate was 98%.
- The 40-sensor deployment result graph started with 97% coverage and increased to 99% at iteration 120 and maintained that rate until iteration 200. After that, the algorithm started looking for a better result. In this application the monitoring area was totally covered, and the rate reached 100%.

According to the results presented above, the curves of the coverage rate, with and without optimization according to the number of the deployed sensors, are shown in Figure 7.

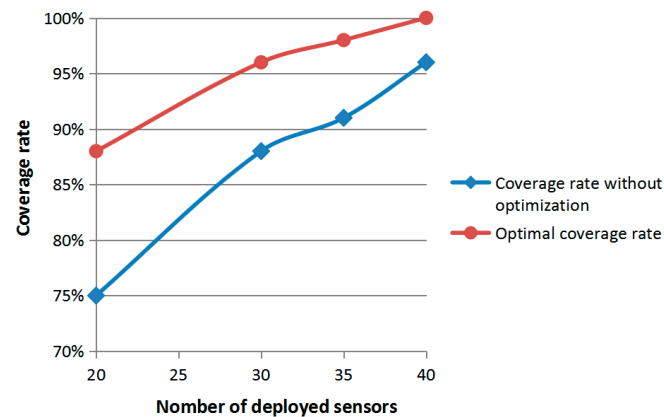


Figure 7. Coverage rate compared to the number of sensor deployment.

The analysis of the Figure 7 shows us that the PSO algorithm was more efficient when the number of sensors used was 20 where the percentage of the coverage rate had increased from 70% to 88%. This was revised in principle of the PSO algorithm which attempted to find a better solution in bad conditions.

It is also noted that the curve slope between the 20 sensors and 35 sensors was large compared to the curve slope between the 35 sensors and 40 sensors. We can conclude that in a RoI with our characteristics ($L \times W, r, n, m$), the optimal number of deployed sensors to have total coverage of the monitoring area varied between 35 and 40 sensors.

In reference [23], 20 sensors from total 40 sensors were used to completely cover an area of 20 m². Therefore, the number of deployed sensors was optimized by 50%. In our work, the deployment of 40 sensors from a total number of 100 sensors was enough to fully cover an area of 100 m². Therefore, we optimized our deployment by 60%. It is obvious that the PSO algorithm successfully found the best deployment strategy for this application. The position and the number of deployed sensors were determined to fulfill the goal of maximum coverage of the monitoring area.

4.2. Coverage Holes Detection

4.2.1. Application Method

The PSO algorithm allowed us to optimize a total coverage of the RoI and to collect the best positions of the deployed sensors. After a period of time (one hour), the base station updated these positions leading to one of the deployed sensors out of the coverage network. Following this, the VD will be used to define the position of the coverage hole.

The target area (RoI) A is also a two-dimensional plane (100 m²). It is digitized into 100 × 100 grids where each grid size is equal to 1. The total number of deployed sensors in coverage optimization is 40 sensors. In this process, we will first randomly move one of these sensors and assume that, after updating the sensor positions, it has been found that the sensor has pulled out of the coverage network as shown in Figure 8. Following this, the 39 sensors remaining in the RoI (their positions are already known) are used to partition the RoI into 39 cells and to discover the location of the coverage hole using the VD assuming that the theorems 1 and 2 are already defined.

