3dDABA: An Algorithm Covering a Three-dimensional WSN Area

Li Tan, Jiaqi Shi, Xiaojiang Tang, Xiaofeng Lian, Haoyu Wang

School of Computer Science and Engineering, Beijing Technology and Business University, China
tanli@th.btbu.edu.cn, 18811357269@163.com, tangxiaojiang_011@126.com,
lianxf@th.btbu.edu.cn,wanghaoyuxw@163.com

Abstract

Due to the complexity of the three-dimensional deployment environment, it is difficult for a single node to complete a full coverage task for the target, while the deployment time of a large number of nodes is long and the algorithm complexity is high. In this paper, a Three-dimension wireless sensor network Deployment Algorithms based on Basic Architecture (3dDABA) is proposed to monitor the target region by constructing basic architecture in different regions. First, two types of basic architecture are identified, then a method of constructing basic architecture is proposed, and the coverage deployment strategy of the 3D sensor network is designed based on basic architecture. Experimental results show that the 3dDABA algorithm can reduce the deployment time by two-thirds compared with the Voronoi graph method, reduce the complexity of the algorithm and reduce the deployment time of the whole network.

Keywords: The deployment of WSN, Basic architecture, Coverage, Travel distance, Iteration time

1 Introduction

Three-dimensional wireless sensor networks provide many scientific, environmental, commercial and military applications, such as atmospheric pollution monitoring, natural disaster prevention and emergency rescue, easy navigation, land and sea exploration, and oil and gas monitoring. Deployment coverage is one of the most critical issues for three-dimensional (3D) sensor networks, and it directly affects the performance of wireless sensor networks.A large number of sensor nodes deployed in an unknown environment can make the sensor network more stable. Some achievements have been made in 3D deployment. Vieira et al. [1] have studied the 3D wireless ad hoc networks' basic characteristics. The random, uniformly-distributed 3D wireless ad hoc networks were estimated for link probability, node degree, and coverage. Miao et al. [2] have proposed a algorithm of 3D autonomous deployment, in order to ensure the connectivity of the entire network they introduced a negotiation strategy. At the same time, a density control strategy was also introduced to adjust the sensor nodes' position, so that the nodes can be distributed more evenly and are in a 3D regional autonomous deployment. Brown et al. [3] have studied indoor wireless video sensor networks for it's 3D full-space coverage issue using greedy heuristics and enhanced depth-first algorithms, which implements solutions to limited node coverage in 3D space without considering other factors.

Li et al. [4] and Boufares et al. [5] autonomously deployed a 3D mobile wireless sensor network using a virtual force strategy, in which each sensor node could receive the virtual force in order to move in a 3D target space. Boufares et al. [6] autonomously deployed the 3D plane and through the autonomous movement of the nodes to completed the node deployment using a virtual force strategy. Sung et al. [7] combined a Voronoi diagram and proposed a distributed greedy algorithm to improve the coverage of directed sensor networks. Other research [8] proposed an autonomous deployment algorithm for k-coverage on three-dimensional surfaces. The algorithm first divides Voronoi into three-dimensional surfaces and then computes the Chebyshev center of each Voronoi. The nodes move to the center of the Voronoi area. The algorithm continues to iterate until the location of the nodes no longer changes to complete deployment. This algorithm continues to iterate except the location of the nodes do not changes, and then the deployment is completed. With a goal of ensuring coverage, Aslam [9] focused on how to minimize the number of nodes required in the target area. Aslam [9] focused on how to make the number of nodes minimized required in the target area First, randomly deployed the sensor nodes in the target area, and then switched the sensor nodes placed in the target location to the active state. In order to reduce energy consumption the nodes were switched to a sleep state. Nguyen et al. [10] optimize the number of barrier paths by minimizing the total moving distance and the number of moving sensor nodes or sensors using two algorithms. These methods optimize

---

*Corresponding Author: Li Tan; E-mail: tanli@th.btbu.edu.cn

DOI: 10.3966/160792642020122107009

the deployment of nodes, reduce energy consumption, and improve coverage.

Cao et al. [11] proposed a distributed particle swarm optimization algorithm of 3D wireless sensor network deployment and discussed deployment strategies for situations when the deployment space encountered obstacles. Yang et al. [12] first proposed the method of detect the coverage holes in 3D surface wireless sensor networks use discrete wavelet transform and completed the 3D surface nodes' deployment by improving the artificial bee colony algorithm. These algorithms study coverage in different situations in the deployment space. However, all of the above methods have mainly studied the coverage. The deployment time of the entire network is generally long and cannot effectively reduce the complexity of deployment.

