The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

Using Voronoi Diagram and Genetic Algorithm to Deploy Nodes in Wireless Sensor Network

Farhad Nematy
Technical and Vocational University
Tabriz Branch
Tabriz, Iran
farhad_nematy@yahoo.com

Naeim Rahmani
Islamic Azad University
Tabriz Branch
Tabriz, Iran
n.rahmani@iaut.ac.ir

***Abstract-* A wireless sensor network (WSN) is a large-scale ad-hoc multi-hop network deployed (usually, at random) in a region of interest for surveillance purpose. Coverage is one of the important aspects of wireless sensor networks and many approaches introduced to maximize it. In this paper, a novel approach for maximizing coverage proposed. Voronoi diagram divides the field into cells and inside of each cell some holes exist. Different number of additional nodes must be placed inside cells to cover the holes because voronoi cells have different sizes. Genetic algorithm is used to determine best places for additional nodes to maximize the coverage. The proposed algorithm is distributed and optimization for each voronoi cell can be done in parallel to others. Optimal placement of nodes can guarantee the maximum coverage with less number of nodes and energy consumption decreases. Simulations results show that our new approach can outperform other earlier works.**

Key words: Node Placement, Voronoi Diagram, Wireless Sensor Networks, Genetic Algorithm, Coverage

I. Introduction

In recent years wireless sensor networks (WSN) have become one of the most active research areas due to the bright and interesting future promised to the world of information technology. Wireless Sensor Networks (WSNs) generally consist of a large number of low-cost, low-power, multifunctional sensor nodes that are small in size and communicate over short distances [1]. A sensor node consists of four basic components [2]: a processing unit, a power unit, a communication unit and a sensing unit. Also a sensor node may include optional units such as GPS, energy generator unit, mobility unit etc. In wireless sensor network each node sense the environment around itself and sends gathered data to base station hop-by-hop. Area coverage is one of the most fundamental problems in ad-hoc wireless sensor networks because it relates directly to optimization of resources in a field. In wireless sensor network, sensors may be deployed either randomly or deterministic, depending upon the application [3]. Deployment in a battlefield or hazardous areas is generally random whereas the deterministic deployment is preferred in amicable environments Also it is possible to use multi-robot systems to fulfill this task [4]. In most applications a random deployment is required and sensor nodes usually scatter from an aircraft. In random deployment method sensor density is not equal in different regions so some regions could not be covered because of lack of sensor nodes in that region. These uncovered areas are called coverage holes. The coverage is one of most important problems in WSN. There are different strategies for solving coverage problem which are categorized into three groups, namely; force based, grid based or computational geometry based [5]; Force based strategies use attraction and repulsion forces to determine the optimal position of the sensors while grid based strategies use grid points for determining the optimal position of the sensors. As for the computational geometry approach, Voronoi diagram and Delaunay triangulation are commonly used in WSN coverage optimization algorithm. Coverage can be classified into three classes; Area coverage, point coverage and barrier coverage. Area coverage, as the name suggests is on how to cover an area with the sensors, while point coverage deals with coverage for a set of points of interest. Decreasing the probability of undetected penetration is the main issue in barrier coverage. We are using area coverage where the objective is to maximize the coverage percentage [6].

Existence of coverage holes are very important in WSN and must be covered to increase QoS and the accuracy. Many researches exist that try to cover these holes [6, 7]. Most of these researches consider the network to be mobile or hybrid. Most of nodes in a hybrid network are stationary and there are few

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:
The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

mobile nodes. The main objective of using mobile sensor nodes is to heal coverage holes after the initial network deployment, such that the area coverage can be maximized while incurring the least moving cost. In some other researches a mobile network is assumed in which all nodes are mobile and thus nodes can move to calculated target locations [11,12].