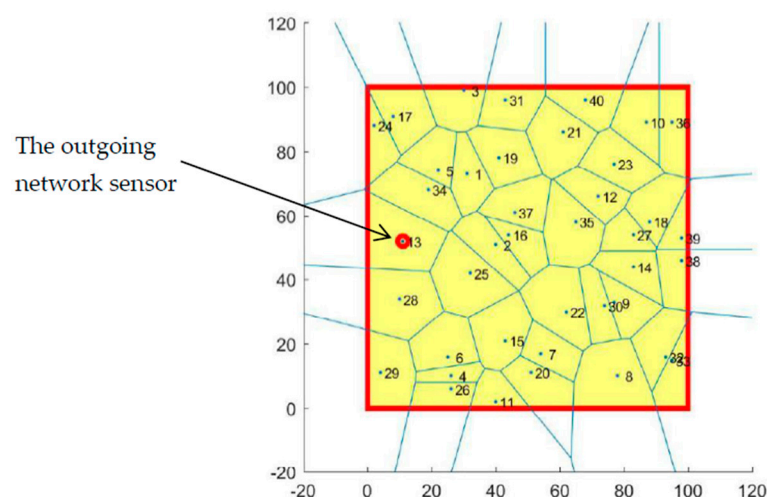


Figure 8. Voronoi cells of the 40 deployed sensors.

4.2.2. Results and Discussion

Using the VD algorithm, the monitoring area was divided into several cells using the selected sensors for full coverage. Each sensor had its own cell that gave us a 40-cell area with no coverage hole found.

The sensor indicated in Figure 8 was randomly chosen to exit the coverage network. The VD algorithm was reapplied to the remaining 39 sensors in the area to find the coverage holes introduced by the outgoing sensor. The result is shown in the Figure 9.

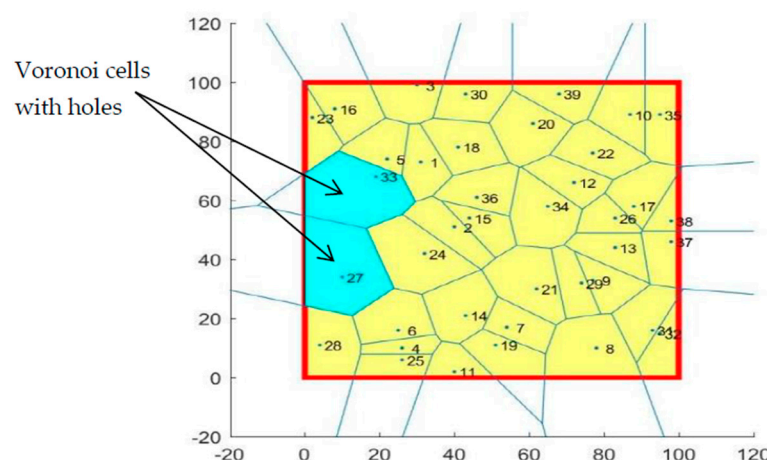


Figure 9. Voronoi cells of remaining 39 sensors.

It was assumed that, after a period of time (one hour), the base station updated the positions of the 40 sensors deployed and that one of the sensors came out of the network. Therefore, the next application started by randomly selecting a sensor to exit the network and then to apply the VD algorithm on the remaining 39 sensors in the network to discover the location of the coverage holes. The results of this application are shown in Figures 9 and 10.

According to Figure 10a, it is noted that the sensor (13) that left the network was at the neighborhood of sensors (34), (25) and (28) so they then covered the space already covered by the sensor (13) but their sensing range was not enough to cover all the space. For this reason, we found a coverage hole between cell (27) and (33), as shown in Figure 10b.