Based on the Voronoi diagram, this paper proposes an algorithm based on basic architecture- 3dDABA (Three-dimension wireless sensor network Deployment Algorithm based on Basic Architecture). By constructing basic architecture to monitor different targets or data, both the algorithm's complexity and the deployment time of the entire network can be reduced.

2 3dDABA Algorithm

2.1 Description of Basic Architecture

Traditional sensor network deployment generally requires a certain number of nodes, and the nodes perform dense coverage, which can avoid large-scale network segmentation and loopholes. The initial space of the 3D sensor network model is wide, which may lead to uneven network density and network segmentation and loopholes.

Due to the complex network deployment environment in 3D space, a single sensor node cannot complete the entire monitoring task for complex targets. Therefore, the concept of "basic architecture" is proposed in this paper. The basic architecture is the smallest structural unit composed of several sensor nodes for complex targets. Every basic architecture is treated as a whole during deployment. During the deployment process, the internal structure of the basic architecture remains unchanged. The entire wireless sensor network is composed of multiple basic architecture units that jointly complete monitoring tasks in complex environments. The basic architecture has no fixed structure; it is designed to be suitable for different application environments. Therefore, it needs to be comprehensively considered based on the application environment of the wireless sensor network, the type of monitoring target, and the monitoring task. Each node in the basic architecture has its randomly generated initial location and can move autonomously during deployment.

Definition of basic architecture: Given the characteristics of three-dimensional sensor network area coverage, two basic architectures are proposed – based on a cube and one based on a regular dodecahedron.

(1) Basic architecture based on cube

Basic architecture has an omnidirectional sensing stereo node model. The basic architecture is composed of eight isomorphic omnidirectional sensor nodes, and the position of each node corresponds to the eight vertices of the cube, forming a basic architecture based on the cube structure. It then selects one node as the leader node and determines its location. The positions of other member nodes are shown in Figure 1. In this paper, S_1 node is selected as the leader node. The edges of basic architecture are $a = \frac{2}{\sqrt{3}} R_s$. Figure 2 presents a simulation diagram of a basic architecture based on a cube to represent the coverage of the node.

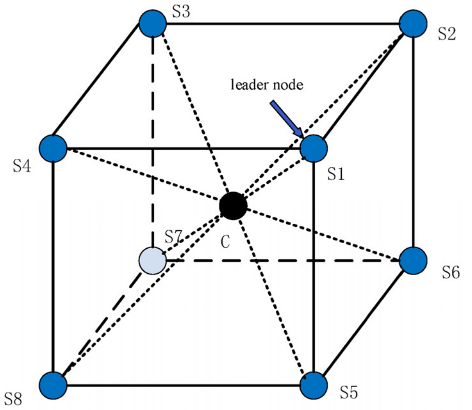


Figure 1. Basic architecture based on a cube

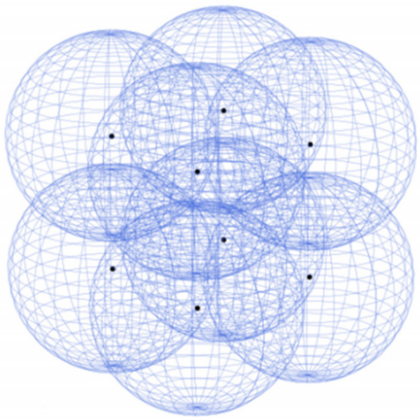


Figure 2. Simulation diagram of a basic architecture based on a cube

Assuming that the coordinates of the nodes S_1 are $\left(x_{S_1}, y_{S_1}, z_{S_1}\right)$, the formula for calculating the coordinates of the center P of the basic architecture is shown in below:

$$P(x_P, y_P, z_P) = (x_{S_1} - \frac{a}{2}, y_{S_1} - \frac{a}{2}, z_{S_1} - \frac{a}{2}) \qquad \textbf{(1)}$$

From this, the mapping relationship between the middle coordinates of the basic architecture and the coordinates of the members of the basic architecture can be obtained, as shown in Equation 2, where the range for i is {1, 2, 3, 4, 5, 6, 7, 8}.

$$S_i(x_{S_i}, y_{S_i}, z_{S_i}) = (x_P \pm \frac{a}{2}, y_P \pm \frac{a}{2}, z_P \pm \frac{a}{2}) \quad (2)$$

(2) Basic architecture based on a regular dodecahedron

The node model of basic architecture based on a regular dodecahedron is different from the node model based on a cube. The node's effective sensing area is the inscribed regular dodecahedron of the node perception ball. The regular dodecahedron node model is shown in Figure3.