When designing a hole healing algorithm, the following issues need to be addressed. First, how to distinguish the existence of a coverage hole and how to estimate the size of a hole. Second, what are the best target locations to relocate mobile nodes to repair coverage holes? Third, how to dispatch mobile nodes to the target locations while minimizing the moving and messaging cost. In this paper we propose a genetic algorithm to find best positions for additional mobile nodes in each voronoi cell. The rest of this paper organized as follows: Section II introduces voronoi diagram. Section III presents related works. In section IV Genetic Algorithm is described and our new approach based on Genetic Algorithm illustrated. The result of simulations presented in section V and section VI concludes the paper.

II. Voronoi Diagram

Computational geometry is frequently used in WSN coverage optimization, the most commonly used computational geometry approach are Voronoi diagram and Delaunay triangulation [5]. Voronoi diagram is partition of sites in such a way that points inside a polygon are closer to the site inside the polygon than any other sites, thus one of the vertices of the polygon is the farthest point of the polygon to the site inside it. Therefore Voronoi diagram can be used as one of the sampling method in determining WSN coverage; with the sensors act as the sites, if all Voronoi polygons vertices are covered then the field is fully covered otherwise coverage holes exist [18]. Delaunay triangulation is the dual of Voronoi diagram [13]. Voronoi diagram for 3 and 4 points is presented in figure 1 and figure 2.

A Delaunay triangle is formed by three sites provided if and only if the site's circumcircle does not contain other sites. The centre point of the circle is a Voronoi vertex with equal distance from each of the three sites. In [14] authors used Voronoi diagram in enhancing WSN coverage using mobility of sensors.

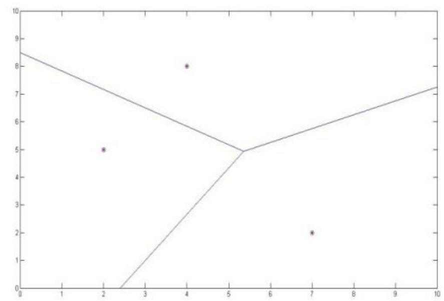

Figure 1- voronoi diagram for 3 points

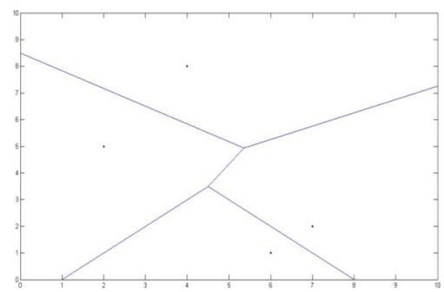

Figure 2- voronoi diagram for 4 points

Voronoi diagram [15] can be used to detect a coverage hole and to calculate the size of a coverage hole [16, 17]. A Voronoi diagram for N sensors s_1; s_2; . . . ; s_N in a plane is defined as the subdivision of the plane into N cells each for one sensor, such that the distance between any point in a cell and the sensor of the cell is closer than that distance between this point and any other sensors. Two Voronoi cells meet along a Voronoi edge and a sensor is a Voronoi neighbor of another sensor if they share a Voronoi edge. A Voronoi diagram is first constructed for all stationary sensor nodes, assuming that each node knows its own and its neighbor's coordinates. In [23] authors propose a localized construction algorithm to construct a local Voronoi diagram: Each node constructs its own Voronoi cell by considering only its 1-hop neighbors. After the local Voronoi diagram construction, the sensor field is divided into sub regions of Voronoi cells and each stationary node is within a Voronoi cell. Figure 3 illustrates a Voronoi diagram in a bounded sensor field, where the boundaries of the sensor field contribute to a Voronoi cell too. According to the property of a Voronoi diagram, all the points within a Voronoi cell are closest to only one node that lies within this cell. Therefore, if some points of a Voronoi cell are not

Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

covered by its generating node, these points will not be covered by any other sensor and contribute to coverage holes. If a sensor covers all of its Voronoi cell's vertices, then there are no uncovered points within its Voronoi cell; otherwise, uncovered points exist within its Voronoi cell.