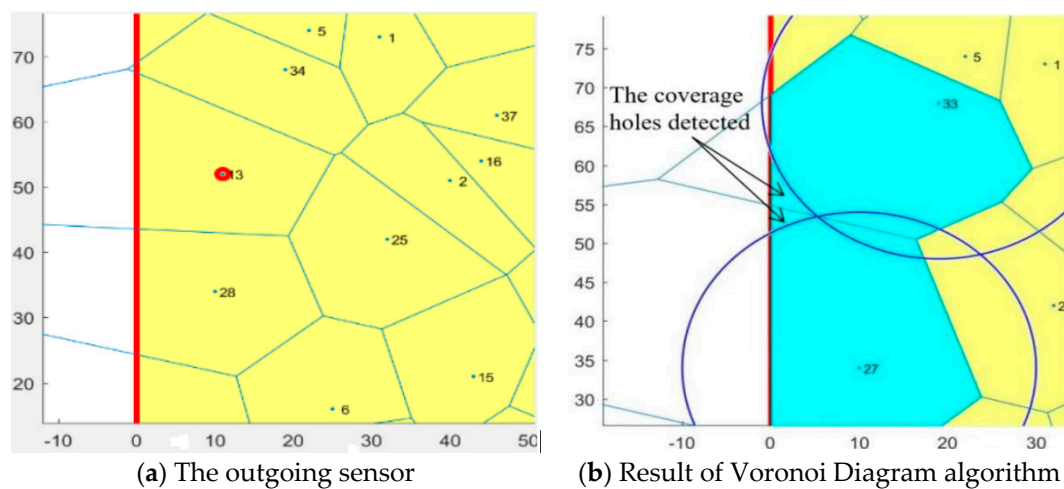


Figure 10. The coverage hole detected by the Voronoi Diagram algorithm.

4.3. Coverage Holes Healing

The next step in this application was to search for the sensors that were originally in the coverage hole cells (cell 27 and 33) by applying the PiP algorithm. Following this, we reapplied the coverage code to find the best sensor to enhance the coverage. After applying the PiP algorithm in the coverage hole cells, we found three sensors in this area as shown Figure 11.

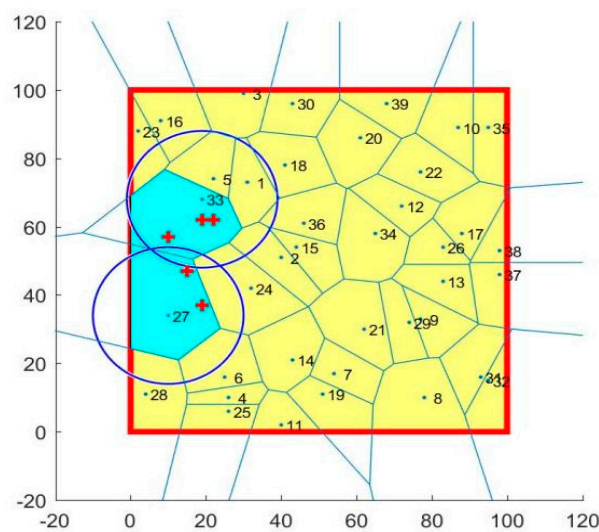


Figure 11. Point in polygon results.

After finding five sensors between cells 27 and 33, we then reapplied them to the coverage algorithm to ascertain the best sensor that could cover the hole and join it to the monitoring area network. The result is shown in Figure 12.

To heal the coverage hole found by the VD algorithm, the PiP algorithm was applied to find the sensors inside the hole cells. The PiP algorithm found three sensors in these two cells and then the coverage algorithm was reapplied to select one of these three sensors. We then latter selected a sensor that covered the hole and increased the coverage rate to 99.13%.

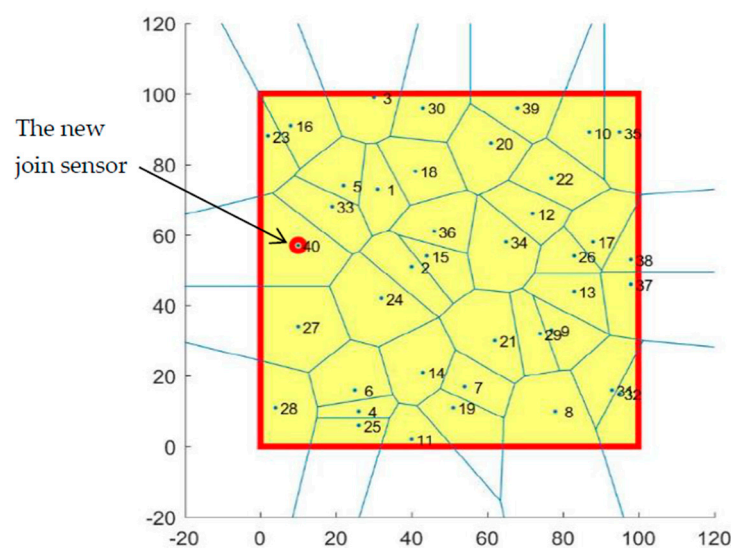


Figure 12. The distribution of the new sensors in the monitoring area network.