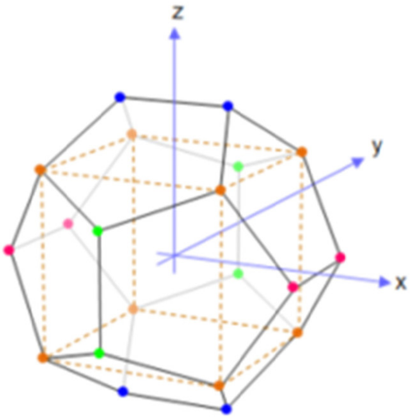


Figure 3. Regular dodecahedron node model

A regular dodecahedron is a regular polyhedron which composed of twelve regular pentagons. It has twenty vertices and thirty edges. Assuming that the coordinates of the current node S_1 are $\left(x_{S_1}, y_{S_1}, z_{S_1}\right)$, and the perceived radius of the node is R_S, the formula for calculating the length of the regular dodecahedron is shown in below:

$$a = \frac{2R_s}{\sqrt{3}\varphi} \quad (3)$$

Where φ is the golden section number that is calculated as follows:

$$\varphi = \frac{1+\sqrt{5}}{2} \quad (4)$$

The face-to-center distance of a regular dodecahedron is calculating in the following manner:

$$d = a\frac{\varphi^2}{2\sqrt{3-\varphi}} \quad (5)$$

In Figure 3, the coordinate system is established by the the node's position, that is, the center of the regular dodecahedron. Then the coordinates of each vertex of the regular dodecahedron can be divided into four types, which are shown in Figure 3 as orange, green, and pink vertices. Orange vertex coordinates are $\left(\pm\varphi a, \pm\varphi a, \pm\varphi a\right)$, green vertex coordinates are $\left(0, \pm\frac{1}{a}, \pm a\right)$, blue vertex coordinates are $\left(\pm a, 0, \pm\frac{1}{a}\right)$, and pink vertex coordinates are $\left(\pm\frac{1}{a}, \pm a, 0\right)$.

A basic architecture based on a regular dodecahedron is composed of thirteen sensor nodes, and the node located at the center position is used as the leader node (S_1 as shown in Figure. 4. The positions of member nodes are determined based on the leader node's position. Each face of the central node is adjacent to a member node. Thirteen nodes together form a basic architecture based on a regular dodecahedron, as shown in Figure 4. The basic architecture simulation based on a regular dodecahedron is shown in Figure 5 to represent the coverage of the node.

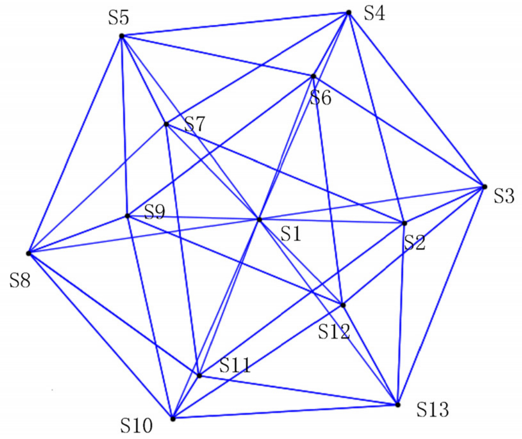


Figure 4. Basic architecture based on a regular dodecahedron

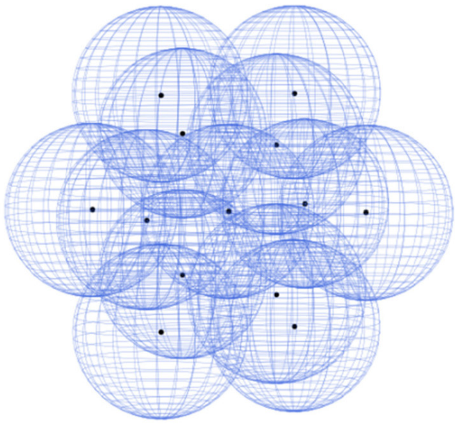


Figure 5. Simulation diagram of basic architecture based on a regular dodecahedron