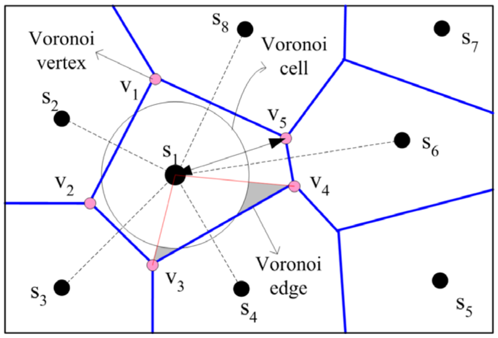

Figure 3 - voronoi diagram

III. Related works

In one random network deployment, it is not uncommon that the covered area of a sensor overlaps others even if nodes are uniformly scattered. In a mobile sensor network where all nodes can move around, the mobile nodes can adjust their positions after initial deployment in order to reduce their overlaps and maximize area coverage. The coordination of mobile node's movement is a much challenging issue in a mobile sensor network. A node's movement may change the already covered area of itself and its neighbors and become a cause for other nodes to move, which may cause the oscillations of node's movement and nodes may never be stopped. This is in contrast to the hole heal problem where mobile nodes are to move to the coverage holes caused by the stationary nodes. Optimal node placement is a very challenging problem that has been proven to be NP-Hard for most of the formulations of sensor deployment [18–20]. Genetic algorithm can be used to find the best places for additional mobile nodes. In our previous work [27] a genetic algorithm based node placement is used for cluster based architecture. At first all nodes scattered in the field randomly and then genetic algorithm is used to find optimal places for cluster heads to cover maximum number of nodes and consequently maximize the area coverage. In [28], an algorithm proposed using a combination of mobile and static sensors to construct sensor network to balance cost and sensor coverage. They identified the problem of deploying mobile sensors in a mixed sensor network as an NP-complete problem and designed bidding protocols to tackle this problem in a distributed fashion, in which static sensors act as bidders and mobile sensors act as hole-healing servers. In their simulations users can determine the percentage of mobile sensors to get the most economical deployment of sensors to construct a network satisfying the coverage requirement.

IV. Genetic algorithm

A. Overview of Genetic Algorithm

The genetic algorithm (GA) is a technique for randomized search and optimization [21,21,223] and has been applied in a wide range of studies in solving optimization problems, especially problems that are not well structured and interact with large numbers of possible solutions. In this paper, we have used standard GA terminology [22, 23] as follows. A GA starts with a set of randomly generated possible solutions called a population. Each individual solution in the population is known as a chromosome or an individual. Each chromosome may be represented as a simple string or an array of genes, which contain a part of the solution. The values of genes are called alleles. The length of all chromosomes in a population should be the same. A fitness function is provided to assign the fitness value for each individual. This function is based on how close an individual is to the optimal solution – the higher the fitness value, the closer is the solution to the optimal solution. Two randomly selected chromosomes, known as parents, can exchange genetic information in a process called recombination or crossover, to produce two new chromosomes known as child or offspring. If both the parents share a particular pattern in their chromosome, then the same pattern will be carried over to the offsprings. To obtain a good solution, mutation is often applied on randomly chosen chromosomes, after the process of crossover. Mutation helps to restore any lost genetic values when the population converges too fast. Once the processes of crossover and mutation have occurred in a population, the chromosomes for the next generation are selected. To ensure that the new generation is at least as good as the previous generation, some of the poorest performing individuals of the current generation can be replaced by the same number of the best performing individuals from the previous generation. This process is called elitism [24]. This cycle is repeated until the stopping criterion of the algorithm is met. The steps of a standard GA [25] are outlined in Algorithm 1.