5. Conclusions

A system that monitors indoor air pollution by combining wireless sensing technology, PSO, VD, and PiP algorithms is proposed in this paper. A PSO algorithm successfully found the best deployment strategy for this application. The simulation results indicate that in a RoI with our characteristics ($l \times w$, r , n , m), the optimal number of deployed sensors varies between 35 and 40 sensors to have total coverage of the monitoring area. Because the initial positions of the sensors were randomly selected, the results will not be the same each time. Admittedly, there may be a small possibility that the initial deployment plan is the optimal one. However, the objective is unchanged and fulfilled by adopting the PSO algorithm, which is to find the best deployment strategy of the sensor network. Moreover, VD successfully finds the exact location of the hole coverage which allowed the application of PiP algorithm inside those holes and found a new join sensor to enhance the network. However, PSO is a heuristic algorithm that requires a big server to get fast results. Therefore, to apply this work in real life in the future, the base station must use big servers for a good quality of service.

Author Contributions: The work described in this article is the collaborative development of all authors. A.N. contributed to the idea of data processing, designed the algorithm, writing and analysis. L.F. supervised the application and provided technical advice on simulation process. G.M. made contributions to programming and data measurement. L.W. provided important advice on the formula derivation as well as the paper revision.

Funding: This research was funded by National Natural Foundation of China (No. 61601010), Algerian Space Agency, and China Great Wall Industry Corporation.

Acknowledgments: Except the support from funding, the authors would like to thank the support of the automation and electrical teaching research laboratory center, as well as Beihang university.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Matko, V.; Brezovec, B. Improved data center energy efficiency and availability with multilayer node event processing. *Energies* **2018**, *11*, 2478. [\[CrossRef\]](#)
2. Jia, M.; Srinivasan, R.S.; Raheem, A.A. From occupancy to occupant behavior: An analytical survey of data acquisition technologies, modeling methodologies and simulation coupling mechanisms for building energy efficiency. *Renew. Sustain. Energy Rev.* **2017**, *68*, 525–540. [\[CrossRef\]](#)
3. Podberšič, M.; Matko, V.; Šegula, M. An EMI filter selection method based on spectrum of digital periodic signal. *Sensors* **2006**, *6*, 90–99. [\[CrossRef\]](#)