In a basic architecture based on a regular dodecahedron, the distance between the leader node and member nodes is equal. The nodes' distance is calculated as follows:

$$D = 2d = a\frac{\varphi}{\sqrt{3-\varphi}} \tag{6}$$

The direction vector from the leader node to the member nodes is the direction vector from the leader node to each face. The plane center of each face of the leading node is N_i, where i is {1, 2, 3, ..., 12}, and the coordinates of N_i are calculated as follows:

$$N_i(x_{N_i}, y_{N_i}, z_{N_i}) = \frac{(\sum x_{v_i}, \sum y_{v_i}, \sum z_{v_i})}{5} \tag{7}$$

where $\left(x_{v_i}, y_{v_i}, z_{v_i}\right)$ are the coordinates of the vertices on each face. Direction vector from the leader node to other member nodes is calculated as follows:

$$\overrightarrow{\alpha_i} = \frac{N_i - S_1}{|N_i - S_1|} \tag{8}$$

From this, the mapping relationship between the coordinates of the member nodes and the coordinates of the leader node can be found as follows:

$$S_i = S_1 + D\overrightarrow{\alpha_i} \tag{9}$$

Assuming that the current node is $C_i\left(x_{C_i}, y_{C_i}, z_{C_i}\right)$, then the set of neighbor nodes of the node C_i is defined as follows:

$$C_n = \{C_j \mid D_{(i,j)} < D_{th}\} \tag{10}$$

Where $D_{(i,j)}$ is the Euclidean distance between node C_i and node C_j, and D_{th} is defined as a critical distance that determines whether the two nodes are each other's neighbors. The following is a calculation for $D_{(i,j)}$:

$$D_{(i,j)} = \sqrt{(x_{C_i} - x_{C_j})^2 + (y_{C_i} - y_{C_j})^2 + (z_{C_i} - z_{C_j})^2} \tag{11}$$

The specific steps of the construction method are listed below:

(1) Build a neighbor node set for all sensor nodes.

(2) Mark the status of each node. A status value of 0 indicates that the node has not formed a basic architecture with other nodes, and a status value of 1 indicates that it has formed a basic architecture with other nodes.

(3) Traverse the nodes to determine the current node status information. If the status value is 0, go to step (4), if the status value is 1, go to step (8).

(4) Traverse the neighbor node set of the current node and count the number of nodes with a status value of 0 in the neighbor node-set. If the number of neighbor nodes with a status value of 0 is ≤ 7, go to step (8), If the number is ≥ 7, but ≤ 12, go to step (5). If the number of neighbor nodes whose status value is 0 is ≥ 12, go to step (6).

(5) The first seven nodes with a 0 status value of the current node and the neighbor node constitute a basic architecture set. Designate the current node as the leader node, build a basic architecture based on a cube, and then go to step (7);

(6) The first twelve neighbor nodes with a 0 status value of the current node and the neighbor node constitute a basic architecture set. Calculate the center position of 13 nodes, and designate the node closest to the center position as the leader node. Build a basic architecture based on a regular dodecahedron, and then go to step (7);

(7) Change the status value of all nodes that participated in the construction of the basic architecture to 1;

(8) If all nodes are traversed, go to step (8); otherwise, go to the next node and return to step (3);

(9) Exit.

2.2 3dDABA Algorithm

In this paper, 3dDABA algorithm that uses the sensor nodes inside a space is proposed for building basic architecture. Based on the center position of the basic architecture that is the center node and the radius of the basic architecture, a 3D weighted Voronoi diagram is constructed, the Voronoi diagram's center corresponding to each basic architecture is calculated, and the center of the basic architecture is moved to the center of the corresponding Voronoi region. Then each member node location is calculated using the center of the basic architecture and the mapping relationship between the members of the node, and multiple iterations are completed with the deployment.

The specific deployment algorithm is outlined below. Where, the critical distance of each other's neighbor nodes is used to judge whether two nodes are neighbor nodes. If the critical value is greater, they cannot be neighbor nodes:

```
Input: the number of nodes $N$, the perceived radius
       of nodes $r$, the critical distance of neighboring
       nodes $D_{th}$ and the Boundary of the region
       $Boundary$;
Output: final position of node $C'$;
(1)  Set the maximum number of iterations as Maxloop;
(2)  Set perceive radius of sensor nodes as $R$;
(3)  Set the boundary monitoring area as $B$;
(4)  Set the critical distance of neighbor nodes as $D_{th}$;
(5)  Set the number of nodes as $N$;
(6)  Randomly set sensor node initial position as $C$;
(7)  Set n=0;
(8)  Compute $D_{(i,j)}$
(9)  if $D_{(i,j)} < D_{th}$: $C_n \leftarrow C_j$;
(10) Create Basic Architecture
```

```
While n < maxloop do
    $V$ ←Voronoi ( $R$ , $B$ , $D_{th}$ )//Construct 3d Voronoi diagram
    for i<=1 to Length ( $V$ ) do
        Compute $P$
        Add $C_i^{'}$ to $C_i$
    End for
    $C \leftarrow C^{'}$
End while
```

In literature [13], the algorithm complexity of constructing a 3D Voronoi diagram is $O(n^2)$. the algorithm presented in this paper constructing a 3D Voronoi diagram through m iterations, so the time complexity is $O(mn^2)$.