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

Algorithm 1: Genetic algorithm

```
Int Main()
{
    Generate an initial population
   Compute the fitness of each individual
   while (not stopping criterion)
     {
      Choose parents from population.
      Perform crossover to produce offsprings.
      Perform mutations.
      Compute fitness of each individual.
     Replace the parents by the corresponding offsprings in
     new generation.
     }
}
```

B. Proposed Algorithm

In our network model it is assumed that all stationary sensors are of the same type and deployed randomly in the sensing field. Initially our proposed algorithm uses voronoi diagram to divide the field into cells and then for each cell it uses genetic algorithm to deploy additional mobile nodes to the holes.

1. Initial population

We use a 2*n array to represent the solutions in which i_{th} cells are coordination of a additional node inside a cell where n is number of additional nodes needed for each cell. Each coordination in each chromosome is random but it must be inside the corresponding voronoi cell.

x	34	14	75	22	18	3
y	69	93	6	51	50	24

Figure 4- representation of a chromosome

2. Fitness Function

Fitness function is used to determine better solutions that can cover more area with less overlap. If there is no overlap between two nodes in a voronoi cell their distance is equal or more than 2r. Figure 3 shows two nodes with no overlap between them. When all of nodes inside a cell are positioned in this way they can cover whole cell with minimum number of nodes. So nodes with overlaps must be avoided or at least nodes with less overlap must be chosen.

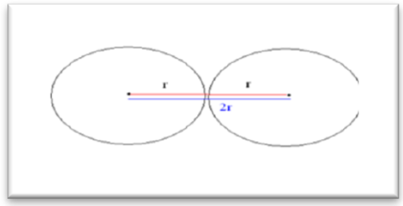

Figure 5-distance between two nodes when there is no overlap

When two nodes have overlap, their distance from each other is less than 2r so we penalty it to make its fitness value less than others. This penalty value is a function of their distance and as their distance decrease the penalty increases and consequently the fitness value decreases. Figure 6 depicts this overlap section. Amount of overlap (blue line in figure 6) can be calculated as follows

X= 2r – distance (p1, p2) (1)

Total_ideal_coverage= n * (r * r * 3.14) (2)

Penalty(i)=((X^ 2)*3.14)/2 (3)

Total_penalty=$\sum_{i=0}^{n}$ Penalty(i) (4)

Fitness= Total_ideal_coverage – Total_penalty (5)

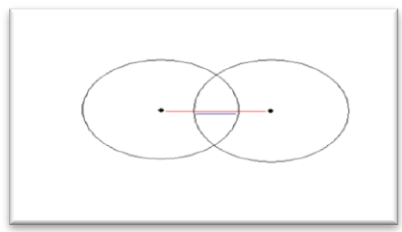

Figure 6- distance between two nodes when overlap exists

3. Selection

Selection mechanism is one of the important parts of a genetic algorithm. We used tournament selection mechanism to select chromosomes for reproducing new generations. In tournament selection each time k individuals are picked randomly and then the chromosome with the best fitness value among them are selected for mating. Here we used k=5.

4. Crossover

To produce new offsprings from the selected parents, we have randomly used a one point crossover. two parents in the population are selected and crossovered with a crossover probability equal to 0.8. Figure 7 represents this crossover process.

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

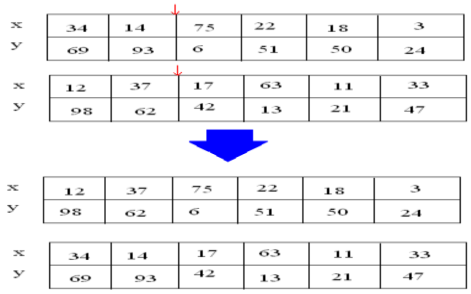

Figure 7- one point cross over

The resulted offsprings after crossover are same to their parents because each coordination in chromosomes is inside the corresponding voronoi cell and no new coordination can be out of its corresponding voronoi cell.

5. Mutation

For mutating chromosomes in our proposed genetic algorithm, with a probability equal to 0.3, a new random coordination is calculated inside the voronoi cell and is assigned to a random point in chromosome. Figure 8 depicts this mutation process.