4. Antonini, G. SPICE Equivalent circuits of frequency—Domain responses. *IEEE Trans. Electromagn. Compat.* **2003**, *5*, 502–512. [\[CrossRef\]](#)
5. Berezowski, K.S. The landscape of wireless sensing in greenhouse monitoring and control. *Int. J. Wirel. Mob. Netw.* **2012**, *4*, 141–154. [\[CrossRef\]](#)
6. Octavian, P.; Pedro, S.G. Smart sensors network for air quality monitoring applications. *IEEE Trans. Instrum. Meas.* **2009**, *58*, 3253–3262.
7. Spachos, P. Designing learned CO₂-based occupancy estimation in smart buildings. *IET Wirel. Sens. Syst.* **2018**, *8*, 249–255.
8. Jung, Y.; Chao, H.; Sang, M. ISSAQ: An integrated sensing systems for real-time indoor air quality monitoring. *IEEE Sens. J.* **2014**, *14*, 4230–4244.
9. Spachos, P.; Dimitrios, H. Real-time indoor carbon dioxide monitoring through cognitive wireless sensor networks. *IEEE Sens. J.* **2015**, *16*, 506–514. [\[CrossRef\]](#)
10. Kazuo, S. Basic study of indoor air quality improvement by atmospheric plasma. *IEEE Trans. Ind. Appl.* **2016**, *52*, 1823–1830.
11. Mohieddine, B.; Abderrazak, A.; Sabbir, A.H.M.; Farid, T.; Abdullah, K. Real-time indoor air quality monitoring through wireless sensor network. *Int. J. Internet Things Web Serv.* **2017**, *2*, 7–13.
12. Daudi, S.S. Industrial air pollution monitoring system based on wireless sensor networks. *J. Inf. Sci. Comput. Technol.* **2017**, *6*, 612–623.
13. Marques, G.; Pitarma, R. An indoor monitoring system for ambient assisted living based on internet of things architecture. *Int. J. Environ. Res. Public Health* **2016**, *13*, 1152. [\[CrossRef\]](#) [\[PubMed\]](#)
14. Pitarma, R.; Marques, G.; Caetano, F. Monitoring indoor air quality to improve occupational health. In *New Advances in Information Systems and Technologies*; Springer International Publishing: Basel, Switzerland, 2016; pp. 13–21.
15. Lozano, J.; Suárez, J.I.; Arroyo, P.; Ordiales, J.M.; Álvarez, F. Wireless sensor network for indoor air quality monitoring. *Chem. Eng. Trans.* **2012**, *30*, 319–324.
16. Kalaimani, G.; Sakthivel, S. Wireless sensor network-based air quality monitoring system for industrial area. *Int. J. Sci. Eng. Res.* **2015**, *3*, 1–5.
17. Yu, T.C.; Lin, C.C. An intelligent wireless sensing and control system to improve indoor air quality: Monitoring, prediction, and preaction. *Int. J. Distrib. Sens. Netw.* **2015**, *11*, 140978. [\[CrossRef\]](#)
18. Boussaïd, I. Perfectionnement de Metaheuristiques Pour l’Optimisation Continue. Ph.D. Thesis, University of Science and Technology Houari Boumediene, Bab Ezzouar, Algeria, 2013.
19. Zou, Y.; Chakrabarty, K. Sensor deployment and target localization based on virtual forces. In Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, IEEE Societies, San Francisco, CA, USA, 30 March–3 April 2003.
20. Qu, Y. Wireless Sensor Network Deployment. Ph.D. Thesis, Florida International University, Miami, FL, USA, 2013.
21. Azlina, N.; Ammar, W.; Mohammad, Y.A. A wireless sensor network coverage optimization algorithm based on particle swarm optimization and voronoi diagram. In Proceedings of the International Conference on Networking, Sensing and Control, Okayama, Japan, 26–29 May 2009.
22. Ait Saadi, N. Multi-Objective Wireless Sensor Network Deployment. Ph.D. Thesis, Pierre and Marie Curie University, Computer Science Telecommunication and Electronic School, Paris, France, 2010.
23. Changxing, L.; Qing, Z.; Long-yao, Z. Research on wireless sensor network coverage based on improved particle swarm optimization algorithm. In Proceedings of the International Conference on Computer Network, Xian, China, 10–12 October 2017.
24. Abdelfatah, F. Rapport: Développement d’une bibliothèque de capteurs. Montpellier 2 University, Science and Technology France. 2008. Available online: www.lirmm.fr (accessed on 23 February 2019).
25. Mohamed Anis, B. L’optimisation par essais de particules: Le concept lire des nuées d’oiseaux. Available online: <https://slideplayer.fr/slide/1149535/release/woothue> (accessed on 23 February 2019).
26. Xia, J.B. Coverage optimization strategy of wireless sensor network based on swarm intelligence algorithm. In Proceedings of the International Conference on Smart City and Systems Engineering (ICSCSE), Hunan, China, 25–26 November 2016.

27. Dai, G.Y.; Lv, H.X.; Chen, L.Y.; Zhou, B.; Xu, P. A novel coverage holes discovery algorithm based on voronoi diagram in wireless sensor networks. *Int. J. Hybrid Inf. Technol.* **2016**, *9*, 273–282. [[CrossRef](#)]
28. AssemblySys Dataservices. Geographic and Thematic Databases. Algorithm du point dans un polygone en PHP. Available online: <https://assemblysys.com/fr/algorithme-point-dans-polygone/> (accessed on 13 September 2018).



© 2019 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).