3 Simulation Results and Analysis

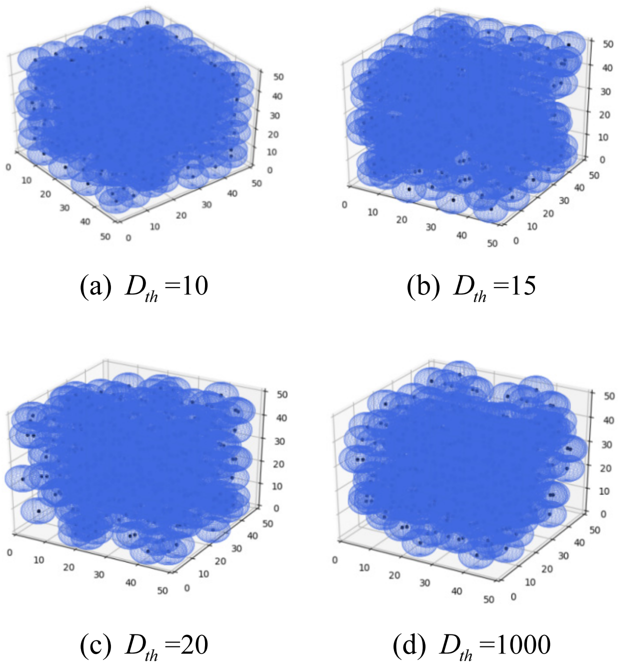
3.1 The Simulation Results

The experiment was conducted using Python 2.7 as the simulation environment and Intel Core i7 6700 HQ CPU. The monitoring area was set to the three-dimensional space of $50m \times 50m \times 50m$. At the start, all nodes were randomly deployed within the monitoring area. The node's perception of radius was 5 m, and the number of nodes was 350. The basic architecture was completed within the three-dimensional space upon the node deployment.

Four different values of D_{th} (10, 15, 20 and 1000) were selected for simulation. In each case, all nodes in the region were neighbors of each other with two times perception radius, three times perception radius, and four times perception radius. By constructing the basic architecture, the deployment algorithm was simulated in the target monitoring area by using a weighted Voronoi diagram. The simulation structure is shown in Figure 6 and Figure 7. Figure 6 shows the coverage of nodes over the whole monitoring area after the algorithm has been iterated 60 times, and Figure 7 shows the simulation results of the basic architecture composed of nodes after the algorithm has been iterated 60 times. In Figure 7 blue represents basic architecture based on a regular dodecahedron and red represents basic architecture based on cube. A single node that does not form a basic architecture with other nodes is considered a special basic architecture, and the single nodes participate in the deployment algorithm at the same level as other basic architectures.

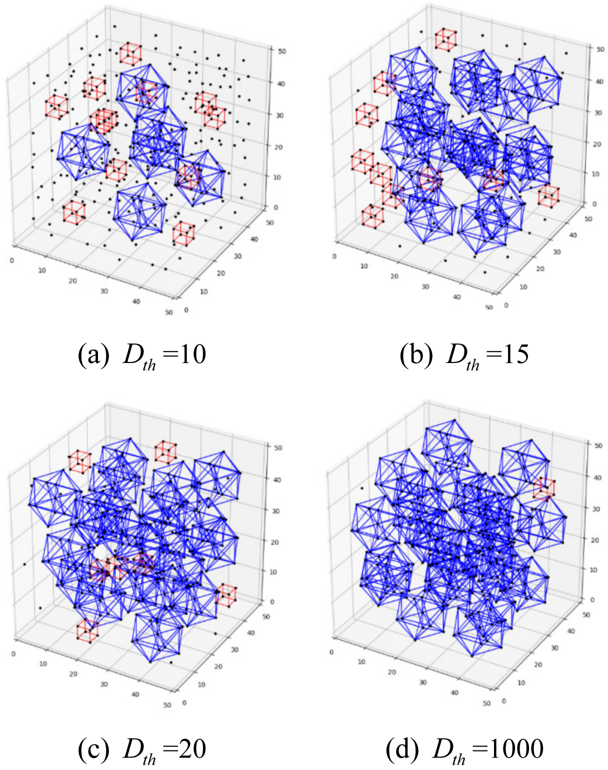
3.2 Results Analysis

The performance of the proposed algorithm was evaluated using the following three indicators: coverage, deployment time, and movement distance.