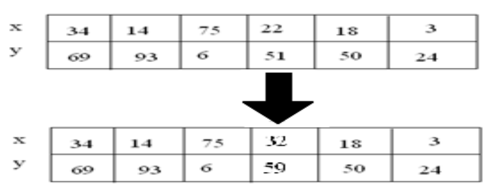

Figure 8-mutation process with new random coordinate inside voronoi cell

6. Termination criterion

A termination criterion is one of the important parameters in genetic algorithms. There are several methods for specifying termination criterion such as determining a constant number of iterations or until a predefined value for fitness acquired or until in M iterations the fitness value doesn't change. Here we used a constant number of iterations equal to 200.

V. Simulation Results

Simulation in MATLAB has been used to evaluate the proposed algorithm. In this simulation, a field of 100×100meters has been used where 60 nodes are scattered randomly which is presented in figure 9. Then voronoi diagram of this field has been calculated and to optimize each voronoi cell the genetic algorithm has been used. Genetic algorithm used to optimize each voronoi cell by placing k new nodes in each voronoi cell and relocating them to find best place for them inside the voronoi diagram to achieve maximum coverage inside each voronoi cell. Simulation parameters are presented in table 1.

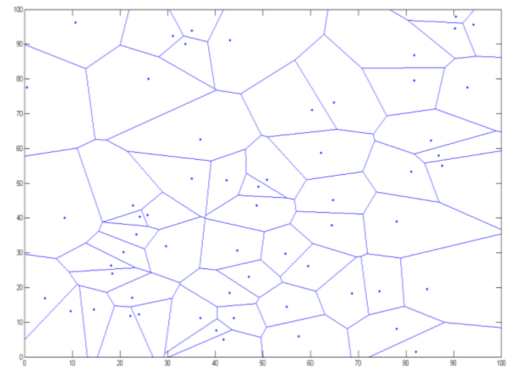

Figure 9- 100 meters in 100 meters field with 60 nodes and its voronoi diagram

Table 1- Simulation Parameters

parameter	value	description
n	60	Number of nodes in field
radius	2,5	Sensing radius of each node
Number_of_iterations	200	Number of iterations
Population_size	200	Number of individuals in population
Number_of_nodes	------	Number of additional nodes to heal holes is dynamic
PC	0.8	Percentage of crossover
PM	0.5	Percentage of mutation

Figure 10 and 11 shows approximate number of nodes needed to cover each voronoi cell when sensing range of each node is 2 and 5 meters.

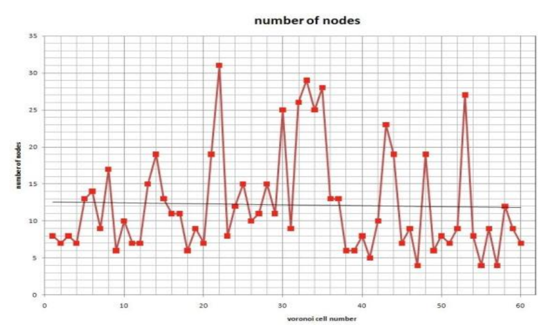

Figure 10- number of nodes for each voronoi cell with r=2m

The International Journal of Soft Computing and Software Engineering [JSCSE], Vol. 3, No. 3, Special Issue:

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13], San Francisco, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