(a) D_{th} =10 (b) D_{th} =15

(c) D_{th} =20 (d) D_{th} =1000

Figure 6. Simulation diagram of node deployment


(a) D_{th} =10 (b) D_{th} =15

(c) D_{th} =20 (d) D_{th} =1000

Figure 7. Simulation diagram of basic architecture during deployment

Coverage is a critical indicator of the performance that reflects the comprehensive monitoring capability of a sensor network for monitoring objects in the required monitoring area. It is calculated in the following manner:

$$\text{Coverage} = \frac{Area\ covered\ by\ nodes}{Total\ area\ covered} .$$

Deployment time refers to the time required to cover as many areas as possible at the point of deployment, that is, the time when coverage reaches a relatively stable state. A node can change its position by moving from its initial position and stop moving when it reaches the target position.

$$\text{Deployment time} = t_{finally} - t_{begin}\,.$$

In which $t_{finally}$ = the time at which the node finally determines its location and t_{begin} = tthe time at which the node begins to move.

Movement distance refers to the distance that nodes travel during the whole deployment process, i.e., the distance that nodes move from the initial location to the location of the final deployment.

$$\text{Movement distance} = \sqrt{\left(\mathrm{x}_{final} - \mathrm{x}_{initial}\right)^2 + \left(\mathrm{y}_{final} - \mathrm{y}_{initial}\right)^2}$$

In which x_{final} and $x_{\mathrm{int}ital}$ are the final x and intital x coordinate of the node, y_{final} and $y_{\mathrm{int}ital}$ are the final y and intital y coordinate of the node.

In the experiment, the traditional Voronoi diagram deployment method was used as the control group to verify the influence of the basic architecture algorithm. At the same time, a critical value between different nodes was selected for the proposed algorithm to discuss the influence of a different number of basic architectures on the algorithm.

In Figure 8, The x-coordinate represents the number of iterations, and the y-coordinate represents the coverage of the node to the region, the Voronoi method and the four cases of the proposed method are compared, in which D_{th} =10 is represented by 3dDABA-A, D_{th} =15 is represented by 3dDABA-B, D_{th} =20 is represented by 3dDABA-C, and D_{th} =1000 is represented by 3dDABA-D. It can be seen that with an increase in the number of iterations, the coverage of all algorithms increases gradually, and the regional finally balance. It is evident from the figure that 3dDABA-A has the highest coverage, reaching more than 90%.

Figure 9 compares the deployment time of different algorithms in different situations. The x-coordinate represents the number of iterations and the y-coordinate represents the deployment time. It can be seen from the figure that the 3dDABA algorithm has an excellent performance in terms of deployment time. Compared to the time of the Voronoi diagram method, the deployment time of 3dDABA-A is one-third of it, the deployment time of 3dDABA-B is one-ninth of it, and the deployment times of 3dDABA-C and 3dDABA-D are much lower than those of the Voronoi diagram method.

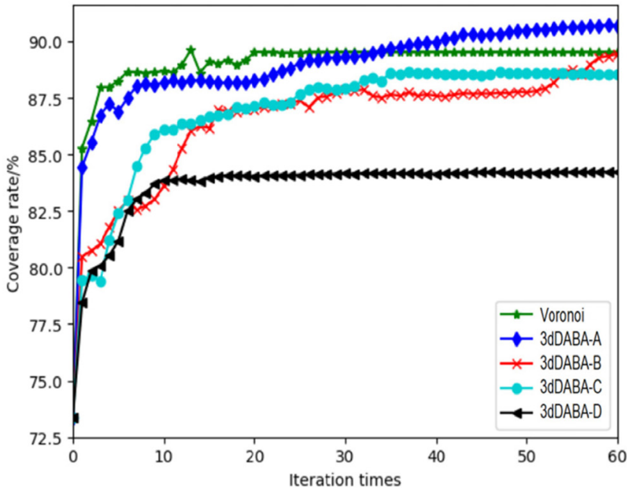


Figure 8. Coverage rate over time

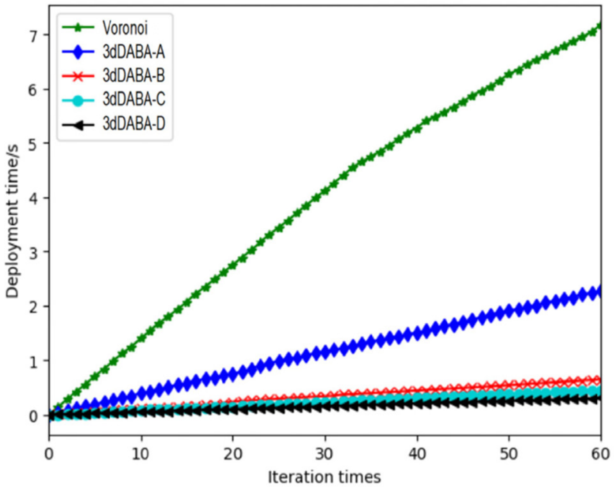


Figure 9. The deployment time varies with the number of iterations

Figure 10 compares the different methods under different conditions, with each node moving average distance during the deployment process. The x-coordinate represents the number of iterations, and the y-coordinate represents the average moving distance of nodes. It can be seen that with an increase in the number of iterations, the average distance is becoming smaller and smaller, approaching zero. This can attest that the proposed algorithm can be convergent, that nodes during the deployment process tend to be static, and that network can achieve a stable state. The results show that the average moving distance of the 3dDABA nodes was only slightly higher than that of the Voronoi diagram, and the difference was not significant.

In Figure 11, the nodes' number in the Voronoi diagram method is compared with the number of basic architectures in the 3dDABA algorithm method. The x-coordinate represents the name of each algorithm and

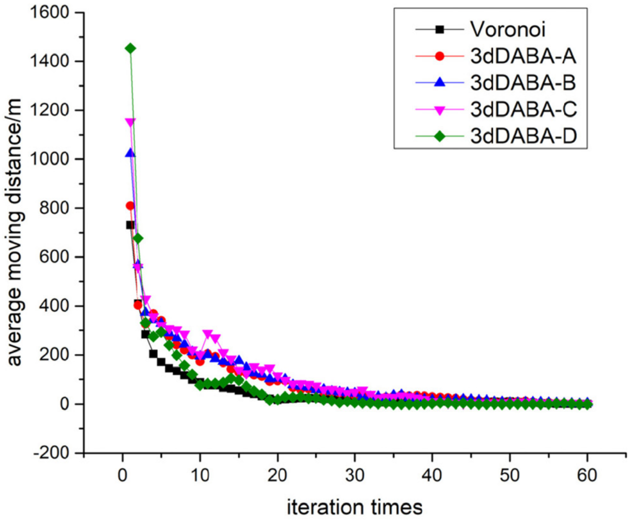


Figure 10. Node's average travel distance in each iteration

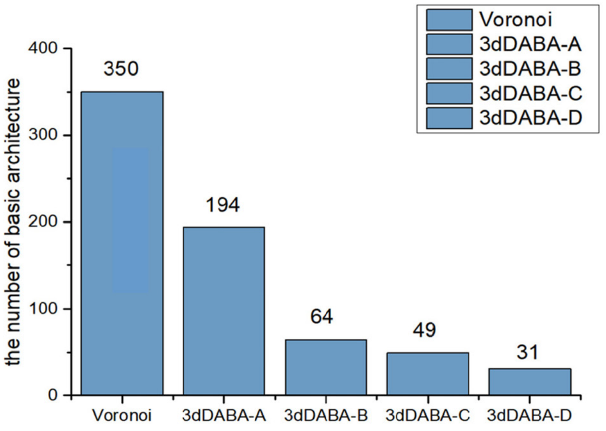


Figure 11 The number of basic architectures under different conditions

the y-coordinate represents the number of basic architecture. The construction of basic architectures greatly reduced the number of nodes involved in the construction of the Voronoi diagram, thus reducing the complexity of the whole algorithm. In the simulation experiment, the number of the 3dDABA-A nodes participating in the construction the Voronoi diagram was 194, which was 50% lower than in the Voronoi diagram method.The number of the 3dDABA-B nodes involved in constructing the Voronoi diagram was 64, which was 80% lower than in the Voronoi diagram method.

A comparison of the simulation results of the algorithm shows that 3dDABA has obvious advantages in deployment time, while coverage and average moving distance are similar to those in the traditional Voronoi diagram method. Combining deployment time, coverage, and average movement distance, 3dDABA-A and 3dDABA-B schemes work best, that is, 3dDABA works best when the critical distance between nodes is two times the perceived radius or three times the perceived radius.

4 Conclusion

To address problems in wireless sensor network node deployment in 3D space, this paper put forward a 3D wireless sensor network deployment algorithms based on Basic Architecture (3dDABA). The coverage rate, deployment time and the algorithm's moving distance deployment results are compared with the traditional Voronoi diagram deployment method through simulation experiment. Experimental results show that the 3dDABA algorithm can reduce the deployment time by two-thirds compared with the Voronoi diagram method, greatly reduce the complexity of the algorithm, and reduce the deployment time of the whole network.