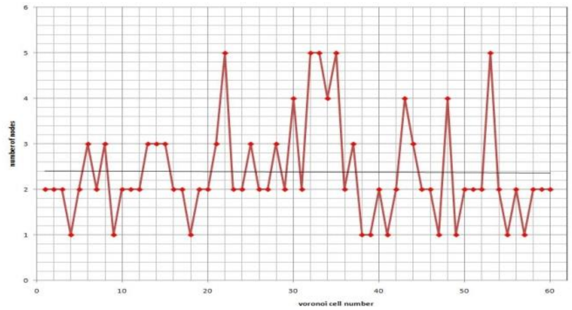

Figure 11-number of nodes for each voronoi cell with r=5m

To calculate the amount of coverage in each voronoi cell, area of each cell must be calculated. The area of voronoi cells in our experiment is presented in table 2. The first experiment done with 5 meters sensing radius and additional nodes added to each voronoi cell and the following results captured for this experiment. The covered area and coverage percentage for each voronoi cell is presented in table 2 and average coverage of nodes is 92.7 %. Figure 12 depicts these results when sensing radius is 5 meters. Second experiment done with sensing radius equal to 2 meters and the results are presented in figure 13 where average coverage is 95.75 percent which represent better coverage than previous experiment. This happens because smaller sensing areas can cover narrow areas inside cells better than bigger sensing areas.

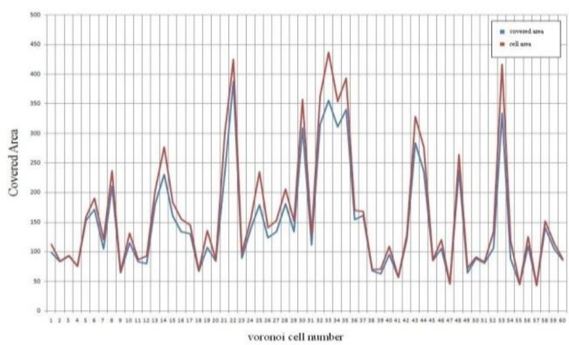

Figure 12- covered area with r=5m

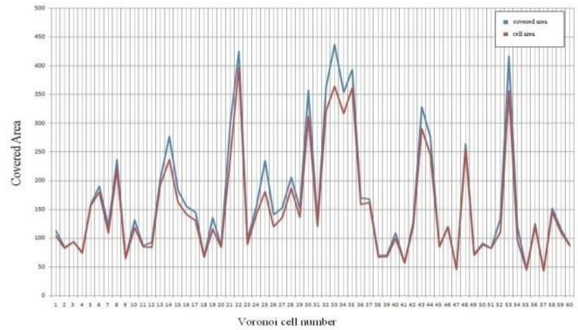

Figure 13-covered area with r=2m

By comparing results of proposed approach with results in [26, 27] better coverage in the new approach has been observed. In [26] a genetic algorithm based layout optimization presented and 3 experiments are done that their results are 70%, 80% and 90% coverage of the field also the genetic algorithm used in [28] has different experiments that best results of these experiments are 92% and 94% coverage of the field. Our proposed algorithm covered 95.75% of the field which is better than previous works. Also in [28] different number of mobile and stationary nodes used to cover the field and in three different scenarios 90%,95% and 98% of field covered by nodes but they used too many nodes to achieve these results. Our proposed approach used less number of nodes than [27] and had better coverage.

VI. Conclusion

In this paper a new approach is proposed to increase coverage which uses voronoi diagram to divide the field into cells and uses genetic algorithm to find best places to put additional mobile nodes to heal the coverage holes. We compared our proposed approach with other works in literature and simulation results show that our proposed approach has better performance.

Table 2- voronoi cell areas

Cell no.	Cell area	Covered area	Coverage percent
1	113.1	99.4905639	87.9669
2	84.9	84.0824979	99.0371
3	94.5	93.0100185	98.4233
4	76.3	76.0498123	99.6721
5	159.1	153.3144876	96.3636
6	191.1	171.6032136	89.7976
7	120.8	105.383504	87.238
8	236.5	211.2089265	89.3061
9	67.2	64.81944	96.4575
10	131.9	115.4830665	87.5535
11	87.6	84.228276	96.151
12	93.2	80.689298	86.5765
13	203.8	181.3536718	88.9861
14	277	230.814959	83.3267
15	184.2	160.7045532	87.2446
16	155.3	134.1068302	86.3534

Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

17	145.7	130.5764857	89.6201	
18	68.1	67.4849208	99.0968	
19	135.8	107.894458	79.451	
20	87.1	85.1172556	97.7236	
21	300.5	233.99334	77.868	
22	424.6	387.9098894	91.3589	
23	98.7	89.8225272	91.0056	
24	156.7	139.9522174	89.3122	
25	235.4	179.5405216	76.2704	
26	141.1	124.200453	88.023	
27	152.8	134.0257696	87.7132	
28	206.4	180.8759568	87.6337	
29	152.1	134.0620047	88.1407	
30	357.6	309.6394032	86.5882	
31	129.3	111.7869615	86.4555	
32	362.1	315.1588044	87.0364	
33	436.8	355.406688	81.366	
34	353.6	310.8787552	87.9182	
35	393.4	340.0781706	86.4459	
36	170.6	154.8802336	90.7856	
37	168.8	161.1891456	95.4912	
38	70	68.38006	97.6858	
39	71.3	63.5077656	89.0712	
40	109.3	95.6917128	87.5496	
41	57.6	57.6	100	
42	127.9	121.6504223	95.1137	
43	328.1	283.8081405	86.5005	
44	276.9	235.4707758	85.0382	
45	86.8	85.7658648	98.8086	
46	120.5	106.0724145	88.0269	
47	46.1	46.1	100	
48	264.3	237.2547096	89.7672	
49	73.1	64.7127253	88.5263	
50	91.6	89.0943736	97.2646	
51	83.4	81.0708048	97.2072	
52	134.2	106.7136928	79.5184	
53	416	334.421568	80.3898	
54	119.7	89.3674215	74.6595	
55	45.4	45.0675358	99.2677	
56	125.8	110.5307734	87.8623	
57	43.7	43.7	100	
58	152.5	140.8978	92.392	
59	114.9	105.1459092	91.5108	
60	87.8	86.77713	98.835	

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey, Computer Networks 38 (2002) 393–422.

[2] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A Survey on Sensor networks. IEEE Communications Magazine,40(8):102-114

[3] M. Ishizuka, M. Aida, *Performance study of node placement in sensor networks*, in: Proc. of 24th International Conference on Distributed Computing Systems Workshops, 2004, pp. 598–603.

[4] F. Nematy , A. M. Rahmani , M. Teshnelab , N. Rahmani, "Ant Colony Based Node Deployment and Search in Wireless Sensor Networks", International Conference on Computational Intelligence and Communication Networks (CICN), 2010 , pp.363 - 366

[5] N. A. Ab. Aziz, K. Ab. Aziz, W. Z. Wan Ismail, Coverage Strategies for Wireless Sensor Networks, World Academy of Science, Engineering and Technology ,2009

[6] A. Ghosh, Estimating coverage holes and enhancing coverage in mixed sensor networks, in: IEEE International Conference on Local Computer Networks, 2004, pp. 68–76.

[7] G. Wang, G. Cao, P. Berman, T.F.L. Porta, Bidding protocols for deploying mobile sensors, IEEE Transactions on Mobile Computing 6 (5) (2007) 515–528.

[8] P.-C. Wang, T.-W. Hou, R.-H. Yan, Maintaining coverage by progressive crystallattice permutation in mobile wireless sensor networks, in: IEEE International Conference on Systems and Networks Communication (ICSNC), 2006, pp. 1–6.

[9] Y.-C. Wang, C.-C. Hu, Y.-C. Tseng, Efficient placement and dispatch of sensors in a wireless sensor network, IEEE Transactions on Mobile Computing 7 (2) (2008) 262–274.

[10] S. Chellappan, X. Bai, B. Ma, D. Xuan, C. Xu, Mobility limited flip-based sensor networks deployment, IEEE Transactions on Parallel and Distributed Systems 18 (2) (2007) 199–211.