Acknowledgments

This research was funded by Beijing Municipal Natural Science Foundation-Haidian original innovation joint fund (L182007); the National Natural Science Foundation of China (61702020) and supporting projects (PXM2018_014213_000033).

References

[1] L. F. M. Vieira, M. G. Almiron, A. A. F. Loureiro, 3D MANETs: Link Probability, Node Degree, Network Coverage and applications, *2011 IEEE Wireless Communications and Networking Conference (WCNC)*, Cancun, Mexico, 2011, pp. 2042-2047

[2] C. Miao, G. Dai, X. Zhao, Z. Tang, Q. Chen, 3D Self-Deployment Algorithm in Mobile Wireless Sensor Networks, *International Journal of Distributed Sensor Networks*, Vol. 11, No. 4, pp. 721921, 1-11, April, 2015.

[3] T. Brown, Z. Wang, T. Shan, F. Wang, J. Xue, On Wireless Video Sensor Network Deployment for 3D Indoor Space coverage, *SoutheastCon 2016*, Norfolk, VA, 2016, pp. 1-8.

[4] X. Li, L. Ci, M. Yang, C. Tian, X. Li, Deploying Three-Dimensional Mobile Sensor Networks Based on Virtual Forces Algorithm, *China Conference on Wireless Sensor Networks (CWSN 2012)*, Huangshan, China, pp. 204-216.

[5] N. Boufares, I. Khoufi, P. Minet, L. Saidane, Y. B. Saied, Three Dimensional Mobile Wireless Sensor Networks Redeployment Based On Virtual Forces, *2015 International Wireless Communications and Mobile Computing Conference (IWCMC)*, Dubrovnik, Croatia, 2015, pp. 563-568.

[6] N. Boufares, P. Minet, I. Khoufi, L. Saidane, Covering a 3D Flat Surface with Autonomous and Mobile Wireless Sensor Nodes, *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Valencia, Spain, 2017, pp. 1628-1633.

[7] T. W. Sung, C. S. Yang, Voronoi-based Coverage Improvement Approach for Wireless Directional Sensor Networks, *Journal*

of Network and Computer Applications, Vol. 39, pp. 202-213, March, 2014.

[8] F. Li, J. Luo, W. Wang, Y. He, Autonomous Deployment for Load Balancing $ k $-Surface Coverage in Sensor Networks, *IEEE Transactions on Wireless Communications*, Vol. 14, No. 1, pp. 279-293, January, 2015.

[9] N. Aslam, Optimizing Coverage in 3D Wireless Sensor Networks, in: Y. K. Tan, D. C. Hoang (Eds.), *Smart Wireless Sensor Networks*, Intech, 2010, pp. 189-204.

[10] T. G. Nguyen, C. So-In, N. G. Nguyen, Barrier Coverage Deployment Algorithms for Mobile Sensor Networks, *Journal of Internet Technology*, Vol. 18, No. 7, pp. 1689-1699, December, 2017.

[11] B. Cao, J. Zhao, Z. Lv, X. Liu, X. Kang, S. Yang, Deployment Optimization for 3D Industrial Wireless Sensor Networks Based on Particle Swarm Optimizers with Distributed Parallelism, *Journal of Network and Computer Applications*, Vol. 103, pp. 225-238, February, 2018.

[12] H. Yang, X. Li, Z. Wang, W. Yu, B. Huang, A Novel Sensor Deployment Method Based on Image Processing and Wavelet Transform to Optimize the Surface Coverage in WSNs, *Chinese Journal of Electronics*, Vol. 25, No. 3, pp. 495-502, May, 2016.

[13] S. Xie, J. Wang, X. Feng, M. Deng, Algorithm for Constructing Voronoi Diagram of Planar Points Based on Approximating and Extracting Vertices, *Acta Geodaetica et Cartographica Sinica*, Vol. 36, No. 4, pp. 436-442, November, 2007.

Biographies

Li Tan, born in 1980, Ph. D., professor. Her research interests include Wireless Sensor Network, Intelligent information network and machine learning.

Jiaqi Shi, born in 1996, M.S. candidate. Her research interests include Wireless Sensor Network.

Xiaojiang Tang, born in 1994,Doctor Degree.His research interests include Wireless Sensor Network.

Xiaofeng Lian, born in 1977, Ph. D., assistant professor. His research interests include Mobile robot, machine vision, 3D reconstruction.

Haoyu Wang, born in 1995, M.S. candidate. His research interests include Wireless Sensor Network.