[11] S. Yang, M. Li, J. Wu, Scan-based movement-assisted sensor deployment methods in wireless sensor networks, IEEE Transactions on Parallel and Distributed Systems 18 (8) (2007) 1108–1121.

[12] J. Wu, S. Yang, Optimal movement-assisted sensor deployment and its extensions in wireless sensor networks, Simulation Modelling Practice and Theory 15 (4) (2007) 383–399.

[13] Aurenhammer, F. and Klein, R "Voronoi diagrams." In J. Sack and G.Urrutia, editors, "Handbook of Computational Geometry" Chapter V, pages 201-290. Elsevier Science Publishing, 2000.

[14] Wang, G., Cao, G. and Porta, T.L "Movement-Assisted Sensor Deployment", IEEE INFOCOM 2004, Vol. 4, pp 2469-2479

[15] F. Aurenhammer, Voronoi diagrams – a survey of a fundamental geometric data structure, ACM Computing Surveys 23 (4) (1991) 345–406

[16] A. Ghosh, Estimating coverage holes and enhancing coverage in mixed sensor networks, in: IEEE International Conference on Local Computer Networks, 2004, pp. 68–76.

[17] G. Wang, G. Cao, P. Berman, T.F.L. Porta, Bidding protocols for deploying mobile sensors, IEEE Transactions on Mobile Computing 6 (5) (2007) 515–528.

[18] A. Efrat, S. Har-Peled, J.S.B. Mitchell, Approximation algorithms for two optimal location problems in sensor networks, in: Proceedings of the 3rd International Conference on Broadband Communications, Networks and Systems (Broadnets'05), Boston, Massachusetts , October 2005.

[19] X. Cheng, DZ Du, L. Wang, B. Xu, Relay sensor placement in wireless sensor networks, ACM/Springer Journal of Wireless Networks, in press.

[20] S. Poduri, S. Pattem, B. Krishnamachari, G.S. Sukhatme, Sensor network configuration and the curse of dimensionality, in: Proceedings of the 3rd IEEE Workshop on Embedded Networked Sensors, Cambridge, MA, May 2006.

[21] C. Gazen, C. Ersoy, Genetic algorithms for designing multihop lightwave network topologies, Artificial Intelligence in Engineering 13 (1999) 211–221.

[22] D.E. Goldberg, Genetic Algorithms in Search, Optimization, and Machine Learning, Addison Wesley, Reading, MA, 1989.

[23] J. Holland, Adaptation in Natural and Artificial Systems, University of Michigan Press, 1975

[24] L. Davis, Handbook of Genetic Algorithms, Van Nostrand Reinhold, New York, 1991.

[25] Ming-Tsung Chen, Shian-Shyong Tseng, A genetic algorithm for multicast routing under delay constraint in WDM network with

The Proceeding of International Conference on Soft Computing and Software Engineering 2013 [SCSE'13],
San Francisco, CA, U.S.A., March 2013
Doi: 10.7321/jscse.v3.n3.107 e-ISSN: 2251-7545

different light splitting, Journal of Information Science and Engineering 21 (2005) 85–108

[26] Amol P. Bhondekar, Renu Vig, Madan Lal Singla, C. Ghanshyam, Pawan Kapur," Genetic Algorithm Based Node Placement Methodology for Wireless Sensor Networks". Proceedings of International MultiConference of Engineers and Computer scientists 2009 vol I. IMECS 2009, March 18-20,2009,Hong Kong.

[27] F. Nematy, N. Rahmani, R. Yagouti, An Evolutionary Approach for Relocating Cluster Heads in Wireless Sensor Networks, ", International Conference on Computational Intelligence and Communication Networks (CICN), 2010 , pp.323 - 326

[28] G. Wang, G. Cao, P. Berman, T.F.L. Porta, Bidding protocols for deploying mobile sensors, IEEE Transactions on Mobile Computing 6 (5) (2007) 515–